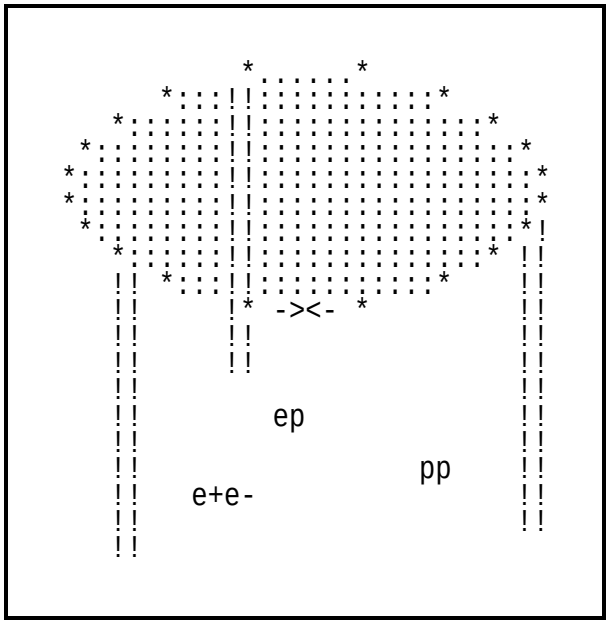
hep-ph/0108264
LU TP 01–21
August 2001

PYTHIA 6.2

Physics and Manual

Torbjörn Sjöstrand, Leif Lönnblad
Department of Theoretical Physics,
Lund University, Sölvegatan 14A,
S-223 62 LUND, SWEDEN

Stephen Mrenna
Physics Department,
University of California at Davis,
One Shields Avenue,
DAVIS, CA 95616, USA



Abstract

The PYTHIA program can be used to generate high-energy-physics 'events', i.e. sets of outgoing particles produced in the interactions between two incoming particles. The objective is to provide as accurate as possible a representation of event properties in a wide range of reactions, with emphasis on those where strong interactions play a rôle, directly or indirectly, and therefore multihadronic final states are produced. The physics is then not understood well enough to give an exact description; instead the program has to be based on a combination of analytical results and various QCD-based models. This physics input is summarized here, for areas such as hard subprocesses, initial- and final-state parton showers, beam remnants and underlying events, fragmentation and decays, and much more. Furthermore, extensive information is provided on all program elements: subroutines and functions, switches and parameters, and particle and process data. This should allow the user to tailor the generation task to the topics of interest.

The information in this edition of the manual refers to PYTHIA version 6.200, of 31 August 2001.

The official reference to the latest published version is
T. Sjöstrand, P. Edén, C. Friberg, L. Lönnblad, G. Miu, S. Mrenna and E. Norrbin, Computer Physics Commun. **135** (2001) 238.

Preface

The PYTHIA program is frequently used for event generation in high-energy physics. The emphasis is on multiparticle production in collisions between elementary particles. This in particular means hard interactions in e^+e^-, pp and ep colliders, although also other applications are envisaged. The program is intended to generate complete events, in as much detail as experimentally observable ones, within the bounds of our current understanding of the underlying physics. Many of the components of the program represents original research, in the sense that models have been developed and implemented for a number of aspects not covered by standard theory.

Historically, the family of event generators from the Lund group was begun with JETSET in 1978. The PYTHIA program followed a few years later. With time, the two programs so often had to be used together that it made sense to merge them. Therefore PYTHIA 5.7 and JETSET 7.4 were the last versions to appear individually; as of PYTHIA 6.1 all the code is collected under the PYTHIA heading. At the same time, the SPYTHIA sideline of PYTHIA was reintegrated. Both programs have a long history, and several manuals have come out. The most recent one is

T. Sjöstrand, P. Edén, C. Friberg, L. Lönnblad, G. Miu, S. Mrenna and E. Norrbin,
Computer Physics Commun. **135** (2001) 238,

so please use this for all official references. Additionally remember to cite the original literature on the physics topics of particular relevance for your studies. (There is no reason to omit references to good physics papers simply because some of their contents have also been made available as program code.)

Event generators often have a reputation for being 'black boxes'; if nothing else, this report should provide you with a glimpse of what goes on inside the program. Some such understanding may be of special interest for new users, who have no background in the field. An attempt has been made to structure the report sufficiently well so that many of the sections can be read independently of each other, so you can pick the sections that interest you. We have tried to keep together the physics and the manual sections on specific topics, where practicable.

A large number of persons should be thanked for their contributions. Bo Andersson and Gösta Gustafson are the originators of the Lund model, and strongly influenced the early development of the programs. Hans-Uno Bengtsson is the originator of the PYTHIA program. Mats Bengtsson is the main author of the final-state parton-shower algorithm. Patrik Edén has contributed an improved popcorn scenario for baryon production. Christer Friberg has helped develop the expanded photon physics machinery, Emanuel Norrbin the new matrix-element matching of the final-state parton shower algorithm and the handling of low-mass strings, and Gabriela Miu the matching of initial-state showers. Peter Skands has contributed the code for lepton-number-violating decays in supersymmetry.

Further comments on the programs and smaller pieces of code have been obtained from users too numerous to be mentioned here, but who are all gratefully acknowledged. To write programs of this size and complexity would be impossible without a strong support and user feedback. So, if you find errors, please let us know.

The moral responsibility for any remaining errors clearly rests with the authors. However, kindly note that this is a 'University World' product, distributed 'as is', free of charge, without any binding guarantees. And always remember that the program does not represent a dead collection of established truths, but rather one of many possible approaches to the problem of multiparticle production in high-energy physics, at the frontline of current research. Be critical!

Contents

1 Introduction **1**

2 Physics Overview **9**
2.1 Hard Processes and Parton Distributions . . . 9
2.2 Initial- and Final-State Radiation . . . 13
2.3 Beam Remnants and Multiple Interactions . . . 16
2.4 Hadronization . . . 17

3 Program Overview **21**
3.1 Update History . . . 21
3.2 Program Installation . . . 24
3.3 Program Philosophy . . . 26
3.4 Manual Conventions . . . 27
3.5 Getting Started with the Simple Routines . . . 28
3.6 Getting Started with the Event Generation Machinery . . . 33

4 Monte Carlo Techniques **39**
4.1 Selection From a Distribution . . . 39
4.2 The Veto Algorithm . . . 41
4.3 The Random Number Generator . . . 43

5 The Event Record **47**
5.1 Particle Codes . . . 47
5.2 The Event Record . . . 55
5.3 How The Event Record Works . . . 57
5.4 The HEPEVT Standard . . . 61

6 The Old e^+e^- Annihilation Routines **65**
6.1 Annihilation Events in the Continuum . . . 65
6.2 Decays of Onia Resonances . . . 74
6.3 Routines and Common Block Variables . . . 75
6.4 Examples . . . 81

7 Process Generation **83**
7.1 Parton Distributions . . . 83
7.2 Kinematics and Cross Section for a $2 \to 2$ Process . . . 89
7.3 Resonance Production . . . 91
7.4 Cross-section Calculations . . . 95
7.5 $2 \to 3$ and $2 \to 4$ Processes . . . 101
7.6 Resonance Decays . . . 103
7.7 Nonperturbative Processes . . . 105

8 Physics Processes **113**
8.1 The Process Classification Scheme . . . 113
8.2 QCD Processes . . . 122
8.3 Physics with Incoming Photons . . . 127
8.4 Electroweak Gauge Bosons . . . 133
8.5 Higgs Production . . . 136
8.6 Non-Standard Physics . . . 142
8.7 Supersymmetry . . . 149
8.8 Polarization . . . 153
8.9 Main Processes by Machine . . . 153

9 The Process Generation Program Elements **156**
9.1 The Main Subroutines . . . 156
9.2 Switches for Event Type and Kinematics Selection . . . 160
9.3 The General Switches and Parameters . . . 166
9.4 Further Couplings . . . 189
9.5 Supersymmetry Common Blocks and Routines . . . 193
9.6 General Event Information . . . 197
9.7 How to Generate Weighted Events . . . 202
9.8 How to Run with Varying Energies . . . 207
9.9 How to Include External Processes . . . 209
9.10 Interfaces to Other Generators . . . 228
9.11 Other Routines and Common Blocks . . . 232

10 Initial- and Final-State Radiation **248**
10.1 Shower Evolution . . . 248
10.2 Final-State Showers . . . 251
10.3 Initial-State Showers . . . 261
10.4 Routines and Common Block Variables . . . 270

11 Beam Remnants and Underlying Events **279**
11.1 Beam Remnants . . . 279
11.2 Multiple Interactions . . . 282
11.3 Pile-up Events . . . 290
11.4 Common Block Variables . . . 291

12 Fragmentation **296**
12.1 Flavour Selection . . . 296
12.2 String Fragmentation . . . 301
12.3 Independent Fragmentation . . . 309
12.4 Other Fragmentation Aspects . . . 311

13 Particles and Their Decays **319**
13.1 The Particle Content . . . 319
13.2 Masses, Widths and Lifetimes . . . 320
13.3 Decays . . . 322

14 The Fragmentation and Decay Program Elements **327**
14.1 Definition of Initial Configuration or Variables . . . 327
14.2 The Physics Routines . . . 329
14.3 The General Switches and Parameters . . . 332
14.4 Further Parameters and Particle Data . . . 346
14.5 Miscellaneous Comments . . . 353
14.6 Examples . . . 355

15 Event Study and Analysis Routines **358**
15.1 Event Study Routines . . . 358
15.2 Event Shapes . . . 363
15.3 Cluster Finding . . . 367
15.4 Event Statistics . . . 371
15.5 Routines and Common Block Variables . . . 373
15.6 Histograms . . . 382

16 Summary and Outlook **386**

References **387**

Subprocess Summary Table **401**

Index of Subprograms and Common Block Variables **404**

1 Introduction

Multiparticle production is the most characteristic feature of current high-energy physics. Today, observed particle multiplicities are typically between ten and a hundred, and with future machines this range will be extended upwards. The bulk of the multiplicity is found in jets, i.e. in collimated bunches of hadrons (or decay products of hadrons) produced by the hadronization of partons, i.e. quarks and gluons. (For some applications it will be convenient to extend the parton concept also to some non-coloured but showering particles, such as electrons and photons.)

The Complexity of High-Energy Processes

To first approximation, all processes have a simple structure at the level of interactions between the fundamental objects of nature, i.e. quarks, leptons and gauge bosons. For instance, a lot can be understood about the structure of hadronic events at LEP just from the 'skeleton' process $e^+e^- \to Z^0 \to q\overline{q}$. Corrections to this picture can be subdivided, arbitrarily but conveniently, into three main classes.

Firstly, there are bremsstrahlung-type modifications, i.e. the emission of additional final-state particles by branchings such as $e \to e\gamma$ or $q \to qg$. Because of the largeness of the strong coupling constant α_s, and because of the presence of the triple gluon vertex, QCD emission off quarks and gluons is especially prolific. We therefore speak about 'parton showers', wherein a single initial parton may give rise to a whole bunch of partons in the final state. Also photon emission may give sizeable effects in e^+e^- and ep processes. The bulk of the bremsstrahlung corrections are universal, i.e. do not depend on the details of the process studied, but only on one or a few key numbers, such as the momentum transfer scale of the process. Such universal corrections may be included to arbitrarily high orders, using a probabilistic language. Alternatively, exact calculations of bremsstrahlung corrections may be carried out order by order in perturbation theory, but rapidly the calculations then become prohibitively complicated and the answers correspondingly lengthy.

Secondly, we have 'true' higher-order corrections, which involve a combination of loop graphs and the soft parts of the bremsstrahlung graphs above, a combination needed to cancel some divergences. In a complete description it is therefore not possible to consider bremsstrahlung separately, as assumed here. The necessary perturbative calculations are usually very difficult; only rarely have results been presented that include more than one non-'trivial' order, i.e. more than one loop. As above, answers are usually very lengthy, but some results are sufficiently simple to be generally known and used, such as the running of α_s, or the correction factor $1+\alpha_s/\pi+\cdots$ in the partial widths of $Z^0 \to q\overline{q}$ decay channels. For high-precision studies it is imperative to take into account the results of loop calculations, but usually effects are minor for the qualitative aspects of high-energy processes.

Thirdly, quarks and gluons are confined. In the two points above, we have used a perturbative language to describe the short-distance interactions of quarks, leptons and gauge bosons. For leptons and colourless bosons this language is sufficient. However, for quarks and gluons it must be complemented with the structure of incoming hadrons, and a picture for the hadronization process, wherein the coloured partons are transformed into jets of colourless hadrons, photons and leptons. The hadronization can be further subdivided into fragmentation and decays, where the former describes the way the creation of new quark-antiquark pairs can break up a high-mass system into lower-mass ones, ultimately hadrons. (The word 'fragmentation' is also sometimes used in a broader sense, but we will here use it with this specific meaning.) This process is still not yet understood from first principles, but has to be based on models. In one sense, hadronization effects are overwhelmingly large, since this is where the bulk of the multiplicity comes from. In another sense, the overall energy flow of a high-energy event is mainly determined by the perturbative processes, with only a minor additional smearing caused by the hadronization step. One may therefore pick different levels of ambition, but in general detailed

studies require a detailed modelling of the hadronization process.

The simple structure that we started out with has now become considerably more complex — instead of maybe two final-state partons we have a hundred final particles. The original physics is not gone, but the skeleton process has been dressed up and is no longer directly visible. A direct comparison between theory and experiment is therefore complicated at best, and impossible at worst.

Event Generators

It is here that event generators come to the rescue. In an event generator, the objective strived for is to use computers to generate events as detailed as could be observed by a perfect detector. This is not done in one step, but rather by 'factorizing' the full problem into a number of components, each of which can be handled reasonably accurately. Basically, this means that the hard process is used as input to generate bremsstrahlung corrections, and that the result of this exercise is thereafter left to hadronize. This sounds a bit easier than it really is — else this report would be a lot thinner. However, the basic idea is there: if the full problem is too complicated to be solved in one go, try to subdivide it into smaller tasks of manageable proportions. In the actual generation procedure, most steps therefore involve the branching of one object into two, or at least into a very small number, with the daughters free to branch in their turn. A lot of book-keeping is involved, but much is of a repetitive nature, and can therefore be left for the computer to handle.

As the name indicates, the output of an event generator should be in the form of 'events', with the same average behaviour and the same fluctuations as real data. In the data, fluctuations arise from the quantum mechanics of the underlying theory. In generators, Monte Carlo techniques are used to select all relevant variables according to the desired probability distributions, and thereby ensure randomness in the final events. Clearly some loss of information is entailed: quantum mechanics is based on amplitudes, not probabilities. However, only very rarely do (known) interference phenomena appear that cannot be cast in a probabilistic language. This is therefore not a more restraining approximation than many others.

Once there, an event generator can be used in many different ways. The five main applications are probably the following:

- To give physicists a feeling for the kind of events one may expect/hope to find, and at what rates.
- As a help in the planning of a new detector, so that detector performance is optimized, within other constraints, for the study of interesting physics scenarios.
- As a tool for devising the analysis strategies that should be used on real data, so that signal-to-background conditions are optimized.
- As a method for estimating detector acceptance corrections that have to be applied to raw data, in order to extract the 'true' physics signal.
- As a convenient framework within which to interpret the observed phenomena in terms of a more fundamental underlying theory (usually the Standard Model).

Where does a generator fit into the overall analysis chain of an experiment? In 'real life', the machine produces interactions. These events are observed by detectors, and the interesting ones are written to tape by the data acquisition system. Afterwards the events may be reconstructed, i.e. the electronics signals (from wire chambers, calorimeters, and all the rest) may be translated into a deduced setup of charged tracks or neutral energy depositions, in the best of worlds with full knowledge of momenta and particle species. Based on this cleaned-up information, one may proceed with the physics analysis. In the Monte Carlo world, the rôle of the machine, namely to produce events, is taken by the event generators described in this report. The behaviour of the detectors — how particles produced by the event generator traverse the detector, spiral in magnetic fields, shower in calorimeters, or sneak out through cracks, etc. — is simulated in programs such as GEANT [Bru89]. Traditionally, this latter activity is called event simulation, which is somewhat unfortunate since the same words could equally well be applied to what,

here, we call event generation. A more appropriate term is detector simulation. Ideally, the output of this simulation has exactly the same format as the real data recorded by the detector, and can therefore be put through the same event reconstruction and physics analysis chain, except that here we know what the 'right answer' should be, and so can see how well we are doing.

Since the full chain of detector simulation and event reconstruction is very time-consuming, one often does 'quick and dirty' studies in which these steps are skipped entirely, or at least replaced by very simplified procedures which only take into account the geometric acceptance of the detector and other trivial effects. One may then use the output of the event generator directly in the physics studies.

There are still many holes in our understanding of the full event structure, despite an impressive amount of work and detailed calculations. To put together a generator therefore involves making a choice on what to include, and how to include it. At best, the spread between generators can be used to give some impression of the uncertainties involved. A multitude of approximations will be discussed in the main part of this report, but already here is should be noted that many major approximations are related to the almost complete neglect of the second point above, i.e. of the non-'trivial' higher-order effects. It can therefore only be hoped that the 'trivial' higher order parts give the bulk of the experimental behaviour. By and large, this seems to be the case; for e^+e^- annihilation it even turns out to be a very good approximation.

The necessity to make compromises has one major implication: to write a good event generator is an art, not an exact science. It is therefore essential not to blindly trust the results of any single event generator, but always to make several cross-checks. In addition, with computer programs of tens of thousands of lines, the question is not whether bugs exist, but how many there are, and how critical their positions. Further, an event generator cannot be thought of as all-powerful, or able to give intelligent answers to ill-posed questions; sound judgement and some understanding of a generator are necessary prerequisites for successful use. In spite of these limitations, the event generator approach is the most powerful tool at our disposal if we wish to gain a detailed and realistic understanding of physics at current or future high-energy colliders.

The Origins of the JETSET and PYTHIA Programs

Over the years, many event generators have appeared. Surveys of generators for e^+e^- physics in general and LEP in particular may be found in [Kle89, Sjö89, Kno96, Lön96], for high-energy hadron–hadron (pp) physics in [Ans90, Sjö92, Kno93, LHC00], and for ep physics in [HER92, HER99]. We refer the reader to those for additional details and references. In this particular report, the two closely connected programs JETSET and PYTHIA, now merged under the PYTHIA label, will be described.

JETSET has its roots in the efforts of the Lund group to understand the hadronization process, starting in the late seventies [And83]. The so-called string fragmentation model was developed as an explicit and detailed framework, within which the long-range confinement forces are allowed to distribute the energies and flavours of a parton configuration among a collection of primary hadrons, which subsequently may decay further. This model, known as the Lund string model, or 'Lund' for short, contained a number of specific predictions, which were confirmed by data from PETRA and PEP, whence the model gained a widespread acceptance. The Lund string model is still today the most elaborate and widely used fragmentation model at our disposal. It remains at the heart of the PYTHIA program.

In order to predict the shape of events at PETRA/PEP, and to study the fragmentation process in detail, it was necessary to start out from the partonic configurations that were to fragment. The generation of complete e^+e^- hadronic events was therefore added, originally based on simple γ exchange and first-order QCD matrix elements, later extended to full γ^*/Z^0 exchange with first-order initial-state QED radiation and second-order QCD matrix elements. A number of utility routines were also provided early on, for everything from event listing to jet finding.

By the mid-eighties it was clear that the matrix-element approach had reached the limit of its usefulness, in the sense that it could not fully describe the multijet topologies of the data. (Later on, the use of optimized perturbation theory was to lead to a resurgence of the matrix-element approach, but only for specific applications.) Therefore a parton-shower description was developed [Ben87a] as an alternative to the matrix-element one. The combination of parton showers and string fragmentation has been very successful, and forms the main approach to the description of hadronic Z^0 events.

In recent years, the JETSET part of the code has been a fairly stable product, covering the four main areas of fragmentation, final-state parton showers, e^+e^- event generation and general utilities.

The successes of string fragmentation in e^+e^- made it interesting to try to extend this framework to other processes, and explore possible physics consequences. Therefore a number of other programs were written, which combined a process-specific description of the hard interactions with the general fragmentation framework of JETSET. The PYTHIA program evolved out of early studies on fixed-target proton–proton processes, addressed mainly at issues related to string drawing.

With time, the interest shifted towards higher energies, first to the SPS $p\overline{p}$ collider, and later to the Tevatron, SSC and LHC, in the context of a number of workshops in the USA and Europe. Parton showers were added, for final-state radiation by making use of the JETSET routine, for initial-state one by the development of the concept of 'backwards evolution', specifically for PYTHIA [Sjö85]. Also a framework was developed for minimum-bias and underlying events [Sjö87a].

Another main change was the introduction of an increasing number of hard processes, within the Standard Model and beyond. A special emphasis was put on the search for the Standard Model Higgs, in different mass ranges and in different channels, with due respect to possible background processes.

The bulk of the machinery developed for hard processes actually depended little on the choice of initial state, as long as the appropriate parton distributions were there for the incoming partons and particles. It therefore made sense to extend the program from being only a pp generator to working also for e^+e^- and ep. This process was only completed in 1991, again spurred on by physics workshop activities. Currently PYTHIA should therefore work equally well for a selection of different possible incoming beam particles.

An effort independent of the Lund group activities got going to include supersymmetric event simulation in PYTHIA. This resulted in the SPYTHIA program.

While JETSET was independent of PYTHIA until 1996, their ties had grown much stronger over the years, and the border-line between the two programs had become more and more artificial. It was therefore decided to merge the two, and also include the SPYTHIA extensions, starting from PYTHIA 6.1. The different origins in part still are reflected in this manual, but the strive is towards a seamless merger.

The tasks of including new processes, and of improving the simulation of parton showers and other aspects of already present processes, are never-ending. Work therefore continues apace.

About this Report

As we see, JETSET and PYTHIA started out as very ideologically motivated programs, developed to study specific physics questions in enough detail that explicit predictions could be made for experimental quantities. As it was recognized that experimental imperfections could distort the basic predictions, the programs were made available for general use by experimentalists. It thus became feasible to explore the models in more detail than would otherwise have been possible. As time went by, the emphasis came to shift somewhat, away from the original strong coupling to a specific fragmentation model, towards a description of high-energy multiparticle production processes in general. Correspondingly, the use expanded from being one of just

comparing data with specific model predictions, to one of extensive use for the understanding of detector performance, for the derivation of acceptance correction factors, for the prediction of physics at future high-energy accelerators, and for the design of related detectors.

While the ideology may be less apparent, it is still there, however. This is not something unique to the programs discussed here, but inherent in any event generator, or at least any generator that attempts to go beyond the simple parton level skeleton description of a hard process. Do not accept the myth that everything available in Monte Carlo form represents ages-old common knowledge, tested and true. Ideology is present by commissions or omissions in any number of details. A programs like PYTHIA represents a major amount of original physics research, often on complicated topics where no simple answers are available. As a (potential) program user you must be aware of this, so that you can form your own opinion, not just about what to trust and what not to trust, but also how much to trust a given prediction, i.e. how uncertain it is likely to be. PYTHIA is particularly well endowed in this respect, since a number of publications exist where most of the relevant physics is explained in considerable detail. In fact, the problem may rather be the opposite, to find the relevant information among all the possible places. One main objective of the current report is therefore to collect much of this information in one single place. Not all the material found in specialized papers is reproduced, by a wide margin, but at least enough should be found here to understand the general picture and to know where to go for details.

The current report is therefore intended to update and extend the previous round of published physics descriptions and program manuals [Sjö86, Sjö87, Ben87, Sjö94, Mre97, Sjö01]. Make all references to the most recent published one in [Sjö01]. Further specification could include a statement of the type 'We use PYTHIA version X.xxx'. (If you are a LaTeX fan, you may want to know that the program name in this report has been generated by the command `\textsc{Pythia}`.) Kindly do not refer to PYTHIA as 'unpublished', 'private communication' or 'in preparation': such phrases are incorrect and only create unnecessary confusion.

In addition, remember that many of the individual physics components are documented in separate publications. If some of these contain ideas that are useful to you, there is every reason to cite them. A reasonable selection would vary as a function of the physics you are studying. The criterion for which to pick should be simple: imagine that a Monte Carlo implementation had not been available. Would you then have cited a given paper on the grounds of its physics contents alone? If so, do not punish the extra effort of turning these ideas into publicly available software. (Monte Carlo manuals are good for nothing in the eyes of many theorists, so often only the acceptance of 'mainstream' publications counts.) Here follows a list of some main areas where the PYTHIA programs contain original research:

- The string fragmentation model [And83, And98].
- The string effect [And80].
- Baryon production (diquark/popcorn) [And82, And85, Edé97].
- Small-mass string fragmentation [Nor98].
- Fragmentation of multiparton systems [Sjö84].
- Colour rearrangement [Sjö94a] and Bose-Einstein effects [Lön95].
- Fragmentation effects on α_s determinations [Sjö84a].
- Initial-state parton showers [Sjö85, Miu99].
- Final-state parton showers [Ben87a, Nor01].
- Photon radiation from quarks [Sjö92c]
- Deeply Inelastic Scattering [And81a, Ben88].
- Photoproduction [Sch93a], $\gamma\gamma$ [Sch94a] and $\gamma^* p/\gamma^*\gamma/\gamma^*\gamma^*$ [Fri00] physics.
- Parton distributions of the photon [Sch95, Sch96].
- Colour flow in hard scatterings [Ben84].
- Elastic and diffractive cross sections [Sch94].
- Minijets (multiple parton–parton interactions) [Sjö87a].
- Rapidity gaps [Dok92].

- Jet clustering in $k_\perp$ [Sjö83].

In addition to a physics survey, the current report also contains a complete manual for the program. Such manuals have always been updated and distributed jointly with the programs, but have grown in size with time. A word of warning may therefore be in place. The program description is fairly lengthy, and certainly could not be absorbed in one sitting. This is not even necessary, since all switches and parameters are provided with sensible default values, based on our best understanding (of the physics, and of what you expect to happen if you do not specify any options). As a new user, you can therefore disregard all the fancy options, and just run the program with a minimum ado. Later on, as you gain experience, the options that seem useful can be tried out. No single user is ever likely to find need for more than a fraction of the total number of possibilities available, yet many of them have been added to meet specific user requests.

In some instances, not even this report will provide you with all the information you desire. You may wish to find out about recent versions of the program, know about related software, pick up a few sample main programs to get going, or get hold of related physics papers. Some such material can be found on the PYTHIA web page:
`http://www.thep.lu.se/~torbjorn/Pythia.html`.

Disclaimer

At all times it should be remembered that this is not a commercial product, developed and supported by professionals. Instead it is a 'University World' product, developed by a very few physicists (mainly the current first author) originally for their own needs, and supplied to other physicists on an 'as-is' basis, free of charge. No guarantees are therefore given for the proper functioning of the program, nor for the validity of physics results. In the end, it is always up to you to decide for yourself whether to trust a given result or not. Usually this requires comparison either with analytical results or with results of other programs, or with both. Even this is not necessarily foolproof: for instance, if an error is made in the calculation of a matrix element for a given process, this error will be propagated both into the analytical results based on the original calculation and into all the event generators which subsequently make use of the published formulae. In the end, there is no substitute for a sound physics judgement.

This does not mean that you are all on your own, with a program nobody feels responsible for. Attempts are made to check processes as carefully as possible, to write programs that do not invite unnecessary errors, and to provide a detailed and accurate documentation. All of this while maintaining the full power and flexibility, of course, since the physics must always take precedence in any conflict of interests. If nevertheless any errors or unclarities are found, please do communicate them to e-mail torbjorn@thep.lu.se, or to another person in charge. For instance, all questions on the supersymmetric machinery are better directed to mrenna@physics.ucdavis.edu. Every attempt will be made to solve problems as soon as is reasonably possible, given that this support is by a few persons, who mainly have other responsibilities.

However, in order to make debugging at all possible, we request that any sample code you want to submit as evidence be completely self-contained, and peeled off from all irrelevant aspects. Use simple write statements or the PYTHIA histogramming routines to make your point. Chances are that, if the error cannot be reproduced by fifty lines of code, in a main program linked only to PYTHIA, the problem is sitting elsewhere. Numerous errors have been caused by linking to other (flawed) libraries, e.g. collaboration-specific frameworks for running PYTHIA. Then you should put the blame elsewhere.

Appendix: The Historical Pythia

The 'PYTHIA' label may need some explanation.

The myth tells how Apollon, the God of Wisdom, killed the powerful dragon-like monster Python, close to the village of Delphi in Greece. To commemorate this victory, Apollon founded the Pythic Oracle in Delphi, on the slopes of Mount Parnassos. Here men could come to learn the will of the Gods and the course of the future. The oracle plays an important rôle in many of the other Greek myths, such as those of Heracles and of King Oedipus.

Questions were to be put to the Pythia, the 'Priestess' or 'Prophetess' of the Oracle. In fact, she was a local woman, usually a young maiden, of no particular religious schooling. Seated on a tripod, she inhaled the obnoxious vapours that seeped up through a crevice in the ground. This brought her to a trance-like state, in which she would scream seemingly random words and sounds. It was the task of the professional priests in Delphi to record those utterings and edit them into the official Oracle prophecies, which often took the form of poems in perfect hexameter. In fact, even these edited replies were often less than easy to interpret. The Pythic oracle acquired a reputation for ambiguous answers.

The Oracle existed already at the beginning of the historical era in Greece, and was universally recognized as the foremost religious seat. Individuals and city states came to consult, on everything from cures for childlessness to matters of war. Lavish gifts allowed the temple area to be built and decorated. Many states supplied their own treasury halls, where especially beautiful gifts were on display. Sideshows included the Omphalos, a stone reputedly marking the centre of the Earth, and the Pythic games, second only to the Olympic ones in importance.

Strife inside Greece eventually led to a decline in the power of the Oracle. A serious blow was dealt when the Oracle of Zeus Ammon (see below) declared Alexander the Great to be the son of Zeus. The Pythic Oracle lived on, however, and was only closed by a Roman Imperial decree in 390 AD, at a time when Christianity was ruthlessly destroying any religious opposition. Pythia then had been at the service of man and Gods for a millennium and a half.

The rôle of the Pythic Oracle prophecies on the course of history is nowhere better described than in 'The Histories' by Herodotus [HerBC], the classical and captivating description of the Ancient World at the time of the Great War between Greeks and Persians. Especially famous is the episode with King Croisus of Lydia. Contemplating a war against the upstart Persian Empire, he resolves to ask an oracle what the outcome of a potential battle would be. However, to have some guarantee for the veracity of any prophecy, he decides to send embassies to all the renowned oracles of the known World. The messengers are instructed to inquire the various divinities, on the hundredth day after their departure, what King Croisus is doing at that very moment. From the Pythia the messengers bring back the reply

> *I know the number of grains of sand as well as the expanse of the sea,*
> *And I comprehend the dumb and hear him who does not speak,*
> *There came to my mind the smell of the hard-shelled turtle,*
> *Boiled in copper together with the lamb,*
> *With copper below and copper above.*

The veracity of the Pythia is thus established by the crafty ruler, who had waited until the appointed day, slaughtered a turtle and a lamb, and boiled them together in a copper cauldron with a copper lid. Also the Oracle of Zeus Ammon in the Libyan desert is able to give a correct reply (lost to posterity), while all others fail. King Croisus now sends a second embassy to Delphi, inquiring after the outcome of a battle against the Persians. The Pythia answers

> *If Croisus passes over the Halys he will dissolve a great Empire.*

Taking this to mean he would win, the King collects his army and crosses the border river, only to suffer a crushing defeat and see his Kingdom conquered. When the victorious King Cyrus allows Croisus to send an embassy to upbraid the Oracle, the God Apollon answers through his Prophetess that he has correctly predicted the destruction of a great empire — Croisus' own — and that he cannot be held responsible if people choose to interpret the Oracle answers to their own liking.

The history of the PYTHIA program is neither as long nor as dignified as that of its eponym. However, some points of contact exist. You must be very careful when you formulate the questions: any ambiguities will corrupt the reply you get. And you must be even more careful not to misinterpret the answers; in particular not to pick the interpretation that suits you before considering the alternatives. Finally, even a perfect God has servants that are only human: a priest might mishear the screams of the Pythia and therefore produce an erroneous oracle reply; the current author might unwittingly let a bug free in the program PYTHIA.

2 Physics Overview

In this section we will try to give an overview of the main physics features of PYTHIA, and also to introduce some terminology. The details will be discussed in subsequent sections.

For the description of a typical high-energy event, an event generator should contain a simulation of several physics aspects. If we try to follow the evolution of an event in some semblance of a time order, one may arrange these aspects as follows:

1. Initially two beam particles are coming in towards each other. Normally each particle is characterized by a set of parton distributions, which defines the partonic substructure in terms of flavour composition and energy sharing.
2. One shower initiator parton from each beam starts off a sequence of branchings, such as $q \to qg$, which build up an initial-state shower.
3. One incoming parton from each of the two showers enters the hard process, where then a number of outgoing partons are produced, usually two. It is the nature of this process that determines the main characteristics of the event.
4. The hard process may produce a set of short-lived resonances, like the $Z^0/W^\pm$ gauge bosons, whose decay to normal partons has to be considered in close association with the hard process itself.
5. The outgoing partons may branch, just like the incoming did, to build up final-state showers.
6. In addition to the hard process considered above, further semihard interactions may occur between the other partons of two incoming hadrons.
7. When a shower initiator is taken out of a beam particle, a beam remnant is left behind. This remnant may have an internal structure, and a net colour charge that relates it to the rest of the final state.
8. The QCD confinement mechanism ensures that the outgoing quarks and gluons are not observable, but instead fragment to colour neutral hadrons.
9. Normally the fragmentation mechanism can be seen as occurring in a set of separate colour singlet subsystems, but interconnection effects such as colour rearrangement or Bose–Einstein may complicate the picture.
10. Many of the produced hadrons are unstable and decay further.

Conventionally, only quarks and gluons are counted as partons, while leptons and photons are not. If pushed *ad absurdum* this may lead to some unwieldy terminology. We will therefore, where it does not matter, speak of an electron or a photon in the 'partonic' substructure of an electron, lump branchings $e \to e\gamma$ together with other 'parton shower' branchings such as $q \to qg$, and so on. With this notation, the division into the above seven points applies equally well to an interaction between two leptons, between a lepton and a hadron, and between two hadrons.

In the following sections, we will survey the above ten aspects, not in the same order as given here, but rather in the order in which they appear in the program execution, i.e. starting with the hard process.

2.1 Hard Processes and Parton Distributions

In the original JETSET code, only two hard processes are available. The first and main one is $e^+e^- \to \gamma^*/Z^0 \to q\overline{q}$. Here the '$*$' of γ^* is used to denote that the photon must be off the mass shell. The distinction is of some importance, since a photon on the mass shell cannot decay. Of course also the Z^0 can be off the mass shell, but here the distinction is less relevant (strictly speaking, a Z^0 is always off the mass shell). In the following we may not always use '$*$' consistently, but the rule of thumb is to use a '$*$' only when a process is not kinematically possible for a particle of nominal mass. The quark q in the final state of $e^+e^- \to \gamma^*/Z^0 \to q\overline{q}$ may be u, d, s, c, b or t; the flavour in each event is picked at random, according to the relative

couplings, evaluated at the hadronic c.m. energy. Also the angular distribution of the final $q\overline{q}$ pair is included. No parton-distribution functions are needed.

The other original JETSET process is a routine to generate ggg and γgg final states, as expected in onium 1^{--} decays such as Υ. Given the large top mass, toponium decays weakly much too fast for these processes to be of any interest, so therefore no new applications are expected.

2.1.1 Hard Processes

The current PYTHIA contains a much richer selection, with around 240 different hard processes. These may be classified in many different ways.

One is according to the number of final-state objects: we speak of '$2 \to 1$' processes, '$2 \to 2$' ones, '$2 \to 3$' ones, etc. This aspect is very relevant from a programming point of view: the more particles in the final state, the more complicated the phase space and therefore the whole generation procedure. In fact, PYTHIA is optimized for $2 \to 1$ and $2 \to 2$ processes. There is currently no generic treatment of processes with three or more particles in the final state, but rather a few different machineries, each tailored to the pole structure of a specific class of graphs.

Another classification is according to the physics scenario. This will be the main theme of section 8. The following major groups may be distinguished:

- Hard QCD processes, e.g. $qg \to qg$.
- Soft QCD processes, such as diffractive and elastic scattering, and minimum-bias events. Hidden in this class is also process 96, which is used internally for the merging of soft and hard physics, and for the generation of multiple interactions.
- Heavy-flavour production, both open and hidden, e.g. $gg \to t\overline{t}$ and $gg \to J/\psi g$.
- Prompt-photon production, e.g. $qg \to q\gamma$.
- Photon-induced processes, e.g. $\gamma g \to q\overline{q}$.
- Deeply Inelastic Scattering, e.g. $q\ell \to q\ell$.
- W/Z production, such as the $e^+e^- \to \gamma^*/Z^0$ or $q\overline{q} \to W^+W^-$.
- Standard model Higgs production, where the Higgs is reasonably light and narrow, and can therefore still be considered as a resonance.
- Gauge boson scattering processes, such as $WW \to WW$, when the Standard Model Higgs is so heavy and broad that resonant and non-resonant contributions have to be considered together.
- Non-standard Higgs particle production, within the framework of a two-Higgs-doublet scenario with three neutral (h^0, H^0 and A^0) and two charged ($H^\pm$) Higgs states. Normally associated with SUSY (see below), but does not have to be.
- Production of new gauge bosons, such as a Z', W' and R (a horizontal boson, coupling between generations).
- Technicolor production, as an alternative scenario to the standard picture of electroweak symmetry breaking by a fundamental Higgs.
- Compositeness is a possibility not only in the Higgs sector, but may also apply to fermions, e.g. giving d^* and u^* production. At energies below the threshold for new particle production, contact interactions may still modify the standard behaviour.
- Left–right symmetric models give rise to doubly charged Higgs states, in fact one set belonging to the left and one to the right **SU(2)** gauge group. Decays involve right-handed W's and neutrinos.
- Leptoquark (L_Q) production is encountered in some beyond-the-standard-model scenarios.
- Supersymmetry (SUSY) is probably the favourite scenario for physics beyond the standard model. A rich set of processes are allowed, even if one obeys *R*-parity conservation. The supersymmetric machinery and process selection is inherited from SPYTHIA [Mre97],

however with many improvements in the event generation chain. Many different SUSY scenarios have been proposed, and the program is flexible enough to allow input from several of these, in addition to the ones provided internally.

- The possibility of extra dimensions at low energies has been a topic of much study in recent years, but has still not settled down to some standard scenarios. Its inclusion into PYTHIA is also only in a very first stage.

This is by no means a survey of all interesting physics. Also, within the scenarios studied, not all contributing graphs have always been included, but only the more important and/or more interesting ones. In many cases, various approximations are involved in the matrix elements coded.

2.1.2 Resonance Decays

As we noted above, the bulk of the processes above are of the $2 \to 2$ kind, with very few leading to the production of more than two final-state particles. This may be seen as a major limitation, and indeed is so at times. However, often one can come quite far with only one or two particles in the final state, since showers will add the required extra activity. The classification may also be misleading at times, since an *s*-channel resonance is considered as a single particle, even if it is assumed always to decay into two final-state particles. Thus the process $e^+e^- \to W^+W^- \to q_1\overline{q}_1' \, q_2\overline{q}_2'$ is classified as $2 \to 2$, although the decay treatment of the W pair includes the full $2 \to 4$ matrix elements (in the doubly resonant approximation, i.e. excluding interference with non-WW four-fermion graphs).

Particles which admit this close connection between the hard process and the subsequent evolution are collectively called resonances in this manual. It includes all particles in mass above the b quark system, such as t, Z^0, $W^\pm$, h^0, supersymmetric particles, and many more. Typically their decays are given by electroweak physics, or physics beyond the Standard Model. What characterizes a (PYTHIA) resonance is that partial widths and branching ratios can be calculated dynamically, as a function of the actual mass of a particle. Therefore not only do branching ratios change between an h^0 of nominal mass 100 GeV and one of 200 GeV, but also for a Higgs of nominal mass 200 GeV, the branching ratios would change between an actual mass of 190 GeV and 210 GeV, say. This is particularly relevant for reasonably broad resonances, and in threshold regions. For an approach like this to work, it is clearly necessary to have perturbative expressions available for all partial widths.

Decay chains can become quite lengthy, e.g. for supersymmetric processes, but follow a straight perturbative pattern. If the simulation is restricted to only some set of decays, the corresponding cross section reduction can easily be calculated. (Except in some rare cases where a nontrivial threshold behaviour could complicate matters.) It is therefore standard in PYTHIA to quote cross sections with such reductions already included. Note that the branching ratios of a particle is affected also by restrictions made in the secondary or subsequent decays. For instance, the branching ratio of $h^0 \to W^+W^-$, relative to $h^0 \to Z^0Z^0$ and other channels, is changed if the allowed W decays are restricted.

The decay products of resonances are typically quarks, leptons, or other resonances, e.g. $W \to q\overline{q}'$ or $h^0 \to W^+W^-$. Ordinary hadrons are not produced in these decays, but only in subsequent hadronization steps. In decays to quarks, parton showers are automatically added to give a more realistic multijet structure, and one may also allow photon emission off leptons. If the decay products in turn are resonances, further decays are necessary. Often spin information is available in resonance decay matrix elements. This means that the angular orientations in the two decays of a W^+W^- pair are properly correlated. In other cases, the information is not available, and then resonances decay isotropically.

Of course, the above 'resonance' terminology is arbitrary. A ρ, for instance, could also be called a resonance, but not in the above sense. The width is not perturbatively calculable, it decays to hadrons by strong interactions, and so on. From a practical point of view, the main dividing line is that the values of — or a change in — branching ratios cannot affect the cross

section of a process. For instance, if one wanted to consider the decay $Z^0 \to c\bar{c}$, with a D meson producing a lepton, not only would there then be the problem of different leptonic branching ratios for different D's (which means that fragmentation and decay treatments would no longer decouple), but also that of additional $c\bar{c}$ pair production in parton-shower evolution, at a rate that is unknown beforehand. In practice, it is therefore next to impossible to force D decay modes in a consistent manner.

2.1.3 Parton Distributions

The cross section for a process $ij \to k$ is given by

$$\sigma_{ij\to k} = \int dx_1 \int dx_2\, f_i^1(x_1)\, f_j^2(x_2)\, \hat{\sigma}_{ij\to k}\,. \tag{1}$$

Here $\hat{\sigma}$ is the cross section for the hard partonic process, as codified in the matrix elements for each specific process. For processes with many particles in the final state it would be replaced by an integral over the allowed final-state phase space. The $f_i^a(x)$ are the parton-distribution functions, which describe the probability to find a parton i inside beam particle a, with parton i carrying a fraction x of the total a momentum. Actually, parton distributions also depend on some momentum scale Q^2 that characterizes the hard process.

Parton distributions are most familiar for hadrons, such as the proton. Hadrons are inherently composite objects, made up of quarks and gluons. Since we do not understand QCD, a derivation from first principles of hadron parton distributions does not yet exist, although some progress is being made in lattice QCD studies. It is therefore necessary to rely on parameterizations, where experimental data are used in conjunction with the evolution equations for the Q^2 dependence, to pin down the parton distributions. Several different groups have therefore produced their own fits, based on slightly different sets of data, and with some variation in the theoretical assumptions.

Also for fundamental particles, such as the electron, is it convenient to introduce parton distributions. The function $f_e^e(x)$ thus parameterizes the probability that the electron that takes part in the hard process retains a fraction x of the original energy, the rest being radiated (into photons) in the initial state. Of course, such radiation could equally well be made part of the hard interaction, but the parton-distribution approach usually is much more convenient. If need be, a description with fundamental electrons is recovered for the choice $f_e^e(x, Q^2) = \delta(x-1)$. Note that, contrary to the proton case, electron parton distributions are calculable from first principles, and reduce to the δ function above for $Q^2 \to 0$.

The electron may also contain photons, and the photon may in its turn contain quarks and gluons. The internal structure of the photon is a bit of a problem, since the photon contains a point-like part, which is perturbatively calculable, and a resolved part (with further subdivisions), which is not. Normally, the photon parton distributions are therefore parameterized, just as the hadron ones. Since the electron ultimately contains quarks and gluons, hard QCD processes like $qg \to qg$ therefore not only appear in pp collisions, but also in ep ones ('resolved photoproduction') and in e^+e^- ones ('doubly resolved 2γ events'). The parton distribution function approach here makes it much easier to reuse one and the same hard process in different contexts.

There is also another kind of possible generalization. The two processes $q\bar{q} \to \gamma^*/Z^0$, studied in hadron colliders, and $e^+e^- \to \gamma^*/Z^0$, studied in e^+e^- colliders, are really special cases of a common process, $f\bar{f} \to \gamma^*/Z^0$, where f denotes a fundamental fermion, i.e. a quark, lepton or neutrino. The whole structure is therefore only coded once, and then slightly different couplings and colour prefactors are used, depending on the initial state considered. Usually the interesting cross section is a sum over several different initial states, e.g. $u\bar{u} \to \gamma^*/Z^0$ and $d\bar{d} \to \gamma^*/Z^0$ in a hadron collider. This kind of summation is always implicitly done, even when not explicitly mentioned in the text.

2.2 Initial- and Final-State Radiation

In every process that contains coloured and/or charged objects in the initial or final state, gluon and/or photon radiation may give large corrections to the overall topology of events. Starting from a basic $2 \to 2$ process, this kind of corrections will generate $2 \to 3$, $2 \to 4$, and so on, final-state topologies. As the available energies are increased, hard emission of this kind is increasingly important, relative to fragmentation, in determining the event structure.

Two traditional approaches exist to the modelling of perturbative corrections. One is the matrix-element method, in which Feynman diagrams are calculated, order by order. In principle, this is the correct approach, which takes into account exact kinematics, and the full interference and helicity structure. The only problem is that calculations become increasingly difficult in higher orders, in particular for the loop graphs. Only in exceptional cases have therefore more than one loop been calculated in full, and often we do not have any loop corrections at all at our disposal. On the other hand, we have indirect but strong evidence that, in fact, the emission of multiple soft gluons plays a significant rôle in building up the event structure, e.g. at LEP, and this sets a limit to the applicability of matrix elements. Since the phase space available for gluon emission increases with the available energy, the matrix-element approach becomes less relevant for the full structure of events at higher energies. However, the perturbative expansion is better behaved at higher energy scales, owing to the running of α_s. As a consequence, inclusive measurements, e.g. of the rate of well-separated jets, should yield more reliable results at high energies.

The second possible approach is the parton-shower one. Here an arbitrary number of branchings of one parton into two (or more) may be combined, to yield a description of multijet events, with no explicit upper limit on the number of partons involved. This is possible since the full matrix-element expressions are not used, but only approximations derived by simplifying the kinematics, and the interference and helicity structure. Parton showers are therefore expected to give a good description of the substructure of jets, but in principle the shower approach has limited predictive power for the rate of well-separated jets (i.e. the 2/3/4/5-jet composition). In practice, shower programs may be matched to first-order matrix elements to describe the hard-gluon emission region reasonably well, in particular for the e^+e^- annihilation process. Nevertheless, the shower description is not optimal for absolute α_s determinations.

Thus the two approaches are complementary in many respects, and both have found use. However, because of its simplicity and flexibility, the parton-shower option is generally the first choice, while the matrix elements one is mainly used for α_s determinations, angular distribution of jets, triple-gluon vertex studies, and other specialized studies. Obviously, the ultimate goal would be to have an approach where the best aspects of the two worlds are harmoniously married. This is currently a topic of quite some study.

2.2.1 Matrix elements

Matrix elements are especially made use of in the older JETSET-originated implementation of the process $e^+e^- \to \gamma^*/Z^0 \to q\bar{q}$.

For initial-state QED radiation, a first order (un-exponentiated) description has been adopted. This means that events are subdivided into two classes, those where a photon is radiated above some minimum energy, and those without such a photon. In the latter class, the soft and virtual corrections have been lumped together to give a total event rate that is correct up to one loop. This approach worked fine at PETRA/PEP energies, but does not do so well for the Z^0 line shape, i.e. in regions where the cross section is rapidly varying and high precision is strived for.

For final-state QCD radiation, several options are available. The default is the parton-shower one (see below), but the matrix-elements options are also frequently used. In the definition of 3- or 4-jet events, a cut is introduced whereby it is required that any two partons have an invariant mass bigger than some fraction of the c.m. energy. 3-jet events which do not fulfil

this requirement are lumped with the 2-jet ones. The first-order matrix-element option, which only contains 3- and 2-jet events therefore involves no ambiguities. In second order, where also 4-jets have to be considered, a main issue is what to do with 4-jet events that fail the cuts. Depending on the choice of recombination scheme, whereby the two nearby partons are joined into one, different 3-jet events are produced. Therefore the second-order differential 3-jet rate has been the subject of some controversy, and the program actually contains two different implementations.

By contrast, the normal PYTHIA event generation machinery does not contain any full higher-order matrix elements, with loop contributions included. There are several cases where higher-order matrix elements are included at the Born level. Consider the case of resonance production at a hadron collider, e.g. of a W, which is contained in the lowest-order process $q\overline{q}' \to W$. In an inclusive description, additional jets recoiling against the W may be generated by parton showers. PYTHIA also contains the two first-order processes $qg \to Wq'$ and $q\overline{q}' \to Wg$. The cross sections for these processes are divergent when the $p_\perp \to 0$. In this region a correct treatment would therefore have to take into account loop corrections, which are not available in PYTHIA.

Even without having these accessible, we know approximately what the outcome should be. The virtual corrections have to cancel the $p_\perp \to 0$ singularities of the real emission. The total cross section of W production therefore receives finite $O(\alpha_s)$ corrections to the lowest-order answer. These corrections can often be neglected to first approximation, except when high precision is required. As for the shape of the W $p_\perp$ spectrum, the large cross section for low-$p_\perp$ emission has to be interpreted as allowing more than one emission to take place. A resummation procedure is therefore necessary to have matrix element make sense at small $p_\perp$. The outcome is a cross section below the naive one, with a finite behaviour in the $p_\perp \to 0$ limit.

Depending on the physics application, one could then use PYTHIA in one of two ways. In an inclusive description, which is dominated by the region of reasonably small $p_\perp$, the preferred option is lowest-order matrix elements combined with parton showers, which actually is one way of achieving the required resummation. For W production as background to some other process, say, only the large-$p_\perp$ tail might be of interest. Then the shower approach may be inefficient, since only few events will end up in the interesting region, while the matrix-element alternative allows reasonable cuts to be inserted from the beginning of the generation procedure. (One would probably still want to add showers to describe additional softer radiation, at the cost of some smearing of the original cuts.) Furthermore, and not less importantly, the matrix elements should give a more precise prediction of the high-$p_\perp$ event rate than the approximate shower procedure.

In the particular case considered here, that of W production, and a few similar processes, actually the shower has been improved by a matching to first-order matrix elements, thus giving a decent description over the whole $p_\perp$ range. This does not provide the first-order corrections to the total W production rate, however, nor the possibility to select only a high-$p_\perp$ tail of events.

2.2.2 Parton showers

The separation of radiation into initial- and final-state showers is arbitrary, but very convenient. There are also situations where it is appropriate: for instance, the process $e^+e^- \to Z^0 \to q\overline{q}$ only contains final-state QCD radiation (QED radiation, however, is possible both in the initial and final state), while $q\overline{q} \to Z^0 \to e^+e^-$ only contains initial-state QCD one. Similarly, the distinction of emission as coming either from the q or from the $\overline{q}$ is arbitrary. In general, the assignment of radiation to a given mother parton is a good approximation for an emission close to the direction of motion of that parton, but not for the wide-angle emission in between two jets, where interference terms are expected to be important.

In both initial- and final-state showers, the structure is given in terms of branchings $a \to bc$, specifically $e \to e\gamma$, $q \to qg$, $q \to q\gamma$, $g \to gg$, and $g \to q\overline{q}$. (Further branchings, like $\gamma \to e^+e^-$

and $\gamma \to q\overline{q}$, could also have been added, but have not yet been of interest.) Each of these processes is characterized by a splitting kernel $P_{a\to bc}(z)$. The branching rate is proportional to the integral $\int P_{a\to bc}(z)\,dz$. The z value picked for a branching describes the energy sharing, with daughter b taking a fraction z and daughter c the remaining $1-z$ of the mother energy. Once formed, the daughters b and c may in turn branch, and so on.

Each parton is characterized by some virtuality scale Q^2, which gives an approximate sense of time ordering to the cascade. In the initial-state shower, Q^2 values are gradually increasing as the hard scattering is approached, while Q^2 is decreasing in the final-state showers. Shower evolution is cut off at some lower scale Q_0, typically around 1 GeV for QCD branchings. From above, a maximum scale Q_{max} is introduced, where the showers are matched to the hard interaction itself. The relation between Q_{max} and the kinematics of the hard scattering is uncertain, and the choice made can strongly affect the amount of well-separated jets.

Despite a number of common traits, the initial- and final-state radiation machineries are in fact quite different, and are described separately below.

Final-state showers are time-like, i.e. partons have $m^2 = E^2 - \mathbf{p}^2 \geq 0$. The evolution variable Q^2 of the cascade is therefore in PYTHIA associated with the m^2 of the branching parton, but this choice is not unique. Starting from Q^2_{max}, an original parton is evolved downwards in Q^2 until a branching occurs. The selected Q^2 value defines the mass of the branching parton, and the z of the splitting kernel the parton energy division between its daughters. These daughters may now, in turn, evolve downwards, in this case with maximum virtuality already defined by kinematics, and so on down to the Q_0 cut-off.

In QCD showers, corrections to the leading-log picture, so-called coherence effects, lead to an ordering of subsequent emissions in terms of decreasing angles. This does not follow automatically from the mass-ordering constraint, but is implemented as an additional requirement on allowed emissions. Photon emission is not affected by angular ordering. It is also possible to obtain non-trivial correlations between azimuthal angles in the various branchings, some of which are implemented as options. Finally, the theoretical analysis strongly suggests the scale choice $\alpha_s = \alpha_s(p_\perp^2) = \alpha_s(z(1-z)m^2)$, and this is the default in the program.

The final-state radiation machinery is normally applied in the c.m. frame of the hard scattering or a decaying resonance. The total energy and momentum of that subsystem is preserved, as is the direction of the outgoing partons (in their common rest frame), where applicable.

In contrast to final-state showers, initial-state ones are space-like. This means that, in the sequence of branchings $a \to bc$ that lead up from the shower initiator to the hard interaction, particles a and b have $m^2 = E^2 - \mathbf{p}^2 < 0$. The 'side branch' particle c, which does not participate in the hard scattering, may be on the mass shell, or have a time-like virtuality. In the latter case a time-like shower will evolve off it, rather like the final-state radiation described above. To first approximation, the evolution of the space-like main branch is characterized by the evolution variable $Q^2 = -m^2$, which is required to be strictly increasing along the shower, i.e. $Q_b^2 > Q_a^2$. Corrections to this picture have been calculated, but are basically absent in PYTHIA.

Initial-state radiation is handled within the backwards evolution scheme. In this approach, the choice of the hard scattering is based on the use of evolved parton distributions, which means that the inclusive effects of initial-state radiation are already included. What remains is therefore to construct the exclusive showers. This is done starting from the two incoming partons at the hard interaction, tracing the showers 'backwards in time', back to the two shower initiators. In other words, given a parton b, one tries to find the parton a that branched into b. The evolution in the Monte Carlo is therefore in terms of a sequence of decreasing space-like virtualities Q^2 and increasing momentum fractions x. Branchings on the two sides are interleaved in a common sequence of decreasing Q^2 values.

In the above formalism, there is no real distinction between gluon and photon emission. Some of the details actually do differ, as will be explained in the full description.

The initial- and final-state radiation shifts around the kinematics of the original hard interaction. In Deeply Inelastic Scattering, this means that the x and Q^2 values that can be derived from the momentum of the scattered lepton do not automatically agree with the values originally

picked. In high-$p_\perp$ processes, it means that one no longer has two jets with opposite and compensating $p_\perp$, but more complicated topologies. Effects of any original kinematics selection cuts are therefore smeared out, an unfortunate side-effect of the parton-shower approach.

2.3 Beam Remnants and Multiple Interactions

In a hadron–hadron collision, the initial-state radiation algorithm reconstructs one shower initiator in each beam. This initiator only takes some fraction of the total beam energy, leaving behind a beam remnant which takes the rest. For a proton beam, a u quark initiator would leave behind a ud diquark beam remnant, with an antitriplet colour charge. The remnant is therefore colour-connected to the hard interaction, and forms part of the same fragmenting system. It is further customary to assign a primordial transverse momentum to the shower initiator, to take into account the motion of quarks inside the original hadron, at least as required by the uncertainty principle by the proton size, probably augmented by unresolved (i.e. not simulated) soft shower activity. This primordial $k_\perp$ is selected according to some suitable distribution, and the recoil is assumed to be taken up by the beam remnant.

Often the remnant is more complicated, e.g. a gluon initiator would leave behind a uud proton remnant system in a colour octet state, which can conveniently be subdivided into a colour triplet quark and a colour antitriplet diquark, each of which are colour-connected to the hard interaction. The energy sharing between these two remnant objects, and their relative transverse momentum, introduces additional degrees of freedom, which are not understood from first principles.

Naïvely, one would expect an ep event to have only one beam remnant, and an e^+e^- event none. This is not always correct, e.g. a $\gamma\gamma \to q\overline{q}$ interaction in an e^+e^- event would leave behind the e^+ and e^- as beam remnants, and a $q\overline{q} \to gg$ interaction in resolved photoproduction in an e^+e^- event would leave behind one $e^\pm$ and one q or $\overline{q}$ in each remnant. Corresponding complications occur for photoproduction in ep events.

There is another source of beam remnants. If parton distributions are used to resolve an electron inside an electron, some of the original energy is not used in the hard interaction, but is rather associated with initial-state photon radiation. The initial-state shower is in principle intended to trace this evolution and reconstruct the original electron before any radiation at all took place. However, because of cut-off procedures, some small amount may be left unaccounted for. Alternatively, you may have chosen to switch off initial-state radiation altogether, but still preserved the resolved electron parton distributions. In either case the remaining energy is given to a single photon of vanishing transverse momentum, which is then considered in the same spirit as 'true' beam remnants.

So far we have assumed that each event only contains one hard interaction, i.e. that each incoming particle has only one parton which takes part in hard processes, and that all other constituents sail through unaffected. This is appropriate in e^+e^- or ep events, but not necessarily so in hadron–hadron collisions. Here each of the beam particles contains a multitude of partons, and so the probability for several interactions in one and the same event need not be negligible. In principle these additional interactions could arise because one single parton from one beam scatters against several different partons from the other beam, or because several partons from each beam take place in separate $2 \to 2$ scatterings. Both are expected, but combinatorics should favour the latter, which is the mechanism considered in PYTHIA.

The dominant $2 \to 2$ QCD cross sections are divergent for $p_\perp \to 0$, and drop rapidly for larger $p_\perp$. Probably the lowest-order perturbative cross sections will be regularized at small $p_\perp$ by colour coherence effects: an exchanged gluon of small $p_\perp$ has a large transverse wave function and can therefore not resolve the individual colour charges of the two incoming hadrons; it will only couple to an average colour charge that vanishes in the limit $p_\perp \to 0$. In the program, some effective $p_{\perp\mathrm{min}}$ scale is therefore introduced, below which the perturbative cross section is either assumed completely vanishing or at least strongly damped. Phenomenologically, $p_{\perp\mathrm{min}}$ comes out to be a number of the order of 1.5–2.0 GeV.

In a typical 'minimum-bias' event one therefore expects to find one or a few scatterings at scales around or a bit above $p_{\perp\mathrm{min}}$, while a high-$p_\perp$ event also may have additional scatterings at the $p_{\perp\mathrm{min}}$ scale. The probability to have several high-$p_\perp$ scatterings in the same event is small, since the cross section drops so rapidly with $p_\perp$.

The understanding of multiple interaction is still very primitive. PYTHIA therefore contains several different options, with a fairly simple one as default. The options differ in particular on the issue of the 'pedestal' effect: is there an increased probability or not for additional interactions in an event which is known to contain a hard scattering, compared with one that contains no hard interactions?

2.4 Hadronization

QCD perturbation theory, formulated in terms of quarks and gluons, is valid at short distances. At long distances, QCD becomes strongly interacting and perturbation theory breaks down. In this confinement regime, the coloured partons are transformed into colourless hadrons, a process called either hadronization or fragmentation. In this paper we reserve the former term for the combination of fragmentation and the subsequent decay of unstable particles.

The fragmentation process has yet to be understood from first principles, starting from the QCD Lagrangian. This has left the way clear for the development of a number of different phenomenological models. Three main schools are usually distinguished, string fragmentation (SF), independent fragmentation (IF) and cluster fragmentation (CF), but many variants and hybrids exist. Being models, none of them can lay claims to being 'correct', although some may be better founded than others. The best that can be aimed for is internal consistency, a good representation of existing data, and a predictive power for properties not yet studied or results at higher energies.

2.4.1 String Fragmentation

The original JETSET program is intimately connected with string fragmentation, in the form of the time-honoured 'Lund model'. This is the default for all PYTHIA applications, but independent fragmentation options also exist, for applications where one wishes to study the importance of string effects.

All current models are of a probabilistic and iterative nature. This means that the fragmentation process as a whole is described in terms of one or a few simple underlying branchings, of the type jet $\to$ hadron + remainder-jet, string $\to$ hadron + remainder-string, and so on. At each branching, probabilistic rules are given for the production of new flavours, and for the sharing of energy and momentum between the products.

To understand fragmentation models, it is useful to start with the simplest possible system, a colour-singlet $\mathrm{q\overline{q}}$ 2-jet event, as produced in $\mathrm{e^+e^-}$ annihilation. Here lattice QCD studies lend support to a linear confinement picture (in the absence of dynamical quarks), i.e. the energy stored in the colour dipole field between a charge and an anticharge increases linearly with the separation between the charges, if the short-distance Coulomb term is neglected. This is quite different from the behaviour in QED, and is related to the presence of a triple-gluon vertex in QCD. The details are not yet well understood, however.

The assumption of linear confinement provides the starting point for the string model. As the q and $\overline{\mathrm{q}}$ partons move apart from their common production vertex, the physical picture is that of a colour flux tube (or maybe a colour vortex line) being stretched between the q and the $\overline{\mathrm{q}}$. The transverse dimensions of the tube are of typical hadronic sizes, roughly 1 fm. If the tube is assumed to be uniform along its length, this automatically leads to a confinement picture with a linearly rising potential. In order to obtain a Lorentz covariant and causal description of the energy flow due to this linear confinement, the most straightforward way is to use the dynamics of the massless relativistic string with no transverse degrees of freedom. The mathematical, one-dimensional string can be thought of as parameterizing the position of the axis of a cylindrically

symmetric flux tube. From hadron spectroscopy, the string constant, i.e. the amount of energy per unit length, is deduced to be $\kappa \approx 1$ GeV/fm. The expression 'massless' relativistic string is somewhat of a misnomer: κ effectively corresponds to a 'mass density' along the string.

Let us now turn to the fragmentation process. As the q and $\overline{q}$ move apart, the potential energy stored in the string increases, and the string may break by the production of a new $q'\overline{q}'$ pair, so that the system splits into two colour-singlet systems $q\overline{q}'$ and $q'\overline{q}$. If the invariant mass of either of these string pieces is large enough, further breaks may occur. In the Lund string model, the string break-up process is assumed to proceed until only on-mass-shell hadrons remain, each hadron corresponding to a small piece of string with a quark in one end and an antiquark in the other.

In order to generate the quark–antiquark pairs $q'\overline{q}'$ which lead to string break-ups, the Lund model invokes the idea of quantum mechanical tunnelling. This leads to a flavour-independent Gaussian spectrum for the $p_\perp$ of $q'\overline{q}'$ pairs. Since the string is assumed to have no transverse excitations, this $p_\perp$ is locally compensated between the quark and the antiquark of the pair. The total $p_\perp$ of a hadron is made up out of the $p_\perp$ contributions from the quark and antiquark that together form the hadron. Some contribution of very soft perturbative gluon emission may also effectively be included in this description.

The tunnelling picture also implies a suppression of heavy-quark production, $u : d : s : c \approx 1 : 1 : 0.3 : 10^{-11}$. Charm and heavier quarks hence are not expected to be produced in the soft fragmentation, but only in perturbative parton-shower branchings $g \to q\overline{q}$.

When the quark and antiquark from two adjacent string breaks are combined to form a meson, it is necessary to invoke an algorithm to choose between the different allowed possibilities, notably between pseudoscalar and vector mesons. Here the string model is not particularly predictive. Qualitatively one expects a $1 : 3$ ratio, from counting the number of spin states, multiplied by some wave-function normalization factor, which should disfavour heavier states.

A tunnelling mechanism can also be used to explain the production of baryons. This is still a poorly understood area. In the simplest possible approach, a diquark in a colour antitriplet state is just treated like an ordinary antiquark, such that a string can break either by quark–antiquark or antidiquark–diquark pair production. A more complex scenario is the 'popcorn' one, where diquarks as such do not exist, but rather quark–antiquark pairs are produced one after the other. This latter picture gives a less strong correlation in flavour and momentum space between the baryon and the antibaryon of a pair.

In general, the different string breaks are causally disconnected. This means that it is possible to describe the breaks in any convenient order, e.g. from the quark end inwards. One therefore is led to write down an iterative scheme for the fragmentation, as follows. Assume an initial quark q moving out along the $+z$ axis, with the antiquark going out in the opposite direction. By the production of a $q_1\overline{q}_1$ pair, a meson with flavour content $q\overline{q}_1$ is produced, leaving behind an unpaired quark q_1. A second pair $q_2\overline{q}_2$ may now be produced, to give a new meson with flavours $q_1\overline{q}_2$, etc. At each step the produced hadron takes some fraction of the available energy and momentum. This process may be iterated until all energy is used up, with some modifications close to the $\overline{q}$ end of the string in order to make total energy and momentum come out right.

The choice of starting the fragmentation from the quark end is arbitrary, however. A fragmentation process described in terms of starting at the $\overline{q}$ end of the system and fragmenting towards the q end should be equivalent. This 'left–right' symmetry constrains the allowed shape of the fragmentation function $f(z)$, where z is the fraction of the remaining light-cone momentum $E \pm p_z$ (+ for the q jet, $-$ for the $\overline{q}$ one) taken by each new particle. The resulting 'Lund symmetric fragmentation function' has two free parameters, which are determined from data.

If several partons are moving apart from a common origin, the details of the string drawing become more complicated. For a $q\overline{q}g$ event, a string is stretched from the q end via the g to the $\overline{q}$ end, i.e. the gluon is a kink on the string, carrying energy and momentum. As a consequence, the gluon has two string pieces attached, and the ratio of gluon to quark string

force is 2, a number which can be compared with the ratio of colour charge Casimir operators, $N_C/C_F = 2/(1-1/N_C^2) = 9/4$. In this, as in other respects, the string model can be viewed as a variant of QCD where the number of colours N_C is not 3 but infinite. Note that the factor 2 above does not depend on the kinematical configuration: a smaller opening angle between two partons corresponds to a smaller string length drawn out per unit time, but also to an increased transverse velocity of the string piece, which gives an exactly compensating boost factor in the energy density per unit string length.

The $q\overline{q}g$ string will fragment along its length. To first approximation this means that there is one fragmenting string piece between q and g and a second one between g and $\overline{q}$. One hadron is straddling both string pieces, i.e. sitting around the gluon corner. The rest of the particles are produced as in two simple $q\overline{q}$ strings, but strings boosted with respect to the overall c.m. frame. When considered in detail, the string motion and fragmentation is more complicated, with the appearance of additional string regions during the time evolution of the system. These corrections are especially important for soft and collinear gluons, since they provide a smooth transition between events where such radiation took place and events where it did not. Therefore the string fragmentation scheme is 'infrared safe' with respect to soft or collinear gluon emission.

For events that involve many partons, there may be several possible topologies for their ordering along the string. An example would be a $q\overline{q}g_1g_2$ (the gluon indices are here used to label two different gluon-momentum vectors), where the string can connect the partons in either of the sequences $q - g_1 - g_2 - \overline{q}$ and $q - g_2 - g_1 - \overline{q}$. The matrix elements that are calculable in perturbation theory contain interference terms between these two possibilities, which means that the colour flow is not always well-defined. Fortunately, the interference terms are down in magnitude by a factor $1/N_C^2$, where $N_C = 3$ is the number of colours, so approximate recipes can be found. In the leading log shower description, on the other hand, the rules for the colour flow are well-defined.

A final comment: in the argumentation for the importance of colour flows there is a tacit assumption that soft-gluon exchanges between partons will not normally mess up the original colour assignment. Colour rearrangement models provide toy scenarios wherein deviations from this rule could be studied. Of particular interest has been the process $e^+e^- \to W^+W^- \to q_1\overline{q}_2q_3\overline{q}_4$, where the original singlets $q_1\overline{q}_2$ and $q_3\overline{q}_4$ could be rearranged to $q_1\overline{q}_4$ and $q_3\overline{q}_2$. So far, there are no experimental evidence for dramatic effects of this kind, but the more realistic models predict effects sufficiently small that these have not been ruled out. Another example of nontrivial effects is that of Bose–Einstein correlations between identical final-state particles, which reflect the true quantum nature of the hadronization process.

2.4.2 Decays

A large fraction of the particles produced by fragmentation are unstable and subsequently decay into the observable stable (or almost stable) ones. It is therefore important to include all particles with their proper mass distributions and decay properties. Although involving little deep physics, this is less trivial than it may sound: while a lot of experimental information is available, there is also very much that is missing. For charm mesons, it is necessary to put together measured exclusive branching ratios with some inclusive multiplicity distributions to obtain a consistent and reasonably complete set of decay channels, a rather delicate task. For bottom even less is known, and for some B baryons only a rather simple phase-space type of generator has been used for hadronic decays.

Normally it is assumed that decay products are distributed according to phase space, i.e. that there is no dynamics involved in their relative distribution. However, in many cases additional assumptions are necessary, e.g. for semileptonic decays of charm and bottom hadrons one needs to include the proper weak matrix elements. Particles may also be produced polarized and impart a non-isotropic distribution to their decay products. Many of these effects are not at all treated in the program. In fact, spin information is not at all carried along, but has to be

reconstructed explicitly when needed.

This normal decay treatment makes use of a set of tables where branching ratios and decay modes are stored. It encompasses all hadrons made out of d, u, s, c and b quarks, and also the leptons. The decay products are hadrons, leptons and photons. Some $b\bar{b}$ states are sufficiently heavy that they are allowed to decay to partonic states, like $\Upsilon \to ggg$, which subsequently fragment, but these are exceptions.

You may at will change the particle properties, decay channels or branching ratios of the above particles. There is no censorship what is allowed or not allowed, beyond energy–momentum and (electrical and colour) charge conservation. There is also no impact e.g. on the cross section of processes, since there is no way of knowing e.g. if the restriction to one specific decay of a particle is because that decay is of particular interest to us, or because recent measurement have shown that this indeed is the only channel. Furthermore, the number of particles produced of each species in the hadronization process is not known beforehand, and so cannot be used to correctly bias the preceding steps of the generation chain. All of this contrasts with the class of 'resonances' described above, in section 2.1.2.

3 Program Overview

This section contains a diverse collection of information. The first part is an overview of previous JETSET and PYTHIA versions. The second gives instructions for installation of the program and describes its philosophy: how it is constructed and how it is supposed to be used. It also contains some information on how to read this manual. The third and final part contains several examples of pieces of code or short programs, to illustrate the general style of program usage. This last part is mainly intended as an introduction for completely new users, and can be skipped by more experienced ones.

The combined PYTHIA package is completely self-contained. Interfaces to externally defined subprocesses, parton-distribution function libraries, τ decay libraries, and a time routine are provided, however, plus a few other optional interfaces.

Many programs written by other persons make use of PYTHIA, especially the string fragmentation machinery. It is not the intention to give a complete list here. A majority of these programs are specific to given collaborations, and therefore not publicly distributed. Below we give a list of a few public programs from the 'Lund group', which may have a somewhat wider application. None of them are supported by the PYTHIA author team, so any requests should be directed to the persons mentioned.

- ARIADNE is a generator for dipole emission, written mainly by L. Lönnblad [Pet88].
- AROMA is a generator for heavy-flavour processes in leptoproduction, written by G. Ingelman and G. Schuler [Ing88].
- FRITIOF is a generator for hadron–hadron, hadron–nucleus and nucleus–nucleus collisions [Nil87].
- LEPTO is a leptoproduction event generator, written mainly by G. Ingelman [Ing80]. It can generate parton configurations in Deeply Inelastic Scattering according to a number of possibilities.
- POMPYT is a generator for pomeron interactions written by G. Ingelman and collaborators [Bru96].

One should also note that a version of PYTHIA has been modified to include the effects of longitudinally polarized incoming protons. This is the work of St. Güllenstern et al. [Gül93].

3.1 Update History

For the record, in Tables 1 and 2 we list the official main versions of JETSET and PYTHIA, respectively, with some brief comments.

All versions preceding PYTHIA 6.1 should now be considered obsolete, and are no longer maintained. For stable applications, the earlier combination JETSET 7.4 and PYTHIA 5.7 could still be used, however. (A note on backwards compatibility: persons who have code that relies on the old `/LUJETS/` single precision commonblock could easily write a translation routine to copy the `/PYJETS/` double precision information to `/LUJETS/`. In fact, among the old JETSET 7 routines, only `LUGIVE` and `LULOGO` routines have access to some PYTHIA commonblocks, and therefore these are the only ones that need to be modified if one, for some reason, would like to combine PYTHIA 6 with the old JETSET 7 routines.)

The move from JETSET 7.4 and PYTHIA 5.7 to PYTHIA 6.1 was a major one. For reasons of space, individual points are therefore not listed separately below, but only the main ones. The PYTHIA web page contains complete update notes, where all changes are documented by topic and subversion.

The main new features of PYTHIA 6.1, either present from the beginning or added later on, include:

- PYTHIA and JETSET have been merged.
- All real variables are declared in double precision.
- The internal mapping of particle codes has changed.

Table 1: The main versions of JETSET, with their date of appearance, published manuals, and main changes from previous versions.

No.	Date	Publ.	Main new or improved features
1	Nov 78	[Sjö78]	single-quark jets
2	May 79	[Sjö79]	heavy-flavour jets
3.1	Aug 79	—	2-jets in e^+e^-, preliminary 3-jets
3.2	Apr 80	[Sjö80]	3-jets in e^+e^- with full matrix elements, toponium $\to$ ggg decays
3.3	Aug 80	—	softer fragmentation spectrum
4.1	Apr 81	—	baryon production and diquark fragmentation, fourth-generation quarks, larger jet systems
4.2	Nov 81	—	low-$p_\perp$ physics
4.3	Mar 82	[Sjö82]	4-jets and QFD structure in e^+e^-,
	Jul 82	[Sjö83]	event-analysis routines
5.1	Apr 83	—	improved string fragmentation scheme, symmetric fragmentation, full 2nd order QCD for e^+e^-
5.2	Nov 83	—	momentum-conservation schemes for IF, initial-state photon radiation in e^+e^-
5.3	May 84	—	'popcorn' model for baryon production
6.1	Jan 85	—	common blocks restructured, parton showers
6.2	Oct 85	[Sjö86]	error detection
6.3	Oct 86	[Sjö87]	new parton-shower scheme
7.1	Feb 89	—	new particle codes and common block structure, more mesons, improved decays, vertex information, Abelian gluon model, Bose–Einstein effects
7.2	Nov 89	—	interface to new standard common block, photon emission in showers
7.3	May 90	[Sjö92d]	expanded support for non-standard particles
7.4	Dec 93	[Sjö94]	updated particle data and defaults

- The supersymmetric process machinery of SPYTHIA has been included and further improved, with several new processes.
- Many new processes of beyond-the-standard-model physics, in areas such as technicolor and doubly-charged Higgses.
- An expanded description of QCD processes in virtual-photon interactions, combined with a new machinery for the flux of virtual photons from leptons.
- Initial-state parton showers are matched to the next-to-leading order matrix elements for gauge boson production.
- Final-state parton showers are matched to a number of different first-order matrix elements for gluon emission, including full mass dependence.
- The hadronization description of low-mass strings has been improved, with consequences especially for heavy-flavour production.
- An alternative baryon production model has been introduced.
- Colour rearrangement is included as a new option, and several alternative Bose-Einstein

Table 2: The main versions of PYTHIA, with their date of appearance, published manuals, and main changes from previous versions.

No.	Date	Publ.	Main new or improved features
1	Dec 82	[Ben84]	synthesis of predecessors COMPTON, HIGHPT and KASSANDRA
2	—		
3.1	—		
3.2	—		
3.3	Feb 84	[Ben84a]	scale-breaking parton distributions
3.4	Sep 84	[Ben85]	more efficient kinematics selection
4.1	Dec 84		initial- and final-state parton showers, W and Z
4.2	Jun 85		multiple interactions
4.3	Aug 85		WW, WZ, ZZ and R processes
4.4	Nov 85		γW, γZ, $\gamma\gamma$ processes
4.5	Jan 86		H^0 production, diffractive and elastic events
4.6	May 86		angular correlation in resonance pair decays
4.7	May 86		Z'^0 and H^+ processes
4.8	Jan 87	[Ben87]	variable impact parameter in multiple interactions
4.9	May 87		gH^+ process
5.1	May 87		massive matrix elements for heavy quarks
5.2	Jun 87		intermediate boson scattering
5.3	Oct 89		new particle and subprocess codes, new common block structure, new kinematics selection, some lepton–lepton and lepton–hadron interactions, new subprocesses
5.4	Jun 90		*s*-dependent widths, resonances not on the mass shell, new processes, new parton distributions
5.5	Jan 91		improved e^+e^- and ep, several new processes
5.6	Sep 91	[Sjö92d]	reorganized parton distributions, new processes, user-defined external processes
5.7	Dec 93	[Sjö94]	new total cross sections, photoproduction, top decay
6.1	Mar 97	[Sjö01]	merger with JETSET, double precision, supersymmetry, technicolor, extra dimensions, etc. new processes, improved initial- and final-state showers, baryon production, virtual photon processes
6.1	Aug 01	this	user processes, lepton number violation

descriptions are added.

By comparison, the move from PYTHIA 6.1 to PYTHIA 6.2 was rather less dramatic. Again update notes tell the full story. Some of the main new features, which partly break backwards compatibility, are:

- A new machinery to handle user-defined external processes, according to the standard in [Boo01]. The old machinery is no longer available. Some of the alternatives for the `FRAME` argument in the `PYINIT` call have also been renamed to make way for a new `'USER'` option.
- The maximum size of the decay channel table has been increased from 4000 to 8000, affecting the `MDME`, `BRAT` and `KFDP` arrays in the `PYDAT3` common block.
- Lepton-number-violating decay channels have been included for supersymmetric particles [Ska01]. Thus the decay tables have grown considerably longer.
- The `PYSHOW` timelike showering routine has been expanded to allow showering inside systems consisting of up to seven particles, which can be made use of in some resonance decays and in user-defined processes.
- Some exotic particles and QCD effective states have been moved from temporary flavour codes to a PDG-consistent naming, and a few new codes have been introduced.
- The maximum number of documentation lines in the beginning of the event record has been expanded from 50 to 100.
- The default parton distribution set for the proton is now CTEQ 5L.
- The default Standard Model Higgs mass has been changed to 115 GeV.

3.2 Program Installation

The PYTHIA 'master copy' is the one found on the web page

```
http://www.thep.lu.se/~torbjorn/Pythia.html
```

There you have, for several subversions `xx`:

`pythia62xx.f`	the PYTHIA 6.2xx code,
`pythia62xx.tex`	this PYTHIA manual, and
`pythia62xx.update`	plain text update notes to the manual.

In addition to these, one may also find older versions of the program and manuals, sample main programs and other pieces of related software, and other physics papers.

The program is written essentially entirely in standard Fortran 77, and should run on any platform with such a compiler. To a first approximation, program compilation should therefore be straightforward.

Unfortunately, experience with many different compilers has been uniform: the options available for obtaining optimized code may actually produce erroneous code (e.g. operations inside `DO` loops are moved out before them, where some of the variables have not yet been properly set). Therefore the general advice is to use a low optimization level. Note that this is often not the default setting.

`SAVE` statements have been included in accordance with the Fortran standard.

All default settings and particle and process data are stored in `BLOCK DATA PYDATA`. This subprogram must be linked for a proper functioning of the other routines. On some platforms this is not done automatically but must be forced by you, e.g. by having a line

```
EXTERNAL PYDATA
```

at the beginning of your main program. This applies in particular if PYTHIA is maintained as a library from which routines are to be loaded only when they are needed. In this connection we

note that the library approach does not give any significant space advantages over a loading of the packages as a whole, since a normal run will call on most of the routines anyway, directly or indirectly.

With the move towards higher energies, e.g. for LHC applications, single-precision (32 bit) real arithmetic has become inappropriate. Therefore a declaration `IMPLICIT DOUBLE PRECISION(A-H,O-Z)` at the beginning of each subprogram is inserted to ensure double-precision (64 bit) real arithmetic. Remember that this also means that all calls to PYTHIA routines have to be done with real variables declared correspondingly in the user-written calling program. An `IMPLICIT INTEGER(I-N)` is also included to avoid problems on some compilers. Integer functions beginning with `PY` have to be declared explicitly. In total, therefore all routines begin with

```
C...Double precision and integer declarations.
      IMPLICIT DOUBLE PRECISION(A-H, O-Z)
      IMPLICIT INTEGER(I-N)
      INTEGER PYK,PYCHGE,PYCOMP
```

and you are recommended to do the same in your main programs. Note that, in running text and in description of commonblock default valuess, the more cumbersome double-precision notation is not always made explicit, but code examples should be correct.

On a machine where `DOUBLE PRECISION` would give 128 bits, it may make sense to use compiler options to revert to 64 bits, since the program is anyway not constructed to make use of 128 bit precision.

Fortran 77 makes no provision for double-precision complex numbers. Therefore complex numbers have been used only sparingly. However, some matrix element expressions, mainly for supersymmetric and technicolor processes, simplify considerably when written in terms of complex variables. In order to achieve a uniform precision, such variables have been declared `COMPLEX*16`, and are manipulated with functions such as `DCMPLX` and `DCONJG`. Affected are `PYSIGH`, `PYWIDT` and several of the supersymmetry routines. Should the compiler not accept this deviation from the standard, or some simple equivalent thereof (like `DOUBLE COMPLEX` instead of `COMPLEX*16`) these code pieces could be rewritten to ordinary `COMPLEX`, also converting the real numbers involved to and from single precision, with some drop in accuracy for the affected processes. `PYRESD` already contains some ordinary `COMPLEX` variables, and should not cause any problems.

Several compilers report problems when an odd number of integers precede a double-precision variable in a commonblock. Therefore an extra integer has been introduced as padding in a few instances, e.g. `NPAD`, `MSELPD` and `NGENPD`.

Since Fortran 77 provides no date-and-time routine, `PYTIME` allows a system-specific routine to be interfaced, with some commented-out examples given in the code. This routine is only used for cosmetic improvements of the output, however, so can be left at the default with time 0 given.

A test program, `PYTEST`, is included in the PYTHIA package. It is disguised as a subroutine, so you have to run a main program

```
      CALL PYTEST(1)
      END
```

This program will generate over a thousand events of different types, under a variety of conditions. If PYTHIA has not been properly installed, this program is likely to crash, or at least generate a number of erroneous events. This will then clearly be marked in the output, which otherwise will just contain a few sample event listings and a table of the number of different particles produced. To switch off the output of normal events and final table, use `PYTEST(0)` instead of `PYTEST(1)`. The final tally of errors detected should read 0.

For a program written to run PYTHIA 5 and JETSET 7, most of the conversion required for PYTHIA 6 is fairly straightforward, and can be automatized. Both a simple Fortran routine and

a more sophisticated Perl [Gar98] script exist to this end, see the PYTHIA web page. Some manual checks and interventions may still be required.

3.3 Program Philosophy

The Monte Carlo program is built as a slave system, i.e. you, the user, have to supply the main program. From this the various subroutines are called on to execute specific tasks, after which control is returned to the main program. Some of these tasks may be very trivial, whereas the 'high-level' routines by themselves may make a large number of subroutine calls. Many routines are not intended to be called directly by you, but only from higher-level routines such as PYEXEC, PYEEVT, PYINIT or PYEVNT.

Basically, this means that there are three ways by which you communicate with the programs. First, by setting common block variables, you specify the details of how the programs should perform specific tasks, e.g. which subprocesses should be generated, which particle masses should be assumed, which coupling constants used, which fragmentation scenarios, and so on with hundreds of options and parameters. Second, by calling subroutines you tell the programs to generate events according to the rules established above. Normally there are few subroutine arguments, and those are usually related to details of the physical situation, such as what c.m. energy to assume for events. Third, you can either look at the common block PYJETS to extract information on the generated event, or you can call on various functions and subroutines to analyse the event further for you.

It should be noted that, while the physics content is obviously at the centre of attention, the PYTHIA package also contains a very extensive setup of auxiliary service routines. The hope is that this will provide a comfortable working environment, where not only events are generated, but where you also linger on to perform a lot of the subsequent studies. Of course, for detailed studies, it may be necessary to interface the output directly to a detector simulation program.

The general rule is that all routines have names that are six characters long, beginning with PY. There are three exceptions the length rules: PYK, PYP and PYR. The former two functions are strongly coupled to the K and P matrices in the PYJETS common block, the latter is very frequently used. Also common block names are six characters long and start with PY. There are three integer functions, PYK,PYCHGE and PYCOMP. In all routines where they are to be used, they have to be declared INTEGER.

On the issue of initialization, the routines of different origin and functionality behave quite differently. Routines that are intended to be called from many different places, such as showers, fragmentation and decays, require no specific initialization (except for the one implied by the presence of BLOCK DATA PYDATA, see above), i.e. each event and each task stands on its own. Current common block values are used to perform the tasks in specific ways, and those rules can be changed from one event to the next (or even within the generation of one and the same event) without any penalty. The random-number generator is initialized at the first call, but usually this is transparent.

In the core process generation machinery (e.g. selection of the hard process kinematics), on the other hand, a sizeable amount of initialization is performed in the PYINIT call, and thereafter the events generated by PYEVNT all obey the rules established at that point. This improves the efficiency of the generation process, and also ties in with the Monte Carlo integration of the process cross section over many events. Therefore common block variables that specify methods and constraints to be used have to be set before the PYINIT call and then not be changed afterwards, with few exceptions. Of course, it is possible to perform several PYINIT calls in the same run, but there is a significant time overhead involved, so this is not something one would do for each new event. The two older separate process generation routines PYEEVT (and some of the routines called by it) and PYONIA also contain some elements of initialization, where there are a few advantages if events are generated in a coherent fashion. The cross section is not as complicated here, however, so the penalty for reinitialization is small, and also does not require any special user calls.

Apart from writing a title page, giving a brief initialization information, printing error messages if need be, and responding to explicit requests for listings, all tasks of the program are performed 'silently'. All output is directed to unit `MSTU(11)`, by default 6, and it is up to you to set this unit open for write. The only exceptions are `PYRGET`, `PYRSET` and `PYUPDA` where, for obvious reasons, the input/output file number is specified at each call. Here you again have to see to it that proper read/write access is set.

The programs are extremely versatile, but the price to be paid for this is having a large number of adjustable parameters and switches for alternative modes of operation. No single user is ever likely to need more than a fraction of the available options. Since all these parameters and switches are assigned sensible default values, there is no reason to worry about them until the need arises.

Unless explicitly stated (or obvious from the context) all switches and parameters can be changed independently of each other. One should note, however, that if only a few switches/parameters are changed, this may result in an artificially bad agreement with data. Many disagreements can often be cured by a subsequent retuning of some other parameters of the model, in particular those that were once determined by a comparison with data in the context of the default scenario. For example, for e^+e^- annihilation, such a retuning could involve one QCD parameter (α_s or Λ), the longitudinal fragmentation function, and the average transverse momentum in fragmentation.

The program contains a number of checks that requested processes have been implemented, that flavours specified for jet systems make sense, that the energy is sufficient to allow hadronization, that the memory space in `PYJETS` is large enough, etc. If anything goes wrong that the program can catch (obviously this may not always be possible), an error message will be printed and the treatment of the corresponding event will be cut short. In serious cases, the program will abort. As long as no error messages appear on the output, it may not be worthwhile to look into the rules for error checking, but if but one message appears, it should be enough cause for alarm to receive prompt attention. Also warnings are sometimes printed. These are less serious, and the experienced user might deliberately do operations which go against the rules, but still can be made to make sense in their context. Only the first few warnings will be printed, thereafter the program will be quiet. By default, the program is set to stop execution after ten errors, after printing the last erroneous event.

It must be emphasized that not all errors will be caught. In particular, one tricky question is what happens if an integer-valued common block switch or subroutine/function argument is used with a value that is not defined. In some subroutine calls, a prompt return will be expedited, but in most instances the subsequent action is entirely unpredictable, and often completely haywire. The same goes for real-valued variables that are assigned values outside the physically sensible range. One example will suffice here: if `PARJ(2)` is defined as the s/u suppression factor, a value > 1 will not give more profuse production of s than of u, but actually a spillover into c production. Users, beware!

3.4 Manual Conventions

In the manual parts of this report, some conventions are used. All names of subprograms, common blocks and variables are given in upper-case 'typewriter' style, e.g. `MSTP(111)=0`. Also program examples are given in this style.

If a common block variable must have a value set at the beginning of execution, then a default value is stored in the block data subprogram `PYDATA`. Such a default value is usually indicated by a '(D=...)' immediately after the variable name, e.g.
`MSTJ(1)` : (D=1) choice of fragmentation scheme.

All variables in the fragmentation-related common blocks (with very few exceptions, clearly marked) can be freely changed from one event to the next, or even within the treatment of one single event; see discussion on initialization in the previous section. In the process generation machinery common blocks the situation is more complicated. The values of many switches and

parameters are used already in the PYINIT call, and cannot be changed after that. The problem is mentioned in the preamble to the afflicted common blocks, which in particular means /PYPARS/ and /PYSUBS/. For the variables which may still be changed from one event to the next, a '(C)' is added after the '(D=...)' statement.

Normally, variables internal to the program are kept in separate common blocks and arrays, but in a few cases such internal variables appear among arrays of switches and parameters, mainly for historical reasons. These are denoted by '(R)' for variables you may want to read, because they contain potentially interesting information, and by '(I)' for purely internal variables. In neither case may the variables be changed by you.

In the description of a switch, the alternatives that this switch may take are often enumerated, e.g.

MSTJ(1) : (D=1) choice of fragmentation scheme.
- = 0 : no jet fragmentation at all.
- = 1 : string fragmentation according to the Lund model.
- = 2 : independent fragmentation, according to specification in MSTJ(2) and MSTJ(3).

If you then use any value other than 0, 1 or 2, results are undefined. The action could even be different in different parts of the program, depending on the order in which the alternatives are identified.

It is also up to you to choose physically sensible values for parameters: there is no check on the allowed ranges of variables. We gave an example of this at the end of the preceding section.

Subroutines you are expected to use are enclosed in a box at the point where they are defined:

```
CALL PYLIST(MLIST)
```

This is followed by a description of input or output parameters. The difference between input and output is not explicitly marked, but should be obvious from the context. In fact, the event-analysis routines of section 15.5 return values, while all the rest only have input variables.

Routines that are only used internally are not boxed in. However, we use boxes for all common blocks, so as to enhance the readability.

In running text, often specific switches and parameters will be mentioned, without a reference to the place where they are described further. The Index at the very end of the document allows you to find this place. Tables 3 and 4 gives a brief summary of almost all common blocks and the variables stored there. Often names for switches begin with MST and parameters with PAR. No common block variables begin with PY. There is thus no possibility to confuse an array element with a function or subroutine call.

3.5 Getting Started with the Simple Routines

Normally PYTHIA is expected to take care of the full event generation process. At times, however, one may want to access the more simple underlying routines, which allow a large flexibility to 'do it yourself'. We therefore start with a few cases of this kind, at the same time introducing some of the more frequently used utility routines.

As a first example, assume that you want to study the production of $u\bar{u}$ 2-jet systems at 20 GeV energy. To do this, write a main program

```
IMPLICIT DOUBLE PRECISION(A-H, O-Z)
CALL PY2ENT(0,2,-2,20D0)
CALL PYLIST(1)
END
```

and run this program, linked together with PYTHIA. The routine PY2ENT is specifically intended for storing two entries (partons or particles). The first argument (0) is a command to perform

Table 3: An almost complete list of common blocks, with brief comments on their main functions. The listing continues in Table 4.

```
C...The event record.
      COMMON/PYJETS/N,NPAD,K(4000,5),P(4000,5),V(4000,5)
C...Parameters.
      COMMON/PYDAT1/MSTU(200),PARU(200),MSTJ(200),PARJ(200)
C...Particle properties + some flavour parameters.
      COMMON/PYDAT2/KCHG(500,4),PMAS(500,4),PARF(2000),VCKM(4,4)
C...Decay information.
      COMMON/PYDAT3/MDCY(500,3),MDME(8000,2),BRAT(8000),KFDP(8000,5)
C...Particle names
      COMMON/PYDAT4/CHAF(500,2)
      CHARACTER CHAF*16
C...Random number generator information.
      COMMON/PYDATR/MRPY(6),RRPY(100)
C...Selection of hard scattering subprocesses.
      COMMON/PYSUBS/MSEL,MSELPD,MSUB(500),KFIN(2,-40:40),CKIN(200)
C...Parameters.
      COMMON/PYPARS/MSTP(200),PARP(200),MSTI(200),PARI(200)
C...Internal variables.
      COMMON/PYINT1/MINT(400),VINT(400)
C...Process information.
      COMMON/PYINT2/ISET(500),KFPR(500,2),COEF(500,20),ICOL(40,4,2)
C...Parton distributions and cross sections.
      COMMON/PYINT3/XSFX(2,-40:40),ISIG(1000,3),SIGH(1000)
C...Resonance width and secondary decay treatment.
      COMMON/PYINT4/MWID(500),WIDS(500,5)
C...Generation and cross section statistics.
      COMMON/PYINT5/NGENPD,NGEN(0:500,3),XSEC(0:500,3)
C...Process names.
      COMMON/PYINT6/PROC(0:500)
      CHARACTER PROC*28
C...Total cross sections.
      COMMON/PYINT7/SIGT(0:6,0:6,0:5)
C...Photon parton distributions: total and valence only.
      COMMON/PYINT8/XPVMD(-6:6),XPANL(-6:6),XPANH(-6:6),XPBEH(-6:6),
     &XPDIR(-6:6)
      COMMON/PYINT9/VXPVMD(-6:6),VXPANL(-6:6),VXPANH(-6:6),VXPDGM(-6:6)
C...Supersymmetry parameters.
      COMMON/PYMSSM/IMSS(0:99),RMSS(0:99)
C...Supersymmetry mixing matrices.
      COMMON/PYSSMT/ZMIX(4,4),UMIX(2,2),VMIX(2,2),SMZ(4),SMW(2),
     &SFMIX(16,4),ZMIXI(4,4),UMIXI(2,2),VMIXI(2,2)
C...R-parity-violating couplings in supersymmetry.
      COMMON/PYMSRV/RVLAM(3,3,3), RVLAMP(3,3,3), RVLAMB(3,3,3)
C...Internal parameters for R-parity-violating processes.
      COMMON/PYRVNV/AB(2,16,2),RMS(0:3),RES(6,5),IDR,IDR2,DCMASS,KFR(3)
      COMMON/PYRVPM/RM(0:3),A(2),B(2),RESM(2),RESW(2),MFLAG
      LOGICAL MFLAG
```

Table 4: Continuation of Table 3.

```
C...Parameters for Gauss integration of supersymmetric widths.
      COMMON/PYINTS/XXM(20)
      COMMON/PYG2DX/X1
C...Histogram information.
      COMMON/PYBINS/IHIST(4),INDX(1000),BIN(20000)
C...HEPEVT commonblock.
      PARAMETER (NMXHEP=4000)
      COMMON/HEPEVT/NEVHEP,NHEP,ISTHEP(NMXHEP),IDHEP(NMXHEP),
     &JMOHEP(2,NMXHEP),JDAHEP(2,NMXHEP),PHEP(5,NMXHEP),VHEP(4,NMXHEP)
      DOUBLE PRECISION PHEP,VHEP
C...User process initialization commonblock.
      INTEGER MAXPUP
      PARAMETER (MAXPUP=100)
      INTEGER IDBMUP,PDFGUP,PDFSUP,IDWTUP,NPRUP,LPRUP
      DOUBLE PRECISION EBMUP,XSECUP,XERRUP,XMAXUP
      COMMON/HEPRUP/IDBMUP(2),EBMUP(2),PDFGUP(2),PDFSUP(2),
     &IDWTUP,NPRUP,XSECUP(MAXPUP),XERRUP(MAXPUP),XMAXUP(MAXPUP),
     &LPRUP(MAXPUP)
C...User process event common block.
      INTEGER MAXNUP
      PARAMETER (MAXNUP=500)
      INTEGER NUP,IDPRUP,IDUP,ISTUP,MOTHUP,ICOLUP
      DOUBLE PRECISION XWGTUP,SCALUP,AQEDUP,AQCDUP,PUP,VTIMUP,SPINUP
      COMMON/HEPEUP/NUP,IDPRUP,XWGTUP,SCALUP,AQEDUP,AQCDUP,IDUP(MAXNUP),
     &ISTUP(MAXNUP),MOTHUP(2,MAXNUP),ICOLUP(2,MAXNUP),PUP(5,MAXNUP),
     &VTIMUP(MAXNUP),SPINUP(MAXNUP)
```

fragmentation and decay directly after the entries have been stored, the second and third that the two entries are u (2) and $\bar{\mathrm{u}}$ (-2), and the last that the c.m. energy of the pair is 20 GeV, in double precision. When this is run, the resulting event is stored in the PYJETS common block. This information can then be read out by you. No output is produced by PY2ENT itself, except for a title page which appears once for every PYTHIA run.

Instead the second command, to PYLIST, provides a simple visible summary of the information stored in PYJETS. The argument (1) indicates that the short version should be used, which is suitable for viewing the listing directly on an 80-column terminal screen. It might look as shown here.

```
                    Event listing (summary)

  I  particle/jet KS     KF orig   p_x     p_y     p_z      E       m

  1  (u)       A  12      2    0   0.000   0.000  10.000  10.000   0.006
  2  (ubar)    V  11     -2    0   0.000   0.000 -10.000  10.000   0.006
  3  (string)     11     92    1   0.000   0.000   0.000  20.000  20.000
  4  (rho+)       11    213    3   0.098  -0.154   2.710   2.856   0.885
  5  (rho-)       11   -213    3  -0.227   0.145   6.538   6.590   0.781
  6  pi+           1    211    3   0.125  -0.266   0.097   0.339   0.140
  7  (Sigma0)     11   3212    3  -0.254   0.034  -1.397   1.855   1.193
  8  (K*+)        11    323    3  -0.124   0.709  -2.753   2.968   0.846
  9  p~-           1  -2212    3   0.395  -0.614  -3.806   3.988   0.938
```

10	pi-	1	-211	3	-0.013	0.146	-1.389	1.403	0.140
11	pi+	1	211	4	0.109	-0.456	2.164	2.218	0.140
12	(pi0)	11	111	4	-0.011	0.301	0.546	0.638	0.135
13	pi-	1	-211	5	0.089	0.343	2.089	2.124	0.140
14	(pi0)	11	111	5	-0.316	-0.197	4.449	4.467	0.135
15	(Lambda0)	11	3122	7	-0.208	0.014	-1.403	1.804	1.116
16	gamma	1	22	7	-0.046	0.020	0.006	0.050	0.000
17	K+	1	321	8	-0.084	0.299	-2.139	2.217	0.494
18	(pi0)	11	111	8	-0.040	0.410	-0.614	0.751	0.135
19	gamma	1	22	12	0.059	0.146	0.224	0.274	0.000
20	gamma	1	22	12	-0.070	0.155	0.322	0.364	0.000
21	gamma	1	22	14	-0.322	-0.162	4.027	4.043	0.000
22	gamma	1	22	14	0.006	-0.035	0.422	0.423	0.000
23	p+	1	2212	15	-0.178	0.033	-1.343	1.649	0.938
24	pi-	1	-211	15	-0.030	-0.018	-0.059	0.156	0.140
25	gamma	1	22	18	-0.006	0.384	-0.585	0.699	0.000
26	gamma	1	22	18	-0.034	0.026	-0.029	0.052	0.000
		sum:	0.00		0.000	0.000	0.000	20.000	20.000

(A few blanks have been removed between the columns to make it fit into the format of this text.) Look in the particle/jet column and note that the first two lines are the original u and $\bar{u}$. The parentheses enclosing the names, '(u)' and '(ubar)', are there as a reminder that these partons actually have been allowed to fragment. The partons are still retained so that event histories can be studied. Also note that the KF (flavour code) column contains 2 in the first line and -2 in the second. These are the codes actually stored to denote the presence of a u and a $\bar{u}$, cf. the PY2ENT call, while the names written are just conveniences used when producing visible output. The A and V near the end of the particle/jet column indicate the beginning and end of a string (or cluster, or independent fragmentation) parton system; any intermediate entries belonging to the same system would have had an I in that column. (This gives a poor man's representation of an up-down arrow, $\updownarrow$.)

In the orig (origin) column, the zeros indicate that u and $\bar{u}$ are two initial entries. The subsequent line, number 3, denotes the fragmenting $u\bar{u}$ string system as a whole, and has origin 1, since the first parton of this string system is entry number 1. The particles in lines 4–10 have origin 3 to denote that they come directly from the fragmentation of this string. In string fragmentation it is not meaningful to say that a particle comes from only the u quark or only the $\bar{u}$ one. It is the string system as a whole that gives a ρ^+, a ρ^-, a π^+, a Σ^0, a K^{*+}, a $\bar{p}^-$, and a π^-. Note that some of the particle names are again enclosed in parentheses, indicating that these particles are not present in the final state either, but have decayed further. Thus the π^- in line 13 and the π^0 in line 14 have origin 5, as an indication that they come from the decay of the ρ^- in line 5. Only the names not enclosed in parentheses remain at the end of the fragmentation/decay chain, and are thus experimentally observable. The actual status code used to distinguish between different classes of entries is given in the KS column; codes in the range 1–10 correspond to remaining entries, and those above 10 to those that have fragmented or decayed.

The columns with p_x, p_y, p_z, E and m are quite self-explanatory. All momenta, energies and masses are given in units of GeV, since the speed of light is taken to be $c = 1$. Note that energy and momentum are conserved at each step of the fragmentation/decay process (although there exist options where this is not true). Also note that the z axis plays the rôle of preferred direction, along which the original partons are placed. The final line is intended as a quick check that nothing funny happened. It contains the summed charge, summed momentum, summed energy and invariant mass of the final entries at the end of the fragmentation/decay chain, and the values should agree with the input implied by the PY2ENT arguments. (In fact, warnings would normally appear on the output if anything untoward happened, but that is another story.)

The above example has illustrated roughly what information is to be had in the event record,

but not so much about how it is stored. This is better seen by using a 132-column format for listing events. Try e.g. the following program

```
      IMPLICIT DOUBLE PRECISION(A-H, O-Z)
      CALL PY3ENT(0,1,21,-1,30D0,0.9D0,0.7D0)
      CALL PYLIST(2)
      CALL PYEDIT(3)
      CALL PYLIST(2)
      END
```

where a 3-jet $dg\bar{d}$ event is generated in the first executable line and listed in the second. This listing will contain the numbers as directly stored in the common block PYJETS

```
      COMMON/PYJETS/N,NPAD,K(4000,5),P(4000,5),V(4000,5)
```

For particle I, K(I,1) thus gives information on whether or not a parton or particle has fragmented or decayed, K(I,2) gives the particle code, K(I,3) its origin, K(I,4) and K(I,5) the position of fragmentation/decay products, and P(I,1)–P(I,5) momentum, energy and mass. The number of lines in current use is given by N, i.e. $1 \leq$ I $\leq$ N. The V matrix contains decay vertices; to view those PYLIST(3) has to be used. NPAD is a dummy, needed to avoid some compiler troubles. It is important to learn the rules for how information is stored in PYJETS.

The third executable line in the program illustrates another important point about PYTHIA: a number of routines are available for manipulating and analyzing the event record after the event has been generated. Thus PYEDIT(3) will remove everything except stable charged particles, as shown by the result of the second PYLIST call. More advanced possibilities include things like sphericity or clustering routines. PYTHIA also contains some simple routines for histogramming, used to give self-contained examples of analysis procedures.

Apart from the input arguments of subroutine calls, control on the doings of PYTHIA may be imposed via many common blocks. Here sensible default values are always provided. A user might want to switch off all particle decays by putting MSTJ(21)=0 or increase the s/u ratio in fragmentation by putting PARJ(2)=0.40D0, to give but two examples. It is by exploring the possibilities offered here that PYTHIA can be turned into an extremely versatile tool, even if all the nice physics is already present in the default values.

As a final, semi-realistic example, assume that the $p_\perp$ spectrum of π^+ particles is to be studied in 91.2 GeV e^+e^- annihilation events, where $p_\perp$ is to be defined with respect to the sphericity axis. Using the internal routines for simple histogramming, a complete program might look like

```
C...Double precision and integer declarations.
      IMPLICIT DOUBLE PRECISION(A-H, O-Z)
      IMPLICIT INTEGER(I-N)
      INTEGER PYK,PYCHGE,PYCOMP

C...Common blocks.
      COMMON/PYJETS/N,NPAD,K(4000,5),P(4000,5),V(4000,5)

C...Book histograms.
      CALL PYBOOK(1,'pT spectrum of pi+',100,0D0,5D0)

C...Number of events to generate. Loop over events.
      NEVT=100
      DO 110 IEVT=1,NEVT

C...Generate event. List first one.
        CALL PYEEVT(0,91.2D0)
```

```
        IF(IEVT.EQ.1) CALL PYLIST(1)

C...Find sphericity axis.
        CALL PYSPHE(SPH,APL)

C...Rotate event so that sphericity axis is along z axis.
        CALL PYEDIT(31)

C...Loop over all particles, but skip if not pi+.
        DO 100 I=1,N
          IF(K(I,2).NE.211) GOTO 100

C...Calculate pT and fill in histogram.
          PT=SQRT(P(I,1)**2+P(I,2)**2)
          CALL PYFILL(1,PT,1D0)

C...End of particle and event loops.
  100   CONTINUE
  110 CONTINUE

C...Normalize histogram properly and list it.
      CALL PYFACT(1,20D0/NEVT)
      CALL PYHIST

      END
```

Study this program, try to understand what happens at each step, and run it to check that it works. You should then be ready to look at the relevant sections of this report and start writing your own programs.

3.6 Getting Started with the Event Generation Machinery

A run with the full PYTHIA event generation machinery has to be more strictly organized than the simple examples above, in that it is necessary to initialize the generation before events can be generated, and in that it is not possible to change switches and parameters freely during the course of the run. A fairly precise recipe for how a run should be structured can therefore be given.

Thus, the nowadays normal usage of PYTHIA can be subdivided into three steps.

1. The initialization step. It is here that all the basic characteristics of the coming generation are specified. The material in this section includes the following.
   - Declarations for double precision and integer variables and integer functions:
     ```
     IMPLICIT DOUBLE PRECISION(A-H, O-Z)
     IMPLICIT INTEGER(I-N)
     INTEGER PYK,PYCHGE,PYCOMP
     ```
   - Common blocks, at least the following, and maybe some more:
     ```
     COMMON/PYJETS/N,NPAD,K(4000,5),P(4000,5),V(4000,5)
     COMMON/PYDAT1/MSTU(200),PARU(200),MSTJ(200),PARJ(200)
     COMMON/PYSUBS/MSEL,MSELPD,MSUB(500),KFIN(2,-40:40),CKIN(200)
     COMMON/PYPARS/MSTP(200),PARP(200),MSTI(200),PARI(200)
     ```
   - Selection of required processes. Some fixed ‘menus’ of subprocesses can be selected with different `MSEL` values, but with `MSEL=0` it is possible to compose ‘à la carte’, using the subprocess numbers. To generate processes 14, 18 and 29, for instance, one needs

```
MSEL=0
MSUB(14)=1
MSUB(18)=1
MSUB(29)=1
```

- Selection of kinematics cuts in the `CKIN` array. To generate hard scatterings with 5 GeV $\leq p_\perp \leq$ 10 GeV, for instance, use

  ```
  CKIN(3)=5D0
  CKIN(4)=10D0
  ```

  Unfortunately, initial- and final-state radiation will shift around the kinematics of the hard scattering, making the effects of cuts less predictable. One therefore always has to be very careful that no desired event configurations are cut out.
- Definition of underlying physics scenario, e.g. Higgs mass.
- Selection of parton-distribution sets, Q^2 definitions, and all other details of the generation.
- Switching off of generator parts not needed for toy simulations, e.g. fragmentation for parton level studies.
- Initialization of the event generation procedure. Here kinematics is set up, maxima of differential cross sections are found for future Monte Carlo generation, and a number of other preparatory tasks carried out. Initialization is performed by `PYINIT`, which should be called only after the switches and parameters above have been set to their desired values. The frame, the beam particles and the energy have to be specified, e.g.

  ```
  CALL PYINIT('CMS','p','pbar',1800D0)
  ```
- Any other initial material required by you, e.g. histogram booking.

2. The generation loop. It is here that events are generated and studied. It includes the following tasks:
   - Generation of the next event, with

     ```
     CALL PYEVNT
     ```
   - Printing of a few events, to check that everything is working as planned, with

     ```
     CALL PYLIST(1)
     ```
   - An analysis of the event for properties of interest, either directly reading out information from the `PYJETS` common block or making use of the utility routines in PYTHIA.
   - Saving of events on disk or tape, or interfacing to detector simulation.
3. The finishing step. Here the tasks are:
   - Printing a table of deduced cross sections, obtained as a by-product of the Monte Carlo generation activity, with the command

     ```
     CALL PYSTAT(1)
     ```
   - Printing histograms and other user output.

To illustrate this structure, imagine a toy example, where one wants to simulate the production of a 300 GeV Higgs particle. In PYTHIA, a program for this might look something like the following.

```
C...Double precision and integer declarations.
      IMPLICIT DOUBLE PRECISION(A-H, O-Z)
      IMPLICIT INTEGER(I-N)
      INTEGER PYK,PYCHGE,PYCOMP

C...Common blocks.
      COMMON/PYJETS/N,NPAD,K(4000,5),P(4000,5),V(4000,5)
      COMMON/PYDAT1/MSTU(200),PARU(200),MSTJ(200),PARJ(200)
      COMMON/PYDAT2/KCHG(500,4),PMAS(500,4),PARF(2000),VCKM(4,4)
```

```
      COMMON/PYDAT3/MDCY(500,3),MDME(8000,2),BRAT(8000),KFDP(8000,5)
      COMMON/PYSUBS/MSEL,MSELPD,MSUB(500),KFIN(2,-40:40),CKIN(200)
      COMMON/PYPARS/MSTP(200),PARP(200),MSTI(200),PARI(200)

C...Number of events to generate. Switch on proper processes.
      NEV=1000
      MSEL=0
      MSUB(102)=1
      MSUB(123)=1
      MSUB(124)=1

C...Select Higgs mass and kinematics cuts in mass.
      PMAS(25,1)=300D0
      CKIN(1)=290D0
      CKIN(2)=310D0

C...For simulation of hard process only: cut out unnecessary tasks.
      MSTP(61)=0
      MSTP(71)=0
      MSTP(81)=0
      MSTP(111)=0

C...Initialize and list partial widths.
      CALL PYINIT('CMS','p','p',14000D0)
      CALL PYSTAT(2)

C...Book histogram.
      CALL PYBOOK(1,'Higgs mass',50,275D0,325D0)

C...Generate events. Look at first few.
      DO 200 IEV=1,NEV
        CALL PYEVNT
        IF(IEV.LE.3) CALL PYLIST(1)

C...Loop over particles to find Higgs and histogram its mass.
        DO 100 I=1,N
          IF(K(I,1).LT.20.AND.K(I,2).EQ.25) HMASS=P(I,5)
  100   CONTINUE
        CALL PYFILL(1,HMASS,1D0)
  200 CONTINUE

C...Print cross sections and histograms.
      CALL PYSTAT(1)
      CALL PYHIST

      END
```

Here 102, 123 and 124 are the three main Higgs production graphs gg $\to$ h, ZZ $\to$ h, and WW $\to$ h, and `MSUB(ISUB)=1` is the command to switch on process `ISUB`. Full freedom to combine subprocesses ‘à la carte’ is ensured by `MSEL=0`; ready-made ‘menus’ can be ordered with other `MSEL` numbers. The `PMAS` command sets the mass of the Higgs, and the `CKIN` variables the desired mass range of the Higgs — a Higgs with a 300 GeV nominal mass actually has a fairly broad Breit–Wigner type mass distribution. The `MSTP` switches that come next are there to modify the generation procedure, in this case to switch off initial- and final-state radiation,

multiple interactions among beam jets, and fragmentation, to give only the 'parton skeleton' of the hard process. The `PYINIT` call initializes PYTHIA, by finding maxima of cross sections, recalculating the Higgs decay properties (which depend on the Higgs mass), etc. The decay properties can be listed with `PYSTAT(2)`.

Inside the event loop, `PYEVNT` is called to generate an event, and `PYLIST(1)` to list the event. The information used by `PYLIST(1)` is the event record, stored in the common block `PYJETS`. Here one finds all produced particles, both final and intermediate ones, with information on particle species and event history (`K` array), particle momenta (`P` array) and production vertices (`V` array). In the loop over all particles produced, `1` through `N`, the Higgs particle is found by its code, `K(I,2)=25`, and its mass is stored in `P(I,5)`.

After all events have been generated, `PYSTAT(1)` gives a summary of the number of events generated in the various allowed channels, and the inferred cross sections.

In the run above, a typical event listing might look like the following.

```
                    Event listing (summary)

   I  particle/jet     KF    p_x      p_y      p_z       E        m

   1  !p+!           2212    0.000    0.000 8000.000 8000.000    0.938
   2  !p+!           2212    0.000    0.000-8000.000 8000.000    0.938
=====================================================================
   3  !g!              21   -0.505   -0.229   28.553   28.558    0.000
   4  !g!              21    0.224    0.041 -788.073  788.073    0.000
   5  !g!              21   -0.505   -0.229   28.553   28.558    0.000
   6  !g!              21    0.224    0.041 -788.073  788.073    0.000
   7  !H0!             25   -0.281   -0.188 -759.520  816.631  300.027
   8  !W+!             24  120.648   35.239 -397.843  424.829   80.023
   9  !W-!            -24 -120.929  -35.426 -361.677  391.801   82.579
  10  !e+!            -11   12.922   -4.760 -160.940  161.528    0.001
  11  !nu_e!           12  107.726   39.999 -236.903  263.302    0.000
  12  !s!               3  -62.423    7.195 -256.713  264.292    0.199
  13  !cbar!           -4  -58.506  -42.621 -104.963  127.509    1.350
=====================================================================
  14  (H0)             25   -0.281   -0.188 -759.520  816.631  300.027
  15  (W+)             24  120.648   35.239 -397.843  424.829   80.023
  16  (W-)            -24 -120.929  -35.426 -361.677  391.801   82.579
  17  e+              -11   12.922   -4.760 -160.940  161.528    0.001
  18  nu_e             12  107.726   39.999 -236.903  263.302    0.000
  19  s         A       3  -62.423    7.195 -256.713  264.292    0.199
  20  cbar      V      -4  -58.506  -42.621 -104.963  127.509    1.350
  21  ud_1      A    2103   -0.101    0.176 7971.328 7971.328    0.771
  22  d         V       1   -0.316    0.001  -87.390   87.390    0.010
  23  u         A       2    0.606    0.052   -0.751    0.967    0.006
  24  uu_1      V    2203    0.092   -0.042-7123.668 7123.668    0.771
=====================================================================
               sum:  2.00     0.00     0.00     0.00 15999.98 15999.98
```

The above event listing is abnormally short, in part because some columns of information were removed to make it fit into this text, in part because all initial- and final-state QCD radiation, all non-trivial beam jet structure, and all fragmentation was inhibited in the generation. Therefore only the skeleton of the process is visible. In lines 1 and 2 one recognizes the two incoming protons. In lines 3 and 4 are incoming partons before initial-state radiation and in 5 and 6 after — since there is no such radiation they coincide here. Line 7 shows the Higgs produced by gg fusion, 8 and 9 its decay products and 10–13 the second-step decay products. Up to this point

lines give a summary of the event history, indicated by the exclamation marks that surround particle names (and also reflected in the `K(I,1)` code, not shown). From line 14 onwards come the particles actually produced in the final states, first in lines 14–16 particles that subsequently decayed, which have their names surrounded by brackets, and finally the particles and partons left in the end, including beam remnants. Here this also includes a number of unfragmented partons, since fragmentation was inhibited. Ordinarily, the listing would have gone on for a few hundred more lines, with the particles produced in the fragmentation and their decay products. The final line gives total charge and momentum, as a convenient check that nothing unexpected happened. The first column of the listing is just a counter, the second gives the particle name and information on status and string drawing (the `A` and `V`), the third the particle-flavour code (which is used to give the name), and the subsequent columns give the momentum components.

One of the main problems is to select kinematics efficiently. Imagine for instance that one is interested in the production of a single Z with a transverse momentum in excess of 50 GeV. If one tries to generate the inclusive sample of Z events, by the basic production graphs $q\overline{q} \to Z$, then most events will have low transverse momenta and will have to be discarded. That any of the desired events are produced at all is due to the initial-state generation machinery, which can build up transverse momenta for the incoming q and $\overline{q}$. However, the amount of initial-state radiation cannot be constrained beforehand. To increase the efficiency, one may therefore turn to the higher-order processes $qg \to Zq$ and $q\overline{q} \to Zg$, where already the hard subprocess gives a transverse momentum to the Z. This transverse momentum can be constrained as one wishes, but again initial- and final-state radiation will smear the picture. If one were to set a $p_\perp$ cut at 50 GeV for the hard-process generation, those events where the Z was given only 40 GeV in the hard process but got the rest from initial-state radiation would be missed. Not only therefore would cross sections come out wrong, but so might the typical event shapes. In the end, it is therefore necessary to find some reasonable compromise, by starting the generation at 30 GeV, say, if one knows that only rarely do events below this value fluctuate up to 50 GeV. Of course, most events will therefore not contain a Z above 50 GeV, and one will have to live with some inefficiency. It is not uncommon that only one event out of ten can be used, and occasionally it can be even worse.

If it is difficult to set kinematics, it is often easier to set the flavour content of a process. In a Higgs study, one might wish, for example, to consider the decay $h^0 \to Z^0Z^0$, with each $Z^0 \to e^+e^-$ or $\mu^+\mu^-$. It is therefore necessary to inhibit all other h^0 and Z^0 decay channels, and also to adjust cross sections to take into account this change, all of which is fairly straightforward. The same cannot be said for decays of ordinary hadrons, where the number produced in a process is not known beforehand, and therefore inconsistencies easily can arise if one tries to force specific decay channels.

In the examples given above, all run-specific parameters are set in the code (in the main program; alternatively it could be in a subroutine called by the main program). This approach is allowing maximum flexibility to change parameters during the course of the run. However, in many experimental collaborations one does not want to allow this freedom, but only one set of parameters, to be read in from an external file at the beginning of a run and thereafter never changed. This in particular applies when PYTHIA is to be linked with other libraries, such as GEANT [Bru89] and detector-specific software. While a linking of a normal-sized main program with PYTHIA is essentially instantaneous on current platforms (typically less than a second), this may not hold for other libraries. For this purpose one then needs a parser of PYTHIA parameters, the core of which can be provided by the `PYGIVE` routine.

As an example, consider a main program of the form

```
C...Double precision and integer declarations.
      IMPLICIT DOUBLE PRECISION(A-H, O-Z)
      IMPLICIT INTEGER(I-N)
      INTEGER PYK,PYCHGE,PYCOMP
C...Input and output strings.
      CHARACTER FRAME*12,BEAM*12,TARGET*12,PARAM*100
```

```
C...Read parameters for PYINIT call.
      READ(*,*) FRAME,BEAM,TARGET,ENERGY

C...Read number of events to generate, and to print.
      READ(*,*) NEV,NPRT

C...Loop over reading and setting parameters/switches.
  100 READ(*,'(A)',END=200) PARAM
      CALL PYGIVE(PARAM)
      GOTO 100

C...Initialize PYTHIA.
  200 CALL PYINIT(FRAME,BEAM,TARGET,ENERGY)

C...Event generation loop
      DO 300 IEV=1,NEV
        CALL PYEVNT
        IF(IEV.LE.NPRT) CALL PYLIST(1)
  300 CONTINUE

C...Print cross sections.
      CALL PYSTAT(1)

      END
```

and a file `indata` with the contents

```
CMS,p,p,14000.
1000,3
! below follows commands sent to PYGIVE
MSEL=0           ! Mix processes freely
MSUB(102)=1      ! g + g -> h0
MSUB(123)=1      ! Z0 + Z0 -> h0
MSUB(124)=1      ! W+ + W- -> h0
PMAS(25,1)=300.  ! Higgs mass
CKIN(1)=290.     ! lower cutoff on mass
CKIN(2)=310.     ! upper cutoff on mass
MSTP(61)=0       ! no initial-state showers
MSTP(71)=0       ! no final-state showers
MSTP(81)=0       ! no multiple interactions
MSTP(111)=0      ! no hadronization
```

Here the text following the exclamation marks is interpreted as a comment by `PYGIVE`, and thus purely intended to allow better documentation of changes. The main program could then be linked to PYTHIA, to an executable `a.out`, and run e.g. with a Unix command line

```
a.out < indata > output
```

to produce results on the file `output`. Here the `indata` could be changed without requiring a recompilation. Of course, the main program would have to be more realistic, e.g. with events saved to disk or tape, but the principle should be clear.

4 Monte Carlo Techniques

Quantum mechanics introduces a concept of randomness in the behaviour of physical processes. The virtue of event generators is that this randomness can be simulated by the use of Monte Carlo techniques. In the process, the program authors have to use some ingenuity to find the most efficient way to simulate an assumed probability distribution. A detailed description of possible techniques would carry us too far, but in this section some of the most frequently used approaches are presented, since they will appear in discussions in subsequent sections. Further examples may be found e.g. in [Jam80].

First of all one assumes the existence of a random number generator. This is a (Fortran) function which, each time it is called, returns a number R in the range between 0 and 1, such that the inclusive distribution of numbers R is flat in the range, and such that different numbers R are uncorrelated. The random number generator that comes with PYTHIA is described at the end of this section, and we defer the discussion until then.

4.1 Selection From a Distribution

The situation that is probably most common is that we know a function $f(x)$ which is non-negative in the allowed x range $x_{\min} \leq x \leq x_{\max}$. We want to select an x 'at random' so that the probability in a small interval $\mathrm{d}x$ around a given x is proportional to $f(x)\,\mathrm{d}x$. Here $f(x)$ might be a fragmentation function, a differential cross section, or any of a number of distributions.

One does not have to assume that the integral of $f(x)$ is explicitly normalized to unity: by the Monte Carlo procedure of picking exactly one accepted x value, normalization is implicit in the final result. Sometimes the integral of $f(x)$ does carry a physics content of its own, as part of an overall weight factor we want to keep track of. Consider, for instance, the case when x represents one or several phase-space variables and $f(x)$ a differential cross section; here the integral has a meaning of total cross section for the process studied. The task of a Monte Carlo is then, on the one hand, to generate events one at a time, and, on the other hand, to estimate the total cross section. The discussion of this important example is deferred to section 7.4.

If it is possible to find a primitive function $F(x)$ which has a known inverse $F^{-1}(x)$, an x can be found as follows (method 1):

$$\int_{x_{\min}}^{x} f(x)\,\mathrm{d}x = R \int_{x_{\min}}^{x_{\max}} f(x)\,\mathrm{d}x$$
$$\Longrightarrow \quad x = F^{-1}(F(x_{\min}) + R(F(x_{\max}) - F(x_{\min}))) \; . \tag{2}$$

The statement of the first line is that a fraction R of the total area under $f(x)$ should be to the left of x. However, seldom are functions of interest so nice that the method above works. It is therefore necessary to use more complicated schemes.

Special tricks can sometimes be found. Consider e.g. the generation of a Gaussian $f(x) = \exp(-x^2)$. This function is not integrable, but if we combine it with the same Gaussian distribution of a second variable y, it is possible to transform to polar coordinates

$$f(x)\,\mathrm{d}x\, f(y)\,\mathrm{d}y = \exp(-x^2 - y^2)\,\mathrm{d}x\,\mathrm{d}y = r\exp(-r^2)\,\mathrm{d}r\,\mathrm{d}\varphi \; , \tag{3}$$

and now the r and φ distributions may be easily generated and recombined to yield x. At the same time we get a second number y, which can also be used. For the generation of transverse momenta in fragmentation, this is very convenient, since in fact we want to assign two transverse degrees of freedom.

If the maximum of $f(x)$ is known, $f(x) \leq f_{\max}$ in the x range considered, a hit-or-miss method will always yield the correct answer (method 2):

1. select an x with even probability in the allowed range, i.e. $x = x_{\min} + R(x_{\max} - x_{\min})$;

2. compare a (new) R with the ratio $f(x)/f_{\max}$; if $f(x)/f_{\max} \leq R$, then reject the x value and return to point 1 for a new try;
3. otherwise the most recent x value is retained as final answer.

The probability that $f(x)/f_{\max} > R$ is proportional to $f(x)$; hence the correct distribution of retained x values. The efficiency of this method, i.e. the average probability that an x will be retained, is $(\int f(x)\,\mathrm{d}x)/(f_{\max}(x_{\max} - x_{\min}))$. The method is acceptable if this number is not too low, i.e. if $f(x)$ does not fluctuate too wildly.

Very often $f(x)$ does have narrow spikes, and it may not even be possible to define an $f_{\max}$. An example of the former phenomenon is a function with a singularity just outside the allowed region, an example of the latter an integrable singularity just at the $x_{\min}$ and/or $x_{\max}$ borders. Variable transformations may then be used to make a function smoother. Thus a function $f(x)$ which blows up as $1/x$ for $x \to 0$, with an $x_{\min}$ close to 0, would instead be roughly constant if transformed to the variable $y = \ln x$.

The variable transformation strategy may be seen as a combination of methods 1 and 2, as follows. Assume the existence of a function $g(x)$, with $f(x) \leq g(x)$ over the x range of interest. Here $g(x)$ is picked to be a 'simple' function, such that the primitive function $G(x)$ and its inverse $G^{-1}(x)$ are known. Then (method 3):

1. select an x according to the distribution $g(x)$, using method 1;
2. compare a (new) R with the ratio $f(x)/g(x)$; if $f(x)/g(x) \leq R$, then reject the x value and return to point 1 for a new try;
3. otherwise the most recent x value is retained as final answer.

This works, since the first step will select x with a probability $g(x)\,\mathrm{d}x = \mathrm{d}G(x)$ and the second retain this choice with probability $f(x)/g(x)$. The total probability to pick a value x is then just the product of the two, i.e. $f(x)\,\mathrm{d}x$.

If $f(x)$ has several spikes, method 3 may work for each spike separately, but it may not be possible to find a $g(x)$ that covers all of them at the same time, and which still has an invertible primitive function. However, assume that we can find a function $g(x) = \sum_i g_i(x)$, such that $f(x) \leq g(x)$ over the x range considered, and such that the functions $g_i(x)$ each are non-negative and simple, in the sense that we can find primitive functions and their inverses. In that case (method 4):

1. select an i at random, with relative probability given by the integrals

$$\int_{x_{\min}}^{x_{\max}} g_i(x)\,\mathrm{d}x = G_i(x_{\max}) - G_i(x_{\min}) \;; \tag{4}$$

2. for the i selected, use method 1 to find an x, i.e.

$$x = G_i^{-1}(G_i(x_{\min}) + R(G_i(x_{\max}) - G_i(x_{\min}))) \;; \tag{5}$$

3. compare a (new) R with the ratio $f(x)/g(x)$; if $f(x)/g(x) \leq R$, then reject the x value and return to point 1 for a new try;
4. otherwise the most recent x value is retained as final answer.

This is just a trivial extension of method 3, where steps 1 and 2 ensure that, on the average, each x value picked there is distributed according to $g(x)$: the first step picks i with relative probability $\int g_i(x)\,\mathrm{d}x$, the second x with absolute probability $g_i(x)/\int g_i(x)\,\mathrm{d}x$ (this is one place where one must remember to do normalization correctly); the product of the two is therefore $g_i(x)$ and the sum over all i gives back $g(x)$.

We have now arrived at an approach that is sufficiently powerful for a large selection of problems. In general, for a function $f(x)$ which is known to have sharp peaks in a few different places, the generic behaviour at each peak separately may be covered by one or a few simple functions $g_i(x)$, to which one adds a few more $g_i(x)$ to cover the basic behaviour away from the peaks. By a suitable selection of the relative strengths of the different g_i's, it is possible to find a function $g(x)$ that matches well the general behaviour of $f(x)$, and thus achieve a reasonable Monte Carlo efficiency.

The major additional complication is when x is a multidimensional variable. Usually the problem is not so much $f(x)$ itself, but rather that the phase-space boundaries may be very complicated. If the boundaries factorize it is possible to pick phase-space points restricted to the desired region. Otherwise the region may have to be inscribed in a hyper-rectangle, with points picked within the whole hyper-rectangle but only retained if they are inside the allowed region. This may lead to a significant loss in efficiency. Variable transformations may often make the allowed region easier to handle.

There are two main methods to handle several dimensions, each with its set of variations. The first method is based on a factorized ansatz, i.e. one attempts to find a function $g(\mathbf{x})$ which is everywhere larger than $f(\mathbf{x})$, and which can be factorized into $g(\mathbf{x}) = g^{(1)}(x_1)\, g^{(2)}(x_2) \cdots g^{(n)}(x_n)$, where $\mathbf{x} = (x_1, x_2, \ldots, x_n)$. Here each $g^{(j)}(x_j)$ may in its turn be a sum of functions $g_i^{(j)}$, as in method 4 above. First, each x_j is selected independently, and afterwards the ratio $f(\mathbf{x})/g(\mathbf{x})$ is used to determine whether to retain the point.

The second method is useful if the boundaries of the allowed region can be written in a form where the maximum range of x_1 is known, the allowed range of x_2 only depends on x_1, that of x_3 only on x_1 and x_2, and so on until x_n, whose range may depend on all the preceding variables. In that case it may be possible to find a function $g(\mathbf{x})$ that can be integrated over x_2 through x_n to yield a simple function of x_1, according to which x_1 is selected. Having done that, x_2 is selected according to a distribution which now depends on x_1, but with x_3 through x_n integrated over. In particular, the allowed range for x_2 is known. The procedure is continued until x_n is reached, where now the function depends on all the preceding x_j values. In the end, the ratio $f(\mathbf{x})/g(\mathbf{x})$ is again used to determine whether to retain the point.

4.2 The Veto Algorithm

The 'radioactive decay' type of problems is very common, in particular in parton showers, but it is also used, e.g. in the multiple interactions description in PYTHIA. In this kind of problems there is one variable t, which may be thought of as giving a kind of time axis along which different events are ordered. The probability that 'something will happen' (a nucleus decay, a parton branch) at time t is described by a function $f(t)$, which is non-negative in the range of t values to be studied. However, this naïve probability is modified by the additional requirement that something can only happen at time t if it did not happen at earlier times $t' < t$. (The original nucleus cannot decay once again if it already did decay; possibly the decay products may decay in their turn, but that is another question.)

The probability that nothing has happened by time t is expressed by the function $\mathcal{N}(t)$ and the differential probability that something happens at time t by $\mathcal{P}(t)$. The basic equation then is

$$\mathcal{P}(t) = -\frac{\mathrm{d}\mathcal{N}}{\mathrm{d}t} = f(t)\, \mathcal{N}(t) \; . \tag{6}$$

For simplicity, we shall assume that the process starts at time $t = 0$, with $\mathcal{N}(0) = 1$.

The above equation can be solved easily if one notes that $\mathrm{d}\mathcal{N}/\mathcal{N} = \mathrm{d}\ln\mathcal{N}$:

$$\mathcal{N}(t) = \mathcal{N}(0) \exp\left\{ -\int_0^t f(t')\, \mathrm{d}t' \right\} = \exp\left\{ -\int_0^t f(t')\, \mathrm{d}t' \right\} \; , \tag{7}$$

and thus

$$\mathcal{P}(t) = f(t) \exp\left\{ -\int_0^t f(t')\, \mathrm{d}t' \right\} \; . \tag{8}$$

With $f(t) = c$ this is nothing but the textbook formulae for radioactive decay. In particular, at small times the correct decay probability, $\mathcal{P}(t)$, agrees well with the input one, $f(t)$, since the exponential factor is close to unity there. At larger t, the exponential gives a dampening which

ensures that the integral of $\mathcal{P}(t)$ never can exceed unity, even if the integral of $f(t)$ does. The exponential can be seen as the probability that nothing happens between the original time 0 and the final time t. In the parton-shower language, this corresponds to the so-called Sudakov form factor.

If $f(t)$ has a primitive function with a known inverse, it is easy to select t values correctly:

$$\int_0^t \mathcal{P}(t')\,\mathrm{d}t' = \mathcal{N}(0) - \mathcal{N}(t) = 1 - \exp\left\{-\int_0^t f(t')\,\mathrm{d}t'\right\} = 1 - R\ , \tag{9}$$

which has the solution

$$F(0) - F(t) = \ln R \quad\Longrightarrow\quad t = F^{-1}(F(0) - \ln R)\ . \tag{10}$$

If $f(t)$ is not sufficiently nice, one may again try to find a better function $g(t)$, with $f(t) \leq g(t)$ for all $t \geq 0$. However to use method 3 with this $g(t)$ would not work, since the method would not correctly take into account the effects of the exponential term in $\mathcal{P}(t)$. Instead one may use the so-called veto algorithm:

1. start with $i = 0$ and $t_0 = 0$;
2. add 1 to i and select $t_i = G^{-1}(G(t_{i-1}) - \ln R)$, i.e. according to $g(t)$, but with the constraint that $t_i > t_{i-1}$,
3. compare a (new) R with the ratio $f(t_i)/g(t_i)$; if $f(t_i)/g(t_i) \leq R$, then return to point 2 for a new try;
4. otherwise t_i is retained as final answer.

It may not be apparent why this works. Consider, however, the various ways in which one can select a specific time t. The probability that the first try works, $t = t_1$, i.e. that no intermediate t values need be rejected, is given by

$$\mathcal{P}_0(t) = \exp\left\{-\int_0^t g(t')\,\mathrm{d}t'\right\} g(t)\,\frac{f(t)}{g(t)} = f(t)\exp\left\{-\int_0^t g(t')\,\mathrm{d}t'\right\}\ , \tag{11}$$

where the exponential times $g(t)$ comes from eq. (8) applied to g, and the ratio $f(t)/g(t)$ is the probability that t is accepted. Now consider the case where one intermediate time t_1 is rejected and $t = t_2$ is only accepted in the second step. This gives

$$\mathcal{P}_1(t) = \int_0^t \mathrm{d}t_1 \exp\left\{-\int_0^{t_1} g(t')\,\mathrm{d}t'\right\} g(t_1)\left[1 - \frac{f(t_1)}{g(t_1)}\right] \exp\left\{-\int_{t_1}^t g(t')\,\mathrm{d}t'\right\} g(t)\,\frac{f(t)}{g(t)}\ , \tag{12}$$

where the first exponential times $g(t_1)$ gives the probability that t_1 is first selected, the square brackets the probability that t_1 is subsequently rejected, the following piece the probability that $t = t_2$ is selected when starting from t_1, and the final factor that t is retained. The whole is to be integrated over all possible intermediate times t_1. The exponentials together give an integral over the range from 0 to t, just as in $\mathcal{P}_0$, and the factor for the final step being accepted is also the same, so therefore one finds that

$$\mathcal{P}_1(t) = \mathcal{P}_0(t)\int_0^t \mathrm{d}t_1\,[g(t_1) - f(t_1)]\ . \tag{13}$$

This generalizes. In $\mathcal{P}_2$ one has to consider two intermediate times, $0 \leq t_1 \leq t_2 \leq t_3 = t$, and so

$$\begin{aligned}\mathcal{P}_2(t) &= \mathcal{P}_0(t)\int_0^t \mathrm{d}t_1\,[g(t_1) - f(t_1)]\int_{t_1}^t \mathrm{d}t_2\,[g(t_2) - f(t_2)] \\ &= \mathcal{P}_0(t)\frac{1}{2}\left(\int_0^t \left[g(t') - f(t')\right]\mathrm{d}t'\right)^2\ .\end{aligned} \tag{14}$$

The last equality is most easily seen if one also considers the alternative region $0 \leq t_2 \leq t_1 \leq t$, where the rôles of t_1 and t_2 have just been interchanged, and the integral therefore has the same value as in the region considered. Adding the two regions, however, the integrals over t_1 and t_2 decouple, and become equal. In general, for $\mathcal{P}_i$, the i intermediate times can be ordered in $i!$ different ways. Therefore the total probability to accept t, in any step, is

$$\begin{aligned}
\mathcal{P}(t) &= \sum_{i=0}^{\infty} \mathcal{P}_i(t) = \mathcal{P}_0(t) \sum_{i=0}^{\infty} \frac{1}{i!} \left(\int_0^t \left[g(t') - f(t') \right] \mathrm{d}t' \right)^i \\
&= f(t) \exp\left\{ -\int_0^t g(t')\,\mathrm{d}t' \right\} \exp\left\{ \int_0^t \left[g(t') - f(t') \right] \mathrm{d}t' \right\} \\
&= f(t) \exp\left\{ -\int_0^t f(t')\,\mathrm{d}t' \right\} , \qquad (15)
\end{aligned}$$

which is the desired answer.

If the process is to be stopped at some scale $t_{\max}$, i.e. if one would like to remain with a fraction $\mathcal{N}(t_{\max})$ of events where nothing happens at all, this is easy to include in the veto algorithm: just iterate upwards in t at usual, but stop the process if no allowed branching is found before $t_{\max}$.

Usually $f(t)$ is a function also of additional variables x. The methods of the preceding section are easy to generalize if one can find a suitable function $g(t,x)$ with $f(t,x) \leq g(t,x)$. The $g(t)$ used in the veto algorithm is the integral of $g(t,x)$ over x. Each time a t_i has been selected also an x_i is picked, according to $g(t_i,x)\,dx$, and the (t,x) point is accepted with probability $f(t_i,x_i)/g(t_i,x_i)$.

4.3 The Random Number Generator

In recent years, progress has been made in constructing portable generators with large periods and other good properties; see the review [Jam90]. Therefore the current version contains a random number generator based on the algorithm proposed by Marsaglia, Zaman and Tsang [Mar90]. This routine should work on any machine with a mantissa of at least 48 digits, i.e. on computers with a 64-bit (or more) representation of double precision real numbers. Given the same initial state, the sequence will also be identical on different platforms. This need not mean that the same sequence of events will be generated, since the different treatments of roundoff errors in numerical operations will lead to slightly different real numbers being tested against these random numbers in IF statements. Also code optimization may lead to a divergence of the event sequence.

Apart from nomenclature issues, the coding of PYR as a function rather than a subroutine, and the extension to double precision, the only difference between our code and the code given in [Jam90] is that slightly different algorithms are used to ensure that the random number is not equal to 0 or 1 within the machine precision. Further developments of the algorithm has been proposed [Lüs94] to remove residual possibilities of small long-range correlations, at the price of a slower generation procedure. However, given that PYTHIA is using random numbers for so many different tasks, without any fixed cycle, this has been deemed unnecessary.

The generator has a period of over 10^{43}, and the possibility to obtain almost 10^9 different and disjoint subsequences, selected by giving an initial integer number. The price to be paid for the long period is that the state of the generator at a given moment cannot be described by a single integer, but requires about 100 words. Some of these are real numbers, and are thus not correctly represented in decimal form. The old-style procedure, which made it possible to restart the generation from a seed value written to the run output, is therefore not convenient. The CERN library implementation keeps track of the number of random numbers generated since the start. With this value saved, in a subsequent run the random generator can be asked to skip ahead the corresponding number of random numbers. PYTHIA is a heavy user of random

numbers, however: typically 30% of the full run time is spent on random number generation. Of this, half is overhead coming from the function call administration, but the other half is truly related to the speed of the algorithm. Therefore a skipping ahead would take place with 15% of the time cost of the original run, i.e. an uncomfortably high figure.

Instead a different solution is chosen here. Two special routines are provided for writing and reading the state of the random number generator (plus some initialization information) on a sequential file, in a platform-dependent internal representation. The file used for this purpose has to be specified by you, and opened for read and write. A state is written as a single record, in free format. It is possible to write an arbitrary number of states on a file, and a record can be overwritten, if so desired. The event generation loop might then look something like:

1. save the state of the generator on file (using flag set in point 3 below),
2. generate an event,
3. study the event for errors or other reasons why to regenerate it later; set flag to overwrite previous generator state if no errors, otherwise set flag to create new record;
4. loop back to point 1.

With this procedure, the file will contain the state before each of the problematical events. These events can therefore be generated in a shorter run, where further information can be printed. (Inside PYTHIA, some initialization may take place in connection with the very first event generated in a run, so it may be necessary to generate one ordinary event before reading in a saved state to generate the interesting events.) An alternative approach might be to save the state every 100 events or so. If the events are subsequently processed through a detector simulation, you may have to save also other sets of seeds, naturally.

Unfortunately, the procedure is not always going to work. For instance, if cross section maximum violations have occured before the interesting event in the original run, there is a possibility that another event is picked in the re-started one, where the maximum weight estimate has not been updated. Another problem is the multiple interaction machinery, where some of the options contain an element of learning, which again means that the event sequence may be broken.

In addition to the service routines, the common block which contains the state of the generator is available for manipulation, if you so desire. In particular, the initial seed value is by default 19780503, i.e. different from the Marsaglia/CERN default 54217137. It is possible to change this value before any random numbers have been generated, or to force re-initialization in mid-run with any desired new seed.

It should be noted that, of course, the appearance of a random number generator package inside PYTHIA does in no way preclude the use of other routines. You can easily revert to having PYR as nothing but an interface to an arbitrary external random number generator; e.g. to call a routine RNDM all you need to have is

```
      FUNCTION PYR(IDUMMY)
      IMPLICIT DOUBLE PRECISION(A-H, O-Z)
  100 PYR=RNDM(IDUMMY)
      IF(PYR.LE.0D0.OR.PYR.GE.1D0) GOTO 100
      RETURN
      END
```

The random generator subpackage consists of the following components.

```
R = PYR(IDUMMY)
```

Purpose: to generate a (pseudo)random number R uniformly in the range 0<R<1, i.e. excluding the endpoints.

IDUMMY : dummy input argument; normally 0.

```
CALL PYRGET(LFN,MOVE)
```

Purpose: to dump the current state of the random number generator on a separate file, using internal representation for real and integer numbers. To be precise, the full contents of the PYDATR common block are written on the file, with the exception of MRPY(6).

LFN : (logical file number) the file number to which the state is dumped. You must associate this number with a true file (with a platform-dependent name), and see to it that this file is open for write.

MOVE : choice of adding a new record to the file or overwriting old record(s). Normally only options 0 or -1 should be used.

- = 0 (or > 0) : add a new record to the end of the file.
- = -1 : overwrite the last record with a new one (i.e. do one BACKSPACE before the new write).
- = $-n$: back up n records before writing the new record. The records following after the new one are lost, i.e. the last n old records are lost and one new added.

```
CALL PYRSET(LFN,MOVE)
```

Purpose: to read in a state for the random number generator, from which the subsequent generation can proceed. The state must previously have been saved by a PYRGET call. Again the full contents of the PYDATR common block are read, with the exception of MRPY(6).

LFN : (logical file number) the file number from which the state is read. You must associate this number with a true file previously written with a PYRGET call, and see to it that this file is open for read.

MOVE : positioning in file before a record is read. With zero value, records are read one after the other for each new call, while non-zero values may be used to navigate back and forth, and e.g. return to the same initial state several times.

- = 0 : read the next record.
- = $+n$: skip ahead n records before reading the record that sets the state of the random number generator.
- = $-n$: back up n records before reading the record that sets the state of the random number generator.

```
COMMON/PYDATR/MRPY(6),RRPY(100)
```

Purpose: to contain the state of the random number generator at any moment (for communication between PYR, PYRGET and PYRSET), and also to provide you with the possibility to initialize different random number sequences, and to know how many numbers have been generated.

MRPY(1) : (D=19780503) the integer number that specifies which of the possible subsequences will be initialized in the next PYR call for which MRPY(2)=0. Allowed values are $0 \leq$ MRPY(1) $\leq 900\,000\,000$, the original Marsaglia (and CERN library) seed is 54217137. The MRPY(1) value is not changed by any of the PYTHIA routines.

MRPY(2) : (D=0) initialization flag, put to 1 in the first PYR call of run. A re-initialization of the random number generator can be made in mid-run by resetting MRPY(2) to 0 by hand. In addition, any time the counter MRPY(3) reaches 1000000000, it is reset to 0 and MRPY(2) is increased by 1.

MRPY(3) : (R) counter for the number of random numbers generated from the beginning of the run. To avoid overflow when very many numbers are generated, MRPY(2) is

used as described above.

`MRPY(4), MRPY(5) : I97` and `J97` of the CERN library implementation; part of the state of the generator.

`MRPY(6) :` (R) current position, i.e. how many records after beginning, in the file; used by `PYRGET` and `PYRSET`.

`RRPY(1) - RRPY(97) :` the `U` array of the CERN library implementation; part of the state of the generator.

`RRPY(98) - RRPY(100) : C`, `CD` and `CM` of the CERN library implementation; the first part of the state of the generator, the latter two constants calculated at initialization.

5 The Event Record

The event record is the central repository for information about the particles produced in the current event: flavours, momenta, event history, and production vertices. It plays a very central rôle: without a proper understanding of what the record is and how information is stored, it is meaningless to try to use PYTHIA. The record is stored in the common block PYJETS. Almost all the routines that the user calls can be viewed as performing some action on the record: fill a new event, let partons fragment or particles decay, boost it, list it, find clusters, etc.

In this section we will first describe the KF flavour code, subsequently the PYJETS common block, and then give a few comments about the rôle of the event record in the programs.

To ease the interfacing of different event generators, a HEPEVT standard common block structure for the event record has been agreed on. For historical reasons the standard common blocks are not directly used in PYTHIA, but a conversion routine comes with the program, and is described at the end of this section.

5.1 Particle Codes

The Particle Data Group particle code [PDG88, PDG92, PDG00] is used consistently throughout the program. Almost all known discrepancies between earlier versions of the PDG standard and the PYTHIA usage have now been resolved. The one known exception is the (very uncertain) classification of $f_0(980)$, with $f_0(1370)$ also affected as a consequence. There is also some slight mixup in the technicolor sector between π'^0_{tc} and η_{tc}. These should not be major problems. The PDG standard, with the local PYTHIA extensions, is referred to as the KF particle code. This code you have to be thoroughly familiar with. It is described below.

The KF code is not convenient for a direct storing of masses, decay data, or other particle properties, since the KF codes are so spread out. Instead a compressed code KC between 1 and 500 is used here. A particle and its antiparticle are mapped to the same KC code, but else the mapping is unique. Normally this code is only used at very specific places in the program, not visible to the user. If need be, the correspondence can always be obtained by using the function PYCOMP, i.e. `KC = PYCOMP(KF)`. This mapping is not hardcoded, but can be changed by user intervention, e.g. by introducing new particles with the PYUPDA facility. It is therefore not intended that you should ever want or need to know any KC codes at all. It may be useful to know, however, that for codes smaller than 80, KF and KC agree. Normally a user would never do the inverse mapping, but we note that this is stored as `KF = KCHG(KC,4)`, making use of the KCHG array in the PYDAT2 common block. Of course, the sign of a particle could never be recovered by this inverse operation.

The particle names printed in the tables in this section correspond to the ones obtained with the routine PYNAME, which is used extensively, e.g. in PYLIST. Greek characters are spelt out in full, with a capital first letter to correspond to a capital Greek letter. Generically the name of a particle is made up of the following pieces:

1. The basic root name. This includes a * for most spin 1 ($L = 0$) mesons and spin $3/2$ baryons, and a $'$ for some spin $1/2$ baryons (where there are two states to be distinguished, cf. Λ–Σ^0). The rules for heavy baryon naming are in accordance with the 1986 Particle Data Group conventions [PDG86]. For mesons with one unit of orbital angular momentum, K (D, B, ...) is used for quark-spin 0 and K* (D*, B*, ...) for quark-spin 1 mesons; the convention for '*' may here deviate slightly from the one used by the PDG.
2. Any lower indices, separated from the root by a _. For heavy hadrons, this is the additional heavy-flavour content not inherent in the root itself. For a diquark, it is the spin.
3. The characters 'bar' for an antiparticle, wherever the distinction between particle and antiparticle is not inherent in the charge information.
4. Charge information: $++$, $+$, 0, $-$, or $--$. Charge is not given for quarks or diquarks. Some neutral particles which are customarily given without a 0 also here lack it, such

Table 5: Quark and lepton codes.

KF	Name	Printed	KF	Name	Printed
1	d	d	11	e^-	e-
2	u	u	12	ν_e	nu_e
3	s	s	13	μ^-	mu-
4	c	c	14	ν_μ	nu_mu
5	b	b	15	τ^-	tau-
6	t	t	16	ν_τ	nu_tau
7	b'	b'	17	τ'	tau'
8	t'	t'	18	ν'_τ	nu'_tau
9			19		
10			20		

as neutrinos, g, γ, and flavour-diagonal mesons other than π^0 and ρ^0. Note that charge is included both for the proton and the neutron. While non-standard, it is helpful in avoiding misunderstandings when looking at an event listing.

Below follows a list of KF particle codes. The list is not complete; a more extensive one may be obtained with CALL PYLIST(11). Particles are grouped together, and the basic rules are described for each group. Whenever a distinct antiparticle exists, it is given the same KF code with a minus sign (whereas KC codes are always positive).

1. Quarks and leptons, Table 5.
   This group contains the basic building blocks of matter, arranged according to family, with the lower member of weak isodoublets also having the smaller code (thus d precedes u). A fourth generation is included as part of the scenarios for exotic physics. The quark codes are used as building blocks for the diquark, meson and baryon codes below.
2. Gauge bosons and other fundamental bosons, Table 6.
   This group includes all the gauge and Higgs bosons of the standard model, as well as some of the bosons appearing in various extensions of it. They correspond to one extra **U(1)** and one extra **SU(2)** group, a further Higgs doublet, a graviton, a horizontal gauge boson R (coupling between families), and a (scalar) leptoquark L_Q.
3. Exotic particle codes.
   The positions 43–80 are used as temporary sites for exotic particles that eventually may be shifted to a separate code sequence. Currently this list is empty. The ones not in use are at your disposal (but with no guarantees that they will remain so).
4. Various special codes, Table 7.
   In a Monte Carlo, it is always necessary to have codes that do not correspond to any specific particle, but are used to lump together groups of similar particles for decay treatment (nowadays largely obsolete), to specify generic decay products (also obsolete), or generic intermediate states in external processes, or additional event record information from jet searches. These codes, which again are non-standard, are found between numbers 81 and 100.
5. Diquark codes, Table 8.
   A diquark made up of a quark with code i and another with code j, where $i \geq j$, and with total spin s, is given the code

$$\mathtt{KF} = 1000i + 100j + 2s + 1 \,, \tag{16}$$

   i.e. the tens position is left empty (cf. the baryon code below). Some of the most frequently used codes are listed in the table. All the lowest-lying spin 0 and 1 diquarks are

Table 6: Gauge boson and other fundamental boson codes.

KF	Name	Printed	KF	Name	Printed
21	g	g	31		
22	γ	gamma	32	Z'^0	Z'0
23	Z^0	Z0	33	Z''^0	Z"0
24	W^+	W+	34	W'^+	W'+
25	h^0	h0	35	H^0	H0
26			36	A^0	A0
27			37	H^+	H+
28			38		
29			39	G	Graviton
30			40		
			41	R^0	R0
			42	L_Q	LQ

Table 7: Various special codes.

KF	Printed	Meaning
81	specflav	Spectator flavour; used in decay-product listings
82	rndmflav	A random u, d, or s flavour; possible decay product
83	phasespa	Simple isotropic phase-space decay
84	c-hadron	Information on decay of generic charm hadron
85	b-hadron	Information on decay of generic bottom hadron
86		
87		
88		
89		(internal use for unspecified resonance data)
90	system	Intermediate pseudoparticle in external process
91	cluster	Parton system in cluster fragmentation
92	string	Parton system in string fragmentation
93	indep.	Parton system in independent fragmentation
94	CMshower	Four-momentum of time-like showering system
95	SPHEaxis	Event axis found with PYSPHE
96	THRUaxis	Event axis found with PYTHRU
97	CLUSjet	Jet (cluster) found with PYCLUS
98	CELLjet	Jet (cluster) found with PYCELL
99	table	Tabular output from PYTABU
100		

Table 8: Diquark codes. For brevity, diquarks containing c or b quarks are not listed, but are defined analogously.

KF	Name	Printed	KF	Name	Printed
			1103	dd_1	dd_1
2101	ud_0	ud_0	2103	ud_1	ud_1
			2203	uu_1	uu_1
3101	sd_0	sd_0	3103	sd_1	sd_1
3201	su_0	su_0	3203	su_1	su_1
			3303	ss_1	ss_1

included in the program.

6. Meson codes, Tables 9 and 10.
   A meson made up of a quark with code i and an antiquark with code $-j$, $j \neq i$, and with total spin s, is given the code

$$\mathtt{KF} = \{100\max(i,j) + 10\min(i,j) + 2s + 1\}\ \mathrm{sign}(i-j)\,(-1)^{\max(i,j)}\ , \tag{17}$$

   assuming it is not orbitally or radially excited. Note the presence of an extra $-$ sign if the heaviest quark is a down-type one. This is in accordance with the particle–antiparticle distinction adopted in the 1986 Review of Particle Properties [PDG86]. It means for example that a B meson contains a $\bar{b}$ antiquark rather than a b quark.
   The flavour-diagonal states are arranged in order of ascending mass. Thus the obvious generalization of eq. (17) to $\mathtt{KF} = 110i + 2s + 1$ is only valid for charm and bottom. The lighter quark states can appear mixed, e.g. the π^0 (111) is an equal mixture of $d\bar{d}$ (naively code 111) and $u\bar{u}$ (naively code 221).
   The standard rule of having the last digit of the form $2s+1$ is broken for the K^0_S–K^0_L system, where it is 0, and this convention should carry over to mixed states in the B meson system, should one choose to define such. For higher multiplets with the same spin, ± 10000, ± 20000, etc., are added to provide the extra distinction needed. Some of the most frequently used codes are given below.
   The full lowest-lying pseudoscalar and vector multiplets are included in the program, Table 9.
   Also the lowest-lying orbital angular momentum $L = 1$ mesons are included, Table 10: one pseudovector multiplet obtained for total quark-spin 0 ($L = 1, S = 0 \Rightarrow J = 1$) and one scalar, one pseudovector and one tensor multiplet obtained for total quark-spin 1 ($L = 1, S = 1 \Rightarrow J = 0, 1$ or 2), where J is what is conventionally called the spin s of the meson. Any mixing between the two pseudovector multiplets is not taken into account. Please note that some members of these multiplets have still not been found, and are included here only based on guesswork. Even for known ones, the information on particles (mass, width, decay modes) is highly incomplete.
   Only two radial excitations are included, the $\psi' = \psi(2S)$ and $\Upsilon' = \Upsilon(2S)$.
7. Baryon codes, Table 11.
   A baryon made up of quarks i, j and k, with $i \geq j \geq k$, and total spin s, is given the code

$$\mathtt{KF} = 1000i + 100j + 10k + 2s + 1\ . \tag{18}$$

   An exception is provided by spin $1/2$ baryons made up of three different types of quarks, where the two lightest quarks form a spin-0 diquark (Λ-like baryons). Here the order of the j and k quarks is reversed, so as to provide a simple means of distinction to baryons with the lightest quarks in a spin-1 diquark (Σ-like baryons).

Table 9: Meson codes, part 1.

KF	Name	Printed	KF	Name	Printed
211	π^+	pi+	213	ρ^+	rho+
311	K^0	K0	313	K^{*0}	K*0
321	K^+	K+	323	K^{*+}	K*+
411	D^+	D+	413	D^{*+}	D*+
421	D^0	D0	423	D^{*0}	D*0
431	D_s^+	D_s+	433	D_s^{*+}	D*_s+
511	B^0	B0	513	B^{*0}	B*0
521	B^+	B+	523	B^{*+}	B*+
531	B_s^0	B_s0	533	B_s^{*0}	B*_s0
541	B_c^+	B_c+	543	B_c^{*+}	B*_c+
111	π^0	pi0	113	ρ^0	rho0
221	η	eta	223	ω	omega
331	η'	eta'	333	ϕ	phi
441	η_c	eta_c	443	J/ψ	J/psi
551	η_b	eta_b	553	Υ	Upsilon
130	K_L^0	K_L0			
310	K_S^0	K_S0			

For hadrons with heavy flavours, the root names are Lambda or Sigma for hadrons with two u or d quarks, Xi for those with one, and Omega for those without u or d quarks.
Some of the most frequently used codes are given in Table 11. The full lowest-lying spin $1/2$ and $3/2$ multiplets are included in the program.

8. QCD effective states, Table 12.
   We here include the pomeron $\mathbb{P}$ and reggeon $\mathbb{R}$ 'particles', which are important e.g. in the description of diffractive scattering, but do not have a simple correspondence with other particles in the classification scheme.
   Also included are codes to be used for denoting diffractive states in PYTHIA, as part of the event history. The first two digits here are 99 to denote the non-standard character. The second, third and fourth last digits give flavour content, while the very last one is 0, to denote the somewhat unusual character of the code. Only a few codes have been introduced with names; depending on circumstances these also have to double up for other diffractive states. Other diffractive codes for strange mesons and baryon beams are also accepted by the program, but do not give nice printouts.
9. Supersymmetric codes, Table 13.
   SUSY doubles the number of states of the Standard Model (at least). Fermions have separate spartners to the left- and right-handed components. In the third generation these are assumed to mix to nontrivial mass eigenstates, while mixing is not included in the first two. Note that all sparticle names begin with a tilde. Default masses are arbitrary and branching ratios not set at all. This is taken care of at initialization if `IMSS(1)` is positive.
10. Technicolor codes, Table 14.
    A set of colourless and coloured technihadrons have been included, the latter specifically for the case of Topcolor assisted Technicolor. Where unclear, indices 1 or 8 denote the colour multiplet. Then there are coloured technirhos and technipions that can mix with the Coloron (or V_8) associated with the breaking of $\mathbf{SU(3)}_2 \times \mathbf{SU(3)}_3$ to ordinary $\mathbf{SU(3)}_C$

Table 10: Meson codes, part 2. For brevity, states with b quark are omitted from this listing, but are defined in the program.

KF	Name	Printed	KF	Name	Printed
10213	b_1	b_1+	10211	a_0^+	a_0+
10313	K_1^0	K_10	10311	K_0^{*0}	K*_00
10323	K_1^+	K_1+	10321	K_0^{*+}	K*_0+
10413	D_1^+	D_1+	10411	D_0^{*+}	D*_0+
10423	D_1^0	D_10	10421	D_0^{*0}	D*_00
10433	D_{1s}^+	D_1s+	10431	D_{0s}^{*+}	D*_0s+
10113	b_1^0	b_10	10111	a_0^0	a_00
10223	h_1^0	h_10	10221	f_0^0	f_00
10333	$h_1'^0$	h'_10	10331	$f_0'^0$	f'_00
10443	h_{1c}^0	h_1c0	10441	χ_{0c}^0	chi_0c0
20213	a_1^+	a_1+	215	a_2^+	a_2+
20313	K_1^{*0}	K*_10	315	K_2^{*0}	K*_20
20323	K_1^{*+}	K*_1+	325	K_2^{*+}	K*_2+
20413	D_1^{*+}	D*_1+	415	D_2^{*+}	D*_2+
20423	D_1^{*0}	D*_10	425	D_2^{*0}	D*_20
20433	D_{1s}^{*+}	D*_1s+	435	D_{2s}^{*+}	D*_2s+
20113	a_1^0	a_10	115	a_2^0	a_20
20223	f_1^0	f_10	225	f_2^0	f_20
20333	$f_1'^0$	f'_10	335	$f_2'^0$	f'_20
20443	χ_{1c}^0	chi_1c0	445	χ_{2c}^0	chi_2c0
100443	ψ'	psi'			
100553	Υ'	Upsilon'			

(where the 2 and 3 indices refer to the first two and the third generation, respectively).
The η_{tc} belongs to an older iteration of Technicolor modelling than the rest. It was originally given the 3000221 code, and thereby now comes to clash with the π'^0_{tc} of the current main scenario. Since the η_{tc} is one-of-a-kind, it was deemed better to move it to make way for the π'^0_{tc}. This leads to a slight inconsistency with the PDG codes.

11. Excited fermion codes, Table 15.
A first generation of excited fermions are included.
12. Exotic particle codes, Table 16.
This section includes the excited graviton, as the first (but probably not last) manifestation of the possibility of large extra dimensions. Although it is not yet in the PDG standard, we assume that such states will go in a new series of numbers.
Included is also a set of particles associated with an extra **SU(2)** gauge group for righthanded states, as required in order to obtain a left–right symmetric theory at high energies. This includes righthanded (Majorana) neutrinos, righthanded Z_R^0 and $W_R^\pm$ gauge bosons, and both left- and righthanded doubly charged Higgses. Such a scenario would also contain other Higgs states, but these do not bring anything new relative to the ones already introduced, from an observational point of view. Here the first two digits are 99 to denote the non-standard character.

A hint on large particle numbers: if you want to avoid mistyping the number of zeros, it may pay off to define a statement like

Table 11: Baryon codes. For brevity, some states with b quarks or multiple c ones are omitted from this listing, but are defined in the program.

KF	Name	Printed	KF	Name	Printed
			1114	Δ^-	Delta-
2112	n	n0	2114	Δ^0	Delta0
2212	p	p+	2214	Δ^+	Delta+
			2224	Δ^{++}	Delta++
3112	Σ^-	Sigma-	3114	Σ^{*-}	Sigma*-
3122	Λ^0	Lambda0			
3212	Σ^0	Sigma0	3214	Σ^{*0}	Sigma*0
3222	Σ^+	Sigma+	3224	Σ^{*+}	Sigma*+
3312	Ξ^-	Xi-	3314	Ξ^{*-}	Xi*-
3322	Ξ^0	Xi0	3324	Ξ^{*0}	Xi*0
			3334	Ω^-	Omega-
4112	Σ_c^0	Sigma_c0	4114	Σ_c^{*0}	Sigma*_c0
4122	Λ_c^+	Lambda_c+			
4212	Σ_c^+	Sigma_c+	4214	Σ_c^{*+}	Sigma*_c+
4222	Σ_c^{++}	Sigma_c++	4224	Σ_c^{*++}	Sigma*_c++
4132	Ξ_c^0	Xi_c0			
4312	$\Xi_c'^0$	Xi'_c0	4314	Ξ_c^{*0}	Xi*_c0
4232	Ξ_c^+	Xi_c+			
4322	$\Xi_c'^+$	Xi'_c+	4324	Ξ_c^{*+}	Xi*_c+
4332	Ω_c^0	Omega_c0	4334	Ω_c^{*0}	Omega*_c0
5112	Σ_b^-	Sigma_b-	5114	Σ_b^{*-}	Sigma*_b-
5122	Λ_b^0	Lambda_b0			
5212	Σ_b^0	Sigma_b0	5214	Σ_b^{*0}	Sigma*_b0
5222	Σ_b^+	Sigma_b+	5224	Σ_b^{*+}	Sigma*_b+

Table 12: QCD effective states.

KF	Printed	Meaning
110	reggeon	reggeon $\mathbb{R}$
990	pomeron	pomeron $\mathbb{P}$
9900110	rho_diff0	Diffractive $\pi^0/\rho^0/\gamma$ state
9900210	pi_diffr+	Diffractive π^+ state
9900220	omega_di0	Diffractive ω state
9900330	phi_diff0	Diffractive ϕ state
9900440	J/psi_di0	Diffractive J/ψ state
9902110	n_diffr	Diffractive n state
9902210	p_diffr+	Diffractive p state

Table 13: Supersymmetric codes.

KF	Name	Printed	KF	Name	Printed
1000001	$\tilde{d}_L$	~d_L	2000001	$\tilde{d}_R$	~d_R
1000002	$\tilde{u}_L$	~u_L	2000002	$\tilde{u}_R$	~u_R
1000003	$\tilde{s}_L$	~s_L	2000003	$\tilde{s}_R$	~s_R
1000004	$\tilde{c}_L$	~c_L	2000004	$\tilde{c}_R$	~c_R
1000005	$\tilde{b}_1$	~b_1	2000005	$\tilde{b}_2$	~b_2
1000006	$\tilde{t}_1$	~t_1	2000006	$\tilde{t}_2$	~t_2
1000011	$\tilde{e}_L$	~e_L-	2000011	$\tilde{e}_R$	~e_R-
1000012	$\tilde{\nu}_{eL}$	~nu_eL	2000012	$\tilde{\nu}_{eR}$	~nu_eR
1000013	$\tilde{\mu}_L$	~mu_L-	2000013	$\tilde{\mu}_R$	~mu_R-
1000014	$\tilde{\nu}_{\mu L}$	~nu_muL	2000014	$\tilde{\nu}_{\mu R}$	~nu_muR
1000015	$\tilde{\tau}_1$	~tau_L-	2000015	$\tilde{\tau}_2$	~tau_R-
1000016	$\tilde{\nu}_{\tau L}$	~nu_tauL	2000016	$\tilde{\nu}_{\tau R}$	~nu_tauR
1000021	$\tilde{g}$	~g	1000025	$\tilde{\chi}_3^0$	~chi_30
1000022	$\tilde{\chi}_1^0$	~chi_10	1000035	$\tilde{\chi}_4^0$	~chi_40
1000023	$\tilde{\chi}_2^0$	~chi_20	1000037	$\tilde{\chi}_2^+$	~chi_2+
1000024	$\tilde{\chi}_1^+$	~chi_1+	1000039	$\tilde{G}$	~Gravitino

Table 14: Technicolor codes.

KF	Name	Printed	KF	Name	Printed
3000111	π^0_{tc}	pi_tc0	3100021	$V_{8,tc}$	V8_tc
3000211	π^+_{tc}	pi_tc+	3100111	$\pi^0_{22,1,tc}$	pi_22_1_tc
3000221	π'^0_{tc}	pi'_tc0	3200111	$\pi^0_{22,8,tc}$	pi_22_8_tc
3000113	ρ^0_{tc}	rho_tc0	3100113	$\rho^0_{11,tc}$	rho_11_tc
3000213	ρ^+_{tc}	rho_tc+	3200113	$\rho^0_{12,tc}$	rho_12_tc
3000223	ω^0_{tc}	omega_tc0	3300113	$\rho^0_{21,tc}$	rho_21_tc
3000331	η_{tc}	eta_tc0	3400113	$\rho^0_{22,tc}$	rho_22_tc

Table 15: Excited fermion codes.

KF	Name	Printed	KF	Name	Printed
4000001	u^*	d*	4000011	e^*	e*-
4000002	d^*	u*	4000012	ν_e^*	nu*_e0

Table 16: Exotic particle codes.

KF	Name	Printed	KF	Name	Printed
5000039	G^*	Graviton*			
9900012	ν_{Re}	nu_Re	9900023	Z_R^0	Z_R0
9900014	$\nu_{R\mu}$	nu_Rmu	9900024	W_R^+	W_R+
9900016	$\nu_{R\tau}$	nu_Rtau	9900041	H_L^{++}	H_L++
			9900042	H_R^{++}	H_R++

```
 PARAMETER (KSUSY1=1000000,KSUSY2=2000000,KTECHN=3000000,
&KEXCIT=4000000,KDIMEN=5000000)
```

at the beginning of your program and then refer to particles as KSUSY1+1 = $\tilde{d}_L$ and so on. This then also agrees with the internal notation (where feasible).

5.2 The Event Record

Each new event generated is in its entirety stored in the common block PYJETS, which thus forms the event record. Here each parton or particle that appears at some stage of the fragmentation or decay chain will occupy one line in the matrices. The different components of this line will tell which parton/particle it is, from where it originates, its present status (fragmented/decayed or not), its momentum, energy and mass, and the space–time position of its production vertex. Note that K(I,3)–K(I,5) and the P and V vectors may take special meaning for some specific applications (e.g. sphericity or cluster analysis), as described in those connections.

The event history information stored in K(I,3)–K(I,5) should not be taken too literally. In the particle decay chains, the meaning of a mother is well-defined, but the fragmentation description is more complicated. The primary hadrons produced in string fragmentation come from the string as a whole, rather than from an individual parton. Even when the string is not included in the history (see MSTU(16)), the pointer from hadron to parton is deceptive. For instance, in a $qg\bar{q}$ event, those hadrons are pointing towards the q ($\bar{q}$) parton that were produced by fragmentation from that end of the string, according to the random procedure used in the fragmentation routine. No particles point to the g. This assignment seldom agrees with the visual impression, and is not intended to.

The common block PYJETS has expanded with time, and can now house 4000 entries. This figure may seem ridiculously large, but actually the previous limit of 2000 was often reached in studies of high-$p_\perp$ processes at the LHC (and SSC). This is because the event record contains not only the final particles, but also all intermediate partons and hadrons, which subsequently showered, fragmented or decayed. Included are also a wealth of photons coming from π^0 decays; the simplest way of reducing the size of the event record is actually to switch off π^0 decays by MDCY(PYCOMP(111),1)=0. Also note that some routines, such as PYCLUS and PYCELL, use memory after the event record proper as a working area. Still, to change the size of the common block, upwards or downwards, is easy: just do a global substitute in the common block and change the MSTU(4) value to the new number. If more than 10000 lines are to be used, the packing of colour information should also be changed, see MSTU(5).

```
COMMON/PYJETS/N,NPAD,K(4000,5),P(4000,5),V(4000,5)
```

Purpose: to contain the event record, i.e. the complete list of all partons and particles (initial, intermediate and final) in the current event. (By parton we here mean the subclass of particles that carry colour, for which extra colour flow information is then required.

Normally this means quarks and gluons, which can fragment to hadrons, but also squarks and other exotic particles fall in this category.)

`N` : number of lines in the `K`, `P` and `V` matrices occupied by the current event. `N` is continuously updated as the definition of the original configuration and the treatment of fragmentation and decay proceed. In the following, the individual parton/particle number, running between 1 and `N`, is called `I`.

`NPAD` : dummy to ensure an even number of integers before the double precision reals, as required by some compilers.

`K(I,1)` : status code KS, which gives the current status of the parton/particle stored in the line. The ground rule is that codes 1–10 correspond to currently existing partons/particles, while larger codes contain partons/particles which no longer exist, or other kinds of event information.

- `= 0` : empty line.
- `= 1` : an undecayed particle or an unfragmented parton, the latter being either a single parton or the last one of a parton system.
- `= 2` : an unfragmented parton, which is followed by more partons in the same colour-singlet parton system.
- `= 3` : an unfragmented parton with special colour flow information stored in `K(I,4)` and `K(I,5)`, such that adjacent partons along the string need not follow each other in the event record.
- `= 4` : a particle which could have decayed, but did not within the allowed volume around the original vertex.
- `= 5` : a particle which is to be forced to decay in the next `PYEXEC` call, in the vertex position given (this code is only set by user intervention).
- `= 11` : a decayed particle or a fragmented parton, the latter being either a single parton or the last one of a parton system, cf. `=1`.
- `= 12` : a fragmented parton, which is followed by more partons in the same colour-singlet parton system, cf. `=2`. Further, a B meson which decayed as a $\overline{\mathrm{B}}$ one, or vice versa, because of B–$\overline{\mathrm{B}}$ mixing, is marked with this code rather than `=11`.
- `= 13` : a parton which has been removed when special colour flow information has been used to rearrange a parton system, cf. `=3`.
- `= 14` : a parton which has branched into further partons, with special colour-flow information provided, cf. `=3`.
- `= 15` : a particle which has been forced to decay (by user intervention), cf. `=5`.
- `= 21` : documentation lines used to give a compressed story of the event at the beginning of the event record.
- `= 31` : lines with information on sphericity, thrust or cluster search.
- `= 32` : tabular output, as generated by `PYTABU`.
- `= 41` : junction (currently not fully implemented).
- `< 0` : these codes are never used by the program, and are therefore usually not affected by operations on the record, such as `PYROBO`, `PYLIST` and event-analysis routines (the exception is some `PYEDIT` calls, where lines are moved but not deleted). Such codes may therefore be useful in some connections.

`K(I,2)` : particle `KF` code, as described in section 5.1.

`K(I,3)` : line number of parent particle, where known, otherwise 0. Note that the assignment of a particle to a given parton in a parton system is unphysical, and what is given there is only related to the way the fragmentation was generated.

`K(I,4)` : normally the line number of the first daughter; it is 0 for an undecayed particle or unfragmented parton.

For `K(I,1) = 3, 13` or `14`, instead, it contains special colour-flow information (for internal use only) of the form

`K(I,4)` = 200000000*MCFR + 100000000*MCTO + 10000*ICFR + ICTO,

where ICFR and ICTO give the line numbers of the partons from which the colour comes and to where it goes, respectively; MCFR and MCTO originally are 0 and are set to 1 when the corresponding colour connection has been traced in the PYPREP rearrangement procedure. (The packing may be changed with MSTU(5).) The 'from' colour position may indicate a parton which branched to produce the current parton, or a parton created together with the current parton but with matched anticolour, while the 'to' normally indicates a parton that the current parton branches into. Thus, for setting up an initial colour configuration, it is normally only the 'from' part that is used, while the 'to' part is added by the program in a subsequent call to parton-shower evolution (for final-state radiation; it is the other way around for initial-state radiation).

Note: normally most users never have to worry about the exact rules for colour-flow storage, since this is used mainly for internal purposes. However, when it is necessary to define this flow, it is recommended to use the PYJOIN routine, since it is likely that this would reduce the chances of making a mistake.

K(I,5) : normally the line number of the last daughter; it is 0 for an undecayed particle or unfragmented parton.
For K(I,1) = 3, 13 or 14, instead, it contains special colour-flow information (for internal use only) of the form
K(I,5) = 200000000*MCFR + 100000000*MCTO + 10000*ICFR + ICTO,
where ICFR and ICTO give the line numbers of the partons from which the anti-colour comes and to where it goes, respectively; MCFR and MCTO originally are 0 and are set to 1 when the corresponding colour connection has been traced in the PYPREP rearrangement procedure. For further discussion, see K(I,4).

P(I,1) : p_x, momentum in the x direction, in GeV/c.

P(I,2) : p_y, momentum in the y direction, in GeV/c.

P(I,3) : p_z, momentum in the z direction, in GeV/c.

P(I,4) : E, energy, in GeV.

P(I,5) : m, mass, in GeV/c^2. In parton showers, with space-like virtualities, i.e. where $Q^2 = -m^2 > 0$, one puts P(I,5)$= -Q$.

V(I,1) : x position of production vertex, in mm.

V(I,2) : y position of production vertex, in mm.

V(I,3) : z position of production vertex, in mm.

V(I,4) : time of production, in mm/c ($\approx 3.33 \times 10^{-12}$ s).

V(I,5) : proper lifetime of particle, in mm/c ($\approx 3.33 \times 10^{-12}$ s). If the particle is not expected to decay, V(I,5)=0. A line with K(I,1)=4, i.e. a particle that could have decayed, but did not within the allowed region, has the proper non-zero V(I,5).
In the absence of electric or magnetic fields, or other disturbances, the decay vertex VP of an unstable particle may be calculated as
VP(j) = V(I,j) + V(I,5)*P(I,j)/P(I,5), j = 1–4.

5.3 How The Event Record Works

The event record is the main repository for information about an event. In the generation chain, it is used as a 'scoreboard' for what has already been done and what remains to do. This information can be studied by you, to access information not only about the final state, but also about what came before.

5.3.1 A simple example

The first example of section 3.5 may help to clarify what is going on. When PY2ENT is called to generate a $q\bar{q}$ pair, the quarks are stored in lines 1 and 2 of the event record, respectively. Colour information is set to show that they belong together as a colour singlet. The counter N

is also updated to the value of 2. At no stage is a previously generated event removed. Lines 1 and 2 are overwritten, but lines 3 onwards still contain whatever may have been there before. This does not matter, since N indicates where the 'real' record ends.

As PYEXEC is called, explicitly by you or indirectly by PY2ENT, the first entry is considered and found to be the first parton of a system. Therefore the second entry is also found, and these two together form a colour singlet parton system, which may be allowed to fragment. The 'string' that fragments is put in line 3 and the fragmentation products in lines 4 through 10 (in this particular case). At the same time, the q and $\overline{q}$ in the first two lines are marked as having fragmented, and the same for the string. At this stage, N is 10. Internally in PYEXEC there is another counter with the value 2, which indicates how far down in the record the event has been studied.

This second counter is gradually increased by one. If the entry in the corresponding line can fragment or decay, then fragmentation or decay is performed. The fragmentation/decay products are added at the end of the event record, and N is updated accordingly. The entry is then also marked as having been treated. For instance, when line 3 is considered, the 'string' entry of this line is seen to have been fragmented, and no action is taken. Line 4, a ρ^+, is allowed to decay to $\pi^+\pi^0$; the decay products are stored in lines 11 and 12, and line 4 is marked as having decayed. Next, entry 5 is allowed to decay. The entry in line 6, π^+, is a stable particle (by default) and is therefore passed by without any action being taken.

In the beginning of the process, entries are usually unstable, and N grows faster than the second counter of treated entries. Later on, an increasing fraction of the entries are stable end products, and the rôles are now reversed, with the second counter growing faster. When the two coincide, the end of the record has been reached, and the process can be stopped. All unstable objects have now been allowed to fragment or decay. They are still present in the record, so as to simplify the tracing of the history.

Notice that PYEXEC could well be called a second time. The second counter would then start all over from the beginning, but slide through until the end without causing any action, since all objects that can be treated already have been. Unless some of the relevant switches were changed meanwhile, that is. For instance, if π^0 decays were switched off the first time around but on the second, all the π^0's found in the record would be allowed to decay in the second call. A particle once decayed is not 'undecayed', however, so if the π^0 is put back stable and PYEXEC is called a third time, nothing will happen.

5.3.2 Complete PYTHIA events

In a full-blown event generated with PYTHIA, the usage of PYJETS is more complicated, although the general principles survive. PYJETS is used extensively by many of the generation routines; indeed it provides the bridge between many of them. The PYTHIA event listing begins (optionally) with a few lines of event summary, specific to the hard process simulated and thus not described in the overview above. These specific parts are covered in the following.

In most instances, only the particles actually produced are of interest. For MSTP(125)=0, the event record starts off with the parton configuration existing after hard interaction, initial- and final-state radiation, multiple interactions and beam remnants have been considered. The partons are arranged in colour singlet clusters, ordered as required for string fragmentation. Also photons and leptons produced as part of the hard interaction (e.g. from $q\overline{q} \to g\gamma$ or $u\overline{u} \to Z^0 \to e^+e^-$) appear in this part of the event record. These original entries appear with pointer K(I,3)=0, whereas the products of the subsequent fragmentation and decay have K(I,3) numbers pointing back to the line of the parent.

The standard documentation, obtained with MSTP(125)=1, includes a few lines at the beginning of the event record, which contain a brief summary of the process that has taken place. The number of lines used depends on the nature of the hard process and is stored in MSTI(4) for the current event. These lines all have K(I,1)=21. For all processes, lines 1 and 2 give the two incoming particles. When listed with PYLIST, these two lines will be separated from

subsequent ones by a sequence of '`======`' signs, to improve readability. For diffractive and elastic events, the two outgoing states in lines 3 and 4 complete the list. Otherwise, lines 3 and 4 contain the two partons that initiate the two initial-state parton showers, and 5 and 6 the end products of these showers, i.e. the partons that enter the hard interaction. With initial-state radiation switched off, lines 3 and 5 and lines 4 and 6 are identical. For a simple $2 \to 2$ hard scattering, lines 7 and 8 give the two outgoing partons/particles from the hard interaction, before any final-state radiation. For $2 \to 2$ processes proceeding via an intermediate resonance such as γ^*/Z^0, $W^\pm$ or h^0, the resonance is found in line 7 and the two outgoing partons/particles in 8 and 9. In some cases one of these may be a resonance in its own right, or both of them, so that further pairs of lines are added for subsequent decays. If the decay of a given resonance has been switched off, then no decay products are listed either in this initial summary or in the subsequent ordinary listing. Whenever partons are listed, they are assumed to be on the mass shell for simplicity. The fact that effective masses may be generated by initial- and final-state radiation is taken into account in the actual parton configuration that is allowed to fragment, however. The listing of the event documentation closes with another line made up of '`======`' signs.

A few examples may help clarify the picture. For a single diffractive event $p\overline{p} \to p_{diffr}\overline{p}$, the event record will start with

```
 I K(I,1)   K(I,2) K(I,3)  comment
 1     21     2212      0  incoming p
 2     21    -2212      0  incoming p̄
========================  not part of record; appears in listings
 3     21  9902210      1  outgoing p_diffr
 4     21    -2212      2  outgoing p̄
========================  again not part of record
```

The typical QCD $2 \to 2$ process would be

```
 I K(I,1)   K(I,2) K(I,3)  comment
 1     21     2212      0  incoming p
 2     21    -2212      0  incoming p̄
========================
 3     21        2      1  u picked from incoming p
 4     21       -1      2  d̄ picked from incoming p̄
 5     21       21      3  u evolved to g at hard scattering
 6     21       -1      4  still d̄ at hard scattering
 7     21       21      0  outgoing g from hard scattering
 8     21       -1      0  outgoing d̄ from hard scattering
========================
```

Note that, where well defined, the `K(I,3)` code does contain information as to which side the different partons come from, e.g. above the gluon in line 5 points back to the u in line 3, which points back to the proton in line 1. In the example above, it would have been possible to associate the scattered g in line 7 with the incoming one in line 5, but this is not possible in the general case, consider e.g. $gg \to gg$.

A special case is provided by W^+W^- or Z^0Z^0 fusion to an h^0. Then the virtual W's or Z's are shown in lines 7 and 8, the h^0 in line 9, and the two recoiling quarks (that emitted the bosons) in 10 and 11, followed by the Higgs decay products. Since the W's and Z's are space-like, what is actually listed as the mass for them is $-\sqrt{-m^2}$. Thus W^+W^- fusion to an h^0 in process 8 (not process 124, which is lengthier) might look like

```
 I K(I,1)   K(I,2) K(I,3)  comment
 1     21     2212      0  first incoming p
 2     21     2212      0  second incoming p
========================
 3     21        2      1  u picked from first p
 4     21       21      2  g picked from second p
```

5	21	2	3	still u after initial-state radiation
6	21	-4	4	g evolved to $\overline{c}$
7	21	24	5	space-like W^+ emitted by u quark
8	21	-24	6	space-like W^- emitted by $\overline{c}$ quark
9	21	25	0	Higgs produced by W^+W^- fusion
10	21	1	5	u turned into d by emission of W^+
11	21	-3	6	$\overline{c}$ turned into $\overline{s}$ by emission of W^-
12	21	23	9	first Z^0 coming from decay of h^0
13	21	23	9	second Z^0 coming from decay of h^0
14	21	12	12	ν_e from first Z^0 decay
15	21	-12	12	$\overline{\nu}_e$ from first Z^0 decay
16	21	5	13	b quark from second Z^0 decay
17	21	-5	13	$\overline{b}$ antiquark from second Z^0 decay

==========

Another special case is when a spectrum of virtual photons are generated inside a lepton beam, i.e. when `PYINIT` is called with one or two `'gamma/`*lepton*`'` arguments. Then the documentation section is expanded to reflect the new layer of administration. Positions 1 and 2 contain the original beam particles, e.g. e and p (or e^+ and e^-). In position 3 (and 4 for e^+e^-) is (are) the scattered outgoing lepton(s). Thereafter comes the normal documentation, but starting from the photon rather than a lepton. For ep, this means 4 and 5 are the γ^* and p, 6 and 7 the shower initiators, 8 and 9 the incoming partons to the hard interaction, and 10 and 11 the outgoing ones. Thus the documentation is 3 lines longer (4 for e^+e^-) than normally.

After these lines with the initial information, the event record looks the same as for `MSTP(125)=0`, i.e. first comes the parton configuration to be fragmented and, after another separator line '`======`' in the output (but not the event record), the products of subsequent fragmentation and decay chains. This ordinary listing begins in position `MST(4)+1`. The `K(I,3)` pointers for the partons, as well as leptons and photons produced in the hard interaction, are now pointing towards the documentation lines above, however. In particular, beam remnants point to 1 or 2, depending on which side they belong to, and partons emitted in the initial-state parton showers point to 3 or 4. In the second example above, the partons produced by final-state radiation will be pointing back to 7 and 8; as usual, it should be remembered that a specific assignment to 7 or 8 need not be unique. For the third example, final-state radiation partons will come both from partons 10 and 11 and from partons 16 and 17, and additionally there will be a neutrino–antineutrino pair pointing to 14 and 15.

A hadronic event may contain several (semi)hard interactions, in the multiple interactions scenario. The hardest interaction of an event is shown in the initial section of the event record, while further ones are not. Therefore these extra partons, documented in the main section of the event, do not have a documentation copy to point back to, and so are assigned `K(I,3)=0`.

There exists a third documentation option, `MSTP(125)=2`. Here the history of initial- and final-state parton branchings may be traced, including all details on colour flow. This information has not been optimized for user-friendliness, and cannot be recommended for general usage. With this option, the initial documentation lines are the same. They are followed by blank lines, `K(I,1)=0`, up to line 100 (can be changed in `MSTP(126)`). From line 101 onwards each parton with `K(I,1)=` 3, 13 or 14 appears with special colour-flow information in the `K(I,4)` and `K(I,5)` positions. For an ordinary $2 \to 2$ scattering, the two incoming partons at the hard scattering are stored in lines 101 and 102, and the two outgoing in 103 and 104. The colour flow between these partons has to be chosen according to the proper relative probabilities in cases when many alternatives are possible, see section 8.2.1. If there is initial-state radiation, the two partons in lines 101 and 102 are copied down to lines 105 and 106, from which the initial-state showers are reconstructed backwards step by step. The branching history may be read by noting that, for a branching $a \to bc$, the `K(I,3)` codes of b and c point towards the line number of a. Since the showers are reconstructed backwards, this actually means that parton b would appear in the listing before parton a and c, and hence have a pointer to a position

below itself in the list. Associated time-like partons c may initiate time-like showers, as may the partons of the hard scattering. Again a showering parton or pair of partons will be copied down towards the end of the list and allowed to undergo successive branchings $c \to de$, with d and e pointing towards c. The mass of time-like partons is properly stored in `P(I,5)`; for space-like partons $-\sqrt{-m^2}$ is stored instead. After this section, containing all the branchings, comes the final parton configuration, properly arranged in colour, followed by all subsequent fragmentation and decay products, as usual.

5.4 The HEPEVT Standard

A set of common blocks was developed and agreed on within the framework of the 1989 LEP physics study, see [Sjö89]. This standard defines an event record structure which should make the interfacing of different event generators much simpler.

It would be a major work to rewrite PYTHIA to agree with this standard event record structure. More importantly, the standard only covers quantities which can be defined unambiguously, i.e. which are independent of the particular program used. There are thus no provisions for the need for colour-flow information in models based on string fragmentation, etc., so the standard common blocks would anyway have to be supplemented with additional event information. For the moment, the adopted approach is therefore to retain the `PYJETS` event record, but supply a routine `PYHEPC` which can convert to or from the standard event record. Owing to a somewhat different content in the two records, some ambiguities do exist in the translation procedure. `PYHEPC` has therefore to be used with some judgement.

In this section, the standard event structure is first presented, i.e. the most important points in [Sjö89] are recapitulated. Thereafter the conversion routine is described, with particular attention to ambiguities and limitations.

The standard event record is stored in two common blocks. The second of these is specifically intended for spin information. Since PYTHIA never (explicitly) makes use of spin information, this latter common block is not addressed here. A third common block for colour flow information has been discussed, but never formalized. Note that a `CALL PYLIST(5)` can be used to obtain a simple listing of the more interesting information in the event record.

In order to make the components of the standard more distinguishable in your programs, the three characters `HEP` (for High Energy Physics) have been chosen to be a part of all names.

Originally it was not specified whether real variables should be in single or double precision. At the time, this meant that single precision became the default choice, but since then the trend has been towards increasing precision. In connection with the 1995 LEP 2 workshop, it was therefore agreed to adopt `DOUBLE PRECISION` real variables as part of the standard, and also to extend the size from 2000 to 4000 entries [Kno96]. If, for some reason, one would want to revert to single precision, this would only require trivial changes to the code of the `PYHEPC` conversion routine described below.

```
  PARAMETER (NMXHEP=4000)
  COMMON/HEPEVT/NEVHEP,NHEP,ISTHEP(NMXHEP),IDHEP(NMXHEP),
 &JMOHEP(2,NMXHEP),JDAHEP(2,NMXHEP),PHEP(5,NMXHEP),VHEP(4,NMXHEP)
  DOUBLE PRECISION PHEP, VHEP
```

Purpose: to contain an event record in a Monte Carlo-independent format.

`NMXHEP:` maximum numbers of entries (particles) that can be stored in the common block. The default value of 4000 can be changed via the parameter construction. In the translation, it is checked that this value is not exceeded.

`NEVHEP:` is normally the event number, but may have special meanings, according to the description below:

`> 0 :` event number, sequentially increased by 1 for each call to the main event generation routine, starting with 1 for the first event generated.

`= 0 :` for a program which does not keep track of event numbers, as some of the PYTHIA routines.

`= -1 :` special initialization record; not used by PYTHIA.

`= -2 :` special final record; not used by PYTHIA.

`NHEP:` the actual number of entries stored in the current event. These are found in the first `NHEP` positions of the respective arrays below. Index `IHEP`, 1≤`IHEP`≤`NHEP`, is used below to denote a given entry.

`ISTHEP(IHEP):` status code for entry `IHEP`, with the following meanings:

`= 0 :` null entry.

`= 1 :` an existing entry, which has not decayed or fragmented. This is the main class of entries, which represents the 'final state' given by the generator.

`= 2 :` an entry which has decayed or fragmented and is therefore not appearing in the final state, but is retained for event history information.

`= 3 :` a documentation line, defined separately from the event history. This could include the two incoming reacting particles, etc.

`= 4 - 10 :` undefined, but reserved for future standards.

`= 11 - 200 :` at the disposal of each model builder for constructs specific to his program, but equivalent to a null line in the context of any other program.

`= 201 - :` at the disposal of users, in particular for event tracking in the detector.

`IDHEP(IHEP) :` particle identity, according to the PDG standard. The four additional codes 91–94 have been introduced to make the event history more legible, see section 5.1 and the `MSTU(16)` description of how daughters can point back to them.

`JMOHEP(1,IHEP) :` pointer to the position where the mother is stored. The value is 0 for initial entries.

`JMOHEP(2,IHEP) :` pointer to position of second mother. Normally only one mother exists, in which case the value 0 is to be used. In PYTHIA, entries with codes 91–94 are the only ones to have two mothers. The flavour contents of these objects, as well as details of momentum sharing, have to be found by looking at the mother partons, i.e. the two partons in positions `JMOHEP(1,IHEP)` and `JMOHEP(2,IHEP)` for a cluster or a shower system, and the range `JMOHEP(1,IHEP)`–`JMOHEP(2,IHEP)` for a string or an independent fragmentation parton system.

`JDAHEP(1,IHEP) :` pointer to the position of the first daughter. If an entry has not decayed, this is 0.

`JDAHEP(2,IHEP) :` pointer to the position of the last daughter. If an entry has not decayed, this is 0. It is assumed that daughters are stored sequentially, so that the whole range `JDAHEP(1,IHEP)`–`JDAHEP(2,IHEP)` contains daughters. This variable should be set also when only one daughter is present, as in $K^0 \to K^0_S$ decays, so that looping from the first daughter to the last one works transparently. Normally daughters are stored after mothers, but in backwards evolution of initial-state radiation the opposite may appear, i.e. that mothers are found below the daughters they branch into. Also, the two daughters then need not appear one after the other, but may be separated in the event record.

`PHEP(1,IHEP) :` momentum in the x direction, in GeV/c.

`PHEP(2,IHEP) :` momentum in the y direction, in GeV/c.

`PHEP(3,IHEP) :` momentum in the z direction, in GeV/c.

`PHEP(4,IHEP) :` energy, in GeV.

`PHEP(5,IHEP) :` mass, in GeV/c^2. For space-like partons, it is allowed to use a negative mass, according to `PHEP(5,IHEP)`$= -\sqrt{-m^2}$.

`VHEP(1,IHEP) :` production vertex x position, in mm.

`VHEP(2,IHEP) :` production vertex y position, in mm.

`VHEP(3,IHEP) :` production vertex z position, in mm.

`VHEP(4,IHEP)` : production time, in mm/c ($\approx 3.33 \times 10^{-12}$ s).

This completes the brief description of the standard. In PYTHIA, the routine `PYHEPC` is provided as an interface.

```
CALL PYHEPC(MCONV)
```

Purpose: to convert between the `PYJETS` event record and the `HEPEVT` event record.

`MCONV` : direction of conversion.

- `= 1` : translates the current `PYJETS` record into the `HEPEVT` one, while leaving the original `PYJETS` one unaffected.
- `= 2` : translates the current `HEPEVT` record into the `PYJETS` one, while leaving the original `HEPEVT` one unaffected.

The conversion of momenta is trivial: it is just a matter of exchanging the order of the indices. The vertex information is but little more complicated; the extra fifth component present in `PYJETS` can be easily reconstructed from other information for particles which have decayed. (Some of the advanced features made possible by this component, such as the possibility to consider decays within expanding spatial volumes in subsequent `PYEXEC` calls, cannot be used if the record is translated back and forth, however.) Also, the particle codes `K(I,2)` and `IDHEP(I)` are identical, since they are both based on the PDG codes.

The remaining, non-trivial areas deal with the status codes and the event history. In moving from `PYJETS` to `HEPEVT`, information on colour flow is lost. On the other hand, the position of a second mother, if any, has to be found; this only affects lines with `K(I,2)=` 91–94. Also, for lines with `K(I,1)=` 13 or 14, the daughter pointers have to be found. By and large, however, the translation from `PYJETS` to `HEPEVT` should cause little problem, and there should never be any need for user intervention. (We assume that PYTHIA is run with the default `MSTU(16)=1` mother pointer assignments, otherwise some discrepancies with respect to the proposed standard event history description will be present.)

In moving from `HEPEVT` to `PYJETS`, information on a second mother is lost. Any codes `IDHEP(I)` not equal to 1, 2 or 3 are translated into `K(I,1)=0`, and so all entries with `K(I,1)`≥ 30 are effectively lost in a translation back and forth. All entries with `IDHEP(I)=2` are translated into `K(I,1)=11`, and so entries of type `K(I,1) = 12, 13, 14` or `15` are never found. There is thus no colour-flow information available for partons which have fragmented. For partons with `IDHEP(I)=1`, i.e. which have not fragmented, an attempt is made to subdivide the partonic system into colour singlets, as required for subsequent string fragmentation. To this end, it is assumed that partons are stored sequentially along strings. Normally, a string would then start at a q ($\overline{\mathrm{q}}$) or $\overline{\mathrm{qq}}$ (qq) entry, cover a number of intermediate gluons, and end at a $\overline{\mathrm{q}}$ (q) or qq ($\overline{\mathrm{qq}}$) entry. Particles could be interspersed in this list with no adverse effects, i.e. a $\mathrm{u}-\mathrm{g}-\gamma-\overline{\mathrm{u}}$ sequence would be interpreted as a $\mathrm{u}-\mathrm{g}-\overline{\mathrm{u}}$ string plus an additional photon. A closed gluon loop would be assumed to be made up of a sequential listing of the gluons, with the string continuing from the last gluon up back to the first one. Contrary to the previous, open string case, the appearance of any particle but a gluon would therefore signal the end of the gluon loop. For example, a $\mathrm{g}-\mathrm{g}-\mathrm{g}-\mathrm{g}$ sequence would be interpreted as one single four-gluon loop, while a $\mathrm{g}-\mathrm{g}-\gamma-\mathrm{g}-\mathrm{g}$ sequence would be seen as composed of two 2-gluon systems.

If these interpretations, which are not unique, are not to your liking, it is up to you to correct them, e.g. by using `PYJOIN` to tell exactly which partons should be joined, in which sequence, to give a string. Calls to `PYJOIN` (or the equivalent) are also necessary if `PYSHOW` is to be used to have some partons develop a shower.

For practical applications, one should note that $\mathrm{e}^+\mathrm{e}^-$ events, which have been allowed to shower but not to fragment, do have partons arranged in the order assumed above, so that a translation to `HEPEVT` and back does not destroy the possibility to perform fragmentation by a simple `PYEXEC` call. Also the hard interactions in hadronic events fulfil this condition, while

problems may appear in the multiple interaction scenario, where several closed gg loops may appear directly following one another, and thus would be interpreted as a single multigluon loop after translation back and forth.

6 The Old e^+e^- Annihilation Routines

From the JETSET package, PYTHIA inherits routines for the dedicated simulation of two hard processes in e^+e^- annihilation. The process of main interest is $e^+e^- \to \gamma^*/Z^0 \to q\bar{q}$. The description provided by the PYEEVT routine has been a main staple from PETRA days up to the LEP1 era. Nowadays it is superseded by process 1 of the main PYTHIA event generation machinery, see section 8.4.2. This latter process offers a better description of flavour selection, resonance shape and initial-state radiation. It can also, optionally, be used with the second-order matrix element machinery documented in this section. For backwards compatibility, however, the old routines have still been retained here. There are also a few features found in the routines in this section, and not in the other ones, such as polarized incoming beams.

For the process $e^+e^- \to \gamma^*/Z^0 \to q\bar{q}$, higher-order QCD corrections can be obtained either with parton showers or with second-order matrix elements. The details of the parton-shower evolution are given in section 10, while this section contains the matrix-element description, including a summary of the older algorithm for initial-state photon radiation used here.

The other standalone hard process in this section is Υ decay to ggg or γgg, which is briefly commented on.

The main sources of information for this chapter are refs. [Sjö83, Sjö86, Sjö89].

6.1 Annihilation Events in the Continuum

The description of e^+e^- annihilation into hadronic events involves a number of components: the s dependence of the total cross section and flavour composition, multiparton matrix elements, angular orientation of events, initial-state photon bremsstrahlung and effects of initial-state electron polarization. Many of the published formulae have been derived for the case of massless outgoing quarks. For each of the components described in the following, we will begin by discussing the massless case, and then comment on what is done to accommodate massive quarks.

6.1.1 Electroweak cross sections

In the standard theory, fermions have the following couplings (illustrated here for the first generation):

$$\begin{array}{lll} e_\nu = 0, & v_\nu = 1, & a_\nu = 1, \\ e_e = -1, & v_e = -1 + 4\sin^2\theta_W, & a_e = -1, \\ e_u = 2/3, & v_u = 1 - 8\sin^2\theta_W/3, & a_\nu = 1, \\ e_d = -1/3, & v_d = -1 + 4\sin^2\theta_W/3, & a_d = -1, \end{array}$$

with e the electric charge, and v and a the vector and axial couplings to the Z^0. The relative energy dependence of the weak neutral current to the electromagnetic one is given by

$$\chi(s) = \frac{1}{16\sin^2\theta_W\cos^2\theta_W}\,\frac{s}{s - m_Z^2 + i m_Z \Gamma_Z}\,, \tag{19}$$

where $s = E_{\rm cm}^2$. In this section the electroweak mixing parameter $\sin^2\theta_W$ and the Z^0 mass m_Z and width Γ_Z are considered as constants to be given by you (while the full PYTHIA event generation machinery itself calculates an s-dependent width).

Although the incoming e^+ and e^- beams are normally unpolarized, we have included the possibility of polarized beams, following the formalism of [Ols80]. Thus the incoming e^+ and e^- are characterized by polarizations $\mathbf{P}^\pm$ in the rest frame of the particles:

$$\mathbf{P}^\pm = P_T^\pm \hat{\mathbf{s}}^\pm + P_L^\pm \hat{\mathbf{p}}^\pm\,, \tag{20}$$

where $0 \leq P_{\rm T}^{\pm} \leq 1$ and $-1 \leq P_{\rm L}^{\pm} \leq 1$, with the constraint

$$(\mathbf{P}^{\pm})^2 = (P_{\rm T}^{\pm})^2 + (P_{\rm L}^{\pm})^2 \leq 1 \; . \tag{21}$$

Here $\hat{\mathbf{s}}^{\pm}$ are unit vectors perpendicular to the beam directions $\hat{\mathbf{p}}^{\pm}$. To be specific, we choose a right-handed coordinate frame with $\hat{\mathbf{p}}^{\pm} = (0,0,\mp 1)$, and standard transverse polarization directions (out of the machine plane for storage rings) $\hat{\mathbf{s}}^{\pm} = (0,\pm 1,0)$, the latter corresponding to azimuthal angles $\varphi^{\pm} = \pm\pi/2$. As free parameters in the program we choose $P_{\rm L}^{+}$, $P_{\rm L}^{-}$, $P_{\rm T} = \sqrt{P_{\rm T}^{+} P_{\rm T}^{-}}$ and $\Delta\varphi = (\varphi^{+} + \varphi^{-})/2$.

In the massless QED case, the probability to produce a flavour f is proportional to $e_{\rm f}^2$, i.e up-type quarks are four times as likely as down-type ones. In lowest-order massless QFD (Quantum Flavour Dynamics; part of the Standard Model) the corresponding relative probabilities are given by [Ols80]

$$\begin{aligned} h_{\rm f}(s) \;\; = \;\; & e_{\rm e}^2 (1 - P_{\rm L}^{+} P_{\rm L}^{-}) \, e_{\rm f}^2 + 2 e_{\rm e} \left\{ v_{\rm e} (1 - P_{\rm L}^{+} P_{\rm L}^{-}) - a_{\rm e} (P_{\rm L}^{-} - P_{\rm L}^{+}) \right\} \Re\chi(s) \, e_{\rm f} v_{\rm f} + \\ & + \left\{ (v_{\rm e}^2 + a_{\rm e}^2)(1 - P_{\rm L}^{+} P_{\rm L}^{-}) - 2 v_{\rm e} a_{\rm e} (P_{\rm L}^{-} - P_{\rm L}^{+}) \right\} |\chi(s)|^2 \left\{ v_{\rm f}^2 + a_{\rm f}^2 \right\} \; , \end{aligned} \tag{22}$$

where $\Re\chi(s)$ denotes the real part of $\chi(s)$. The $h_{\rm f}(s)$ expression depends both on the s value and on the longitudinal polarization of the $\rm e^{\pm}$ beams in a non-trivial way.

The cross section for the process $\rm e^+e^- \to \gamma^*/Z^0 \to f\bar{f}$ may now be written as

$$\sigma_{\rm f}(s) = \frac{4\pi\alpha_{\rm em}^2}{3s} R_{\rm f}(s) \; , \tag{23}$$

where $R_{\rm f}$ gives the ratio to the lowest-order QED cross section for the process $\rm e^+e^- \to \mu^+\mu^-$,

$$R_{\rm f}(s) = N_C R_{\rm QCD} \, h_{\rm f}(s) \; . \tag{24}$$

The factor of $N_C = 3$ counts the number of colour states available for the $\rm q\bar{q}$ pair. The $R_{\rm QCD}$ factor takes into account QCD loop corrections to the cross section. For n_f effective flavours (normally $n_f = 5$)

$$R_{\rm QCD} \approx 1 + \frac{\alpha_{\rm s}}{\pi} + (1.986 - 0.115 n_f) \left(\frac{\alpha_{\rm s}}{\pi} \right)^2 + \cdots \tag{25}$$

in the $\overline{\rm MS}$ renormalization scheme [Din79]. Note that $R_{\rm QCD}$ does not affect the relative quark-flavour composition, and so is of peripheral interest here. (For leptons the N_C and $R_{\rm QCD}$ factors would be absent, i.e. $N_C R_{\rm QCD} = 1$, but leptonic final states are not generated by this routine.)

Neglecting higher-order QCD and QFD effects, the corrections for massive quarks are given in terms of the velocity $\beta_{\rm f}$ of a fermion with mass $m_{\rm f}$, $\beta_{\rm f} = \sqrt{1 - 4m_{\rm f}^2/s}$, as follows. The vector quark current terms in $h_{\rm f}$ (proportional to $e_{\rm f}^2$, $e_{\rm f} v_{\rm f}$, or $v_{\rm f}^2$) are multiplied by a threshold factor $\beta_{\rm f}(3 - \beta_{\rm f}^2)/2$, while the axial vector quark current term (proportional to $a_{\rm f}^2$) is multiplied by $\beta_{\rm f}^3$. While inclusion of quark masses in the QFD formulae decreases the total cross section, first-order QCD corrections tend in the opposite direction [Jer81]. Naïvely, one would expect one factor of $\beta_{\rm f}$ to get cancelled. So far, the available options are either to include threshold factors in full or not at all.

Given that all five quarks are light at the scale of the $\rm Z^0$, the issue of quark masses is not really of interest at LEP. Here, however, purely weak corrections are important, in particular since they change the b quark partial width differently from that of the other ones [Küh89]. No such effects are included in the program.

6.1.2 First-order QCD matrix elements

The Born process $e^+e^- \to q\overline{q}$ is modified in first-order QCD by the probability for the q or $\overline{q}$ to radiate a gluon, i.e. by the process $e^+e^- \to q\overline{q}g$. The matrix element is conveniently given in terms of scaled energy variables in the c.m. frame of the event, $x_1 = 2E_q/E_{cm}$, $x_2 = 2E_{\overline{q}}/E_{cm}$, and $x_3 = 2E_g/E_{cm}$, i.e. $x_1 + x_2 + x_3 = 2$. For massless quarks the matrix element reads [Ell76]

$$\frac{1}{\sigma_0}\frac{d\sigma}{dx_1\,dx_2} = \frac{\alpha_s}{2\pi} C_F \frac{x_1^2 + x_2^2}{(1-x_1)(1-x_2)} , \tag{26}$$

where σ_0 is the lowest-order cross section, $C_F = 4/3$ is the appropriate colour factor, and the kinematically allowed region is $0 \le x_i \le 1, i = 1,2,3$. By kinematics, the x_k variable for parton k is related to the invariant mass m_{ij} of the other two partons i and j by $y_{ij} = m_{ij}^2/E_{cm}^2 = 1 - x_k$.

The strong coupling constant α_s is in first order given by

$$\alpha_s(Q^2) = \frac{12\pi}{(33-2n_f)\ln(Q^2/\Lambda^2)} . \tag{27}$$

Conventionally $Q^2 = s = E_{cm}^2$; we will return to this issue below. The number of flavours n_f is 5 for LEP applications, and so the Λ value determined is Λ_5 (while e.g. most Deeply Inelastic Scattering studies refer to Λ_4, the energies for these experiments being below the bottom threshold). The α_s values are matched at flavour thresholds, i.e. as n_f is changed the Λ value is also changed. It is therefore the derivative of α_s that changes at a threshold, not α_s itself.

In order to separate 2-jets from 3-jets, it is useful to introduce jet-resolution parameters. This can be done in several different ways. Most famous are the y and (ε, δ) procedures. We will only refer to the y cut, which is the one used in the program. Here a 3-parton configuration is called a 2-jet event if

$$\min_{i,j}(y_{ij}) = \min_{i,j}\left(\frac{m_{ij}^2}{E_{cm}^2}\right) < y . \tag{28}$$

The cross section in eq. (26) diverges for $x_1 \to 1$ or $x_2 \to 1$ but, when first-order propagator and vertex corrections are included, a corresponding singularity with opposite sign appears in the $q\overline{q}$ cross section, so that the total cross section is finite. In analytical calculations, the average value of any well-behaved quantity Q can therefore be calculated as

$$\langle Q \rangle = \frac{1}{\sigma_{tot}} \lim_{y\to 0}\left(Q(2\text{parton})\,\sigma_{2\text{parton}}(y) + \int_{y_{ij}>y} Q(x_1,x_2)\,\frac{d\sigma_{3\text{parton}}}{dx_1\,dx_2}\,dx_1\,dx_2 \right) , \tag{29}$$

where any explicit y dependence disappears in the limit $y \to 0$.

In a Monte Carlo program, it is not possible to work with a negative total 2-jet rate, and thus it is necessary to introduce a fixed non-vanishing y cut in the 3-jet phase space. Experimentally, there is evidence for the need of a low y cut, i.e. a large 3-jet rate. For LEP applications, the recommended value is $y = 0.01$, which is about as far down as one can go and still retain a positive 2-jet rate. With $\alpha_s = 0.12$, in full second-order QCD (see below), the $2:3:4$ jet composition is then approximately 11% : 77% : 12%.

Note, however, that initial-state QED radiation may occasionally lower the c.m. energy significantly, i.e. increase α_s, and thereby bring the 3-jet fraction above unity if y is kept fixed at 0.01 also in those events. Therefore, at PETRA/PEP energies, y values slightly above 0.01 are needed. In addition to the y cut, the program contains a cut on the invariant mass m_{ij} between any two partons, which is typically required to be larger than 2 GeV. This cut corresponds to the actual merging of two nearby parton jets, i.e. where a treatment with two separate partons

rather than one would be superfluous in view of the smearing arising from the subsequent fragmentation. Since the cut-off mass scale $\sqrt{y}E_{\mathrm{cm}}$ normally is much larger, this additional cut only enters for events at low energies.

For massive quarks, the amount of QCD radiation is slightly reduced [Iof78]:

$$\begin{aligned}\frac{1}{\sigma_0}\frac{\mathrm{d}\sigma}{\mathrm{d}x_1\,\mathrm{d}x_2} &= \frac{\alpha_s}{2\pi}C_F\left\{\frac{x_1^2+x_2^2}{(1-x_1)(1-x_2)}-\frac{4m_q^2}{s}\left(\frac{1}{1-x_1}+\frac{1}{1-x_2}\right)\right.\\ &\left.-\frac{2m_q^2}{s}\left(\frac{1}{(1-x_1)^2}+\frac{1}{(1-x_2)^2}\right)-\frac{4m_q^4}{s^2}\left(\frac{1}{1-x_1}+\frac{1}{1-x_2}\right)^2\right\}\,. \qquad (30)\end{aligned}$$

Properly, the above expression is only valid for the vector part of the cross section, with a slightly different expression for the axial part, but here the one above is used for it all. In addition, the phase space for emission is reduced by the requirement

$$\frac{(1-x_1)(1-x_2)(1-x_3)}{x_3^2}\geq\frac{m_q^2}{s}\,. \qquad (31)$$

For b quarks at LEP energies, these corrections are fairly small.

6.1.3 4-jet matrix elements

Two new event types are added in second-order QCD, $e^+e^-\to q\bar{q}gg$ and $e^+e^-\to q\bar{q}q'\bar{q}'$. The 4-jet cross section has been calculated by several groups [Ali80a, Gae80, Ell81, Dan82], which agree on the result. The formulae are too lengthy to be quoted here. In one of the calculations [Ali80a], quark masses were explicitly included, but here only the massless expressions are included, as taken from [Ell81]. Here the angular orientation of the event has been integrated out, so that five independent internal kinematical variables remain. These may be related to the six y_{ij} and the four y_{ijk} variables, $y_{ij}=m_{ij}^2/s=(p_i+p_j)^2/s$ and $y_{ijk}=m_{ijk}^2/s=(p_i+p_j+p_k)^2/s$, in terms of which the matrix elements are given.

The original calculations were for the pure γ-exchange case; it has been pointed out [Kni89] that an additional contribution to the $e^+e^-\to q\bar{q}q'\bar{q}'$ cross section arises from the axial part of the Z^0. This term is not included in the program, but fortunately it is finite and small.

Whereas the way the string, i.e. the fragmenting colour flux tube, is stretched is uniquely given in $q\bar{q}g$ event, for $q\bar{q}gg$ events there are two possibilities: $q-g_1-g_2-\bar{q}$ or $q-g_2-g_1-\bar{q}$. A knowledge of quark and gluon colours, obtained by perturbation theory, will uniquely specify the stretching of the string, as long as the two gluons do not have the same colour. The probability for the latter is down in magnitude by a factor $1/N_C^2=1/9$. One may either choose to neglect these terms entirely, or to keep them for the choice of kinematical setup, but then drop them at the choice of string drawing [Gus82]. We have adopted the latter procedure. Comparing the two possibilities, differences are typically 10–20% for a given kinematical configuration, and less for the total 4-jet cross section, so from a practical point of view this is not a major problem.

In higher orders, results depend on the renormalization scheme; we will use $\overline{\mathrm{MS}}$ throughout. In addition to this choice, several possible forms can be chosen for α_s, all of which are equivalent to that order but differ in higher orders. We have picked the recommended standard [PDG88]

$$\alpha_s(Q^2)=\frac{12\pi}{(33-2n_f)\,\ln(Q^2/\Lambda^2_{\overline{\mathrm{MS}}})}\left\{1-6\,\frac{153-19n_f}{(33-2n_f)^2}\,\frac{\ln(\ln(Q^2/\Lambda^2_{\overline{\mathrm{MS}}}))}{\ln(Q^2/\Lambda^2_{\overline{\mathrm{MS}}})}\right\}\,. \qquad (32)$$

6.1.4 Second-order 3-jet matrix elements

As for first order, a full second-order calculation consists both of real parton emission terms and of vertex and propagator corrections. These modify the 3-jet and 2-jet cross sections. Although there was some initial confusion, everybody soon agreed on the size of the loop corrections [Ell81, Ver81, Fab82]. In analytic calculations, the procedure of eq. (29), suitably expanded, can therefore be used unambiguously for a well-behaved variable.

For Monte Carlo event simulation, it is again necessary to impose some finite jet-resolution criterion. This means that four-parton events which fail the cuts should be reassigned either to the 3-jet or to the 2-jet event class. It is this area that caused quite a lot of confusion in the past [Kun81, Got82, Ali82, Zhu83, Gut84, Gut87, Kra88], and where full agreement does not exist. Most likely, agreement will never be reached, since there are indeed ambiguous points in the procedure, related to uncertainties on the theoretical side, as follows.

For the y-cut case, any two partons with an invariant mass $m_{ij}^2 < yE_{\mathrm{cm}}^2$ should be recombined into one. If the four-momenta are simply added, the sum will correspond to a parton with a positive mass, namely the original m_{ij}. The loop corrections are given in terms of final massless partons, however. In order to perform the (partial) cancellation between the four-parton real and the 3-parton virtual contributions, it is therefore necessary to get rid of the bothersome mass in the four-parton states. Several recombinations are used in practice, which go under names such as 'E', 'E0', 'p' and 'p0' [OPA91]. In the 'E'-type schemes, the energy of a recombined parton is given by $E_{ij} = E_i + E_j$, and three-momenta may have to be adjusted accordingly. In the 'p'-type schemes, on the other hand, three-momenta are added, $\mathbf{p}_{ij} = \mathbf{p}_i + \mathbf{p}_j$, and then energies may have to be adjusted. These procedures result in different 3-jet topologies, and therefore in different second-order differential 3-jet cross sections.

Within each scheme, a number of lesser points remain to be dealt with, in particular what to do if a recombination of a nearby parton pair were to give an event with a non-$\mathrm{q\bar{q}g}$ flavour structure.

This code contains two alternative second-order 3-jet implementations, GKS and ERT(Zhu). The latter is the recommended one and default. Other parameterizations have also been made available that run together with JETSET 6 (but not adopted to the current program), see [Sjö89, Mag89].

The GKS option is based on the GKS [Gut84] calculation, where some of the original mistakes in FKSS [Fab82] have been corrected. The GKS formulae have the advantage of giving the second-order corrections in closed analytic form, as not-too-long functions of x_1, x_2, and the y cut. However, it is today recognized, also by the authors, that important terms are still missing, and that the matrix elements should therefore not be taken too seriously. The option is thus kept mainly for backwards compatibility.

The ERT(Zhu) generator [Zhu83] is based on the ERT matrix elements [Ell81], with a Monte Carlo recombination procedure suggested by Kunszt [Kun81] and developed by Ali [Ali82]. It has the merit of giving corrections in a convenient, parameterized form. For practical applications, the main limitation is that the corrections are only given for discrete values of the cut-off parameter y, namely y = 0.01, 0.02, 0.03, 0.04, and 0.05. At these y values, the full second-order 3-jet cross section is written in terms of the 'ratio function' $R(X,Y;y)$, defined by

$$\frac{1}{\sigma_0}\frac{\mathrm{d}\sigma_3^{\mathrm{tot}}}{\mathrm{d}X\,\mathrm{d}Y} = \frac{\alpha_{\mathrm{s}}}{\pi}A_0(X,Y)\left\{1+\frac{\alpha_{\mathrm{s}}}{\pi}R(X,Y;y)\right\}\ , \tag{33}$$

where $X = x_1 - x_2 = x_{\mathrm{q}} - x_{\bar{\mathrm{q}}}$, $Y = x_3 = x_g$, σ_0 is the lowest-order hadronic cross section, and $A_0(X,Y)$ the standard first-order 3-jet cross section, cf. eq. (26). By Monte Carlo integration, the value of $R(X,Y;y)$ is evaluated in bins of (X,Y), and the result parameterized by a simple function $F(X,Y;y)$. Further details are found in [Sjö89].

6.1.5 The matrix-element event generator scheme

The program contains parameterizations, separately, of the total first-order 3-jet rate, the total second-order 3-jet rate, and the total 4-jet rate, all as functions of y (with α_s as a separate prefactor). These parameterizations have been obtained as follows:

- The first-order 3-jet matrix element is almost analytically integrable; some small finite pieces were obtained by a truncated series expansion of the relevant integrand.
- The GKS second-order 3-jet matrix elements were integrated for 40 different y-cut values, evenly distributed in $\ln y$ between a smallest value $y = 0.001$ and the kinematical limit $y = 1/3$. For each y value, 250 000 phase-space points were generated, evenly in $\mathrm{d}\ln(1-x_i) = \mathrm{d}x_i/(1-x_i)$, $i = 1,2$, and the second-order 3-jet rate in the point evaluated. The properly normalized sum of weights in each of the 40 y points were then fitted to a polynomial in $\ln(y^{-1}-2)$. For the ERT(Zhu) matrix elements the parameterizations in eq. (33) were used to perform a corresponding Monte Carlo integration for the five y values available.
- The 4-jet rate was integrated numerically, separately for $\mathrm{q\bar{q}gg}$ and $\mathrm{q\bar{q}q'\bar{q}'}$ events, by generating large samples of 4-jet phase-space points within the boundary $y = 0.001$. Each point was classified according to the actual minimum y between any two partons. The same events could then be used to update the summed weights for 40 different counters, corresponding to y values evenly distributed in $\ln y$ between $y = 0.001$ and the kinematical limit $y = 1/6$. In fact, since the weight sums for large y values only received contributions from few phase-space points, extra (smaller) subsamples of events were generated with larger y cuts. The summed weights, properly normalized, were then parameterized in terms of polynomials in $\ln(y^{-1}-5)$. Since it turned out to be difficult to obtain one single good fit over the whole range of y values, different parameterizations are used above and below $y = 0.018$. As originally given, the $\mathrm{q\bar{q}q'\bar{q}'}$ parameterization only took into account four $\mathrm{q'}$ flavours, i.e. secondary $\mathrm{b\bar{b}}$ pairs were not generated, but this has been corrected for LEP.

In the generation stage, each event is treated on its own, which means that the α_s and y values may be allowed to vary from event to event. The main steps are the following.

1. The y value to be used in the current event is determined. If possible, this is the value given by you, but additional constraints exist from the validity of the parameterizations ($y \geq 0.001$ for GKS, $0.01 \leq y \leq 0.05$ for ERT(Zhu)) and an extra (user-modifiable) requirement of a minimum absolute invariant mass between jets (which translates into varying y cuts due to the effects of initial-state QED radiation).
2. The α_s value is calculated.
3. For the y and α_s values given, the relative two/three/four-jet composition is determined. This is achieved by using the parameterized functions of y for 3- and 4-jet rates, multiplied by the relevant number of factors of α_s. In ERT(Zhu), where the second-order 3-jet rate is available only at a few y values, intermediate results are obtained by linear interpolation in the ratio of second-order to first-order 3-jet rates. The 3-jet and 4-jet rates are normalized to the analytically known second-order total event rate, i.e. divided by R_{QCD} of eq. (25). Finally, the 2-jet rate is obtained by conservation of total probability.
4. If the combination of y and α_s values is such that the total 3- plus 4-jet fraction is larger than unity, i.e. the remainder 2-jet fraction negative, the y-cut value is raised (for that event), and the process is started over at point 3.
5. The choice is made between generating a 2-, 3- or 4-jet event, according to the relative probabilities.
6. For the generation of 4-jets, it is first necessary to make a choice between $\mathrm{q\bar{q}gg}$ and $\mathrm{q\bar{q}q'\bar{q}'}$ events, according to the relative (parameterized) total cross sections. A phase-space point is then selected, and the differential cross section at this point is evaluated and compared with a parameterized maximum weight. If the phase-space point is rejected, a new one is selected, until an acceptable 4-jet event is found.

7. For 3-jets, a phase-space point is first chosen according to the first-order cross section. For this point, the weight

$$W(x_1,x_2;y) = 1 + \frac{\alpha_s}{\pi} R(x_1,x_2;y) \tag{34}$$

is evaluated. Here $R(x_1,x_2;y)$ is analytically given for GKS [Gut84], while it is approximated by the parameterization $F(X,Y;y)$ of eq. (33) for ERT(Zhu). Again, linear interpolation of $F(X,Y;y)$ has to be applied for intermediate y values. The weight W is compared with a maximum weight

$$W_{\max}(y) = 1 + \frac{\alpha_s}{\pi} R_{\max}(y) \ , \tag{35}$$

which has been numerically determined beforehand and suitably parameterized. If the phase-space point is rejected, a new point is generated, etc.
8. Massive matrix elements are not available for second-order QCD (but are in the first-order option). However, if a 3- or 4-jet event determined above falls outside the phase-space region allowed for massive quarks, the event is rejected and reassigned to be a 2-jet event. (The way the y_{ij} and y_{ijk} variables of 4-jet events should be interpreted for massive quarks is not even unique, so some latitude has been taken here to provide a reasonable continuity from 3-jet events.) This procedure is known not to give the expected full mass suppression, but is a reasonable first approximation.
9. Finally, if the event is classified as a 2-jet event, either because it was initially so assigned, or because it failed the massive phase-space cuts for 3- and 4-jets, the generation of 2-jets is trivial.

6.1.6 Optimized perturbation theory

Theoretically, it turns out that the second-order corrections to the 3-jet rate are large. It is therefore not unreasonable to expect large third-order corrections to the 4-jet rate. Indeed, the experimental 4-jet rate is much larger than second order predicts (when fragmentation effects have been included), if α_s is determined based on the 3-jet rate [Sjö84a, JAD88].

The only consistent way to resolve this issue is to go ahead and calculate the full next order. This is a tough task, however, so people have looked at possible shortcuts. For example, one can try to minimize the higher-order contributions by a suitable choice of the renormalization scale [Ste81] — 'optimized perturbation theory'. This is equivalent to a different choice for the Q^2 scale in α_s, a scale which is not unambiguous anyway. Indeed the standard value $Q^2 = s = E_{\rm cm}^2$ is larger than the natural physical scale of gluon emission in events, given that most gluons are fairly soft. One could therefore pick another scale, $Q^2 = fs$, with $f < 1$. The $O(\alpha_s)$ 3-jet rate would be increased by such a scale change, and so would the number of 4-jet events, including those which collapse into 3-jet ones. The loop corrections depend on the Q^2 scale, however, and compensate the changes above by giving a larger negative contribution to the 3-jet rate.

The possibility of picking an optimized scale f is implemented as follows [Sjö89]. Assume that the differential 3-jet rate at scale $Q^2 = s$ is given by the expression

$$R_3 = r_1\alpha_s + r_2\alpha_s^2 \ , \tag{36}$$

where R_3, r_1 and r_2 are functions of the kinematical variables x_1 and x_2 and the y cut, as implied by the second-order formulae above, see e.g. eq. (33). When the coupling is chosen at a different scale, $Q'^2 = fs$, the 3-jet rate has to be changed to

$$R_3' = r_1'\alpha_s' + r_2\alpha_s'^2 \ , \tag{37}$$

where $r'_1 = r_1$,

$$r'_2 = r_2 + r_1 \frac{33 - 2n_f}{12\pi} \ln f \ , \tag{38}$$

and $\alpha'_s = \alpha_s(fs)$. Since we only have the Born term for 4-jets, here the effects of a scale change come only from the change in the coupling constant. Finally, the 2-jet cross section can still be calculated from the difference between the total cross section and the 3- and 4-jet cross sections.

If an optimized scale is used in the program, the default value is $f = 0.002$, which is favoured by the studies in ref. [Bet89]. (In fact, it is also possible to use a correspondingly optimized R_{QCD} factor, eq. (25), but then the corresponding f is chosen independently and much closer to unity.) The success of describing the jet rates should not hide the fact that one is dabbling in (educated, hopefully) guesswork, and that any conclusions based on this method have to be taken with a pinch of salt.

One special problem associated with the use of optimized perturbation theory is that the differential 3-jet rate may become negative over large regions of the (x_1, x_2) phase space. This problem already exists, at least in principle, even for a scale $f = 1$, since r_2 is not guaranteed to be positive definite. Indeed, depending on the choice of y cut, α_s value and recombination scheme, one may observe a small region of negative differential 3-jet rate for the full second-order expression. This region is centred around $\mathrm{q\overline{q}g}$ configurations, where the q and $\overline{\mathrm{q}}$ are close together in one hemisphere and the g is alone in the other, i.e. $x_1 \approx x_2 \approx 1/2$. It is well understood why second-order corrections should be negative in this region [Dok89]: the q and $\overline{\mathrm{q}}$ of a $\mathrm{q\overline{q}g}$ state are in a relative colour octet state, and thus the colour force between them is repulsive, which translates into a negative second-order term.

However, as f is decreased below unity, r'_2 receives a negative contribution from the $\ln f$ term, and the region of negative differential cross section has a tendency to become larger, also after taking into account related changes in α_s. In an event-generator framework, where all events are supposed to come with unit weight, it is clearly not possible to simulate negative cross sections. What happens in the program is therefore that no 3-jet events at all are generated in the regions of negative differential cross section, and that the 3-jet rate in regions of positive cross sections is reduced by a constant factor, chosen so that the total number of 3-jet events comes out as it should. This is a consequence of the way the program works, where it is first decided what kind of event to generate, based on integrated 3-jet rates in which positive and negative contributions are added up with sign, and only thereafter the kinematics is chosen.

Based on our physics understanding of the origin of this negative cross section, the approach adopted is as sensible as any, at least to that order in perturbation theory (what one might strive for is a properly exponentiated description of the relevant region). It can give rise to funny results for low f values, however, as observed by OPAL [OPA92] for the energy–energy correlation asymmetry.

6.1.7 Angular orientation

While pure γ exchange gives a simple $1 + \cos^2\theta$ distribution for the q (and $\overline{\mathrm{q}}$) direction in $\mathrm{q\overline{q}}$ events, Z^0 exchange and γ^*/Z^0 interference results in a forward–backward asymmetry. If one introduces

$$\begin{aligned} h'_{\mathrm{f}}(s) &= 2e_{\mathrm{e}} \left\{ a_{\mathrm{e}}(1 - P_{\mathrm{L}}^+ P_{\mathrm{L}}^-) - v_{\mathrm{e}}(P_{\mathrm{L}}^- - P_{\mathrm{L}}^+) \right\} \Re\chi(s) e_{\mathrm{f}} a_{\mathrm{f}} \\ &\quad + \left\{ 2v_{\mathrm{e}} a_{\mathrm{e}}(1 - P_{\mathrm{L}}^+ P_{\mathrm{L}}^-) - (v_{\mathrm{e}}^2 + a_{\mathrm{e}}^2)(P_{\mathrm{L}}^- - P_{\mathrm{L}}^+) \right\} |\chi(s)|^2 v_{\mathrm{f}} a_{\mathrm{f}} \ , \end{aligned} \tag{39}$$

then the angular distribution of the quark is given by

$$\frac{\mathrm{d}\sigma}{\mathrm{d}(\cos\theta_{\mathrm{f}})} \propto h_{\mathrm{f}}(s)(1 + \cos^2\theta_{\mathrm{f}}) + 2h'_{\mathrm{f}}(s)\cos\theta_{\mathrm{f}} \ . \tag{40}$$

The angular orientation of a 3- or 4-jet event may be described in terms of three angles χ, θ and φ; for 2-jet events only θ and φ are necessary. From a standard orientation, with the q

along the $+z$ axis and the $\overline{q}$ in the xz plane with $p_x > 0$, an arbitrary orientation may be reached by the rotations $+\chi$ in azimuthal angle, $+\theta$ in polar angle, and $+\varphi$ in azimuthal angle, in that order. Differential cross sections, including QFD effects and arbitrary beam polarizations have been given for 2- and 3-jet events in refs. [Ols80, Sch80]. We use the formalism of ref. [Ols80], with translation from their terminology according to$\chi \to \pi - \chi$ and $\varphi^- \to -(\varphi + \pi/2)$. The resulting formulae are tedious, but straightforward to apply, once the internal jet configuration has been chosen. 4-jet events are approximated by 3-jet ones, by joining the two gluons of a $q\overline{q}gg$ event and the q' and $\overline{q}'$ of a $q\overline{q}q'\overline{q}'$ event into one effective jet. This means that some angular asymmetries are neglected [Ali80a], but that weak effects are automatically included. It is assumed that the second-order 3-jet events have the same angular orientation as the first-order ones, some studies on this issue may be found in [Kör85]. Further, the formulae normally refer to the massless case; only for the QED 2- and 3-jet cases are mass corrections available.

The main effect of the angular distribution of multijet events is to smear the lowest-order result, i.e. to reduce any anisotropies present in 2-jet systems. In the parton-shower option of the program, only the initial $q\overline{q}$ axis is determined. The subsequent shower evolution then *de facto* leads to a smearing of the jet axis, although not necessarily in full agreement with the expectations from multijet matrix-element treatments.

6.1.8 Initial-state radiation

Initial-state photon radiation has been included using the formalism of ref. [Ber82]. Here each event contains either no photon or one, i.e. it is a first-order non-exponentiated description. The main formula for the hard radiative photon cross section is

$$\frac{d\sigma}{dx_\gamma} = \frac{\alpha_{em}}{\pi}\left(\ln\frac{s}{m_e^2} - 1\right)\frac{1+(1-x_\gamma)^2}{x_\gamma}\sigma_0(\hat{s}) \ , \tag{41}$$

where x_γ is the photon energy fraction of the beam energy, $\hat{s} = (1-x_\gamma)s$ is the squared reduced hadronic c.m. energy, and σ_0 is the ordinary annihilation cross section at the reduced energy. In particular, the selection of jet flavours should be done according to expectations at the reduced energy. The cross section is divergent both for $x_\gamma \to 1$ and $x_\gamma \to 0$. The former is related to the fact that σ_0 has a $1/\hat{s}$ singularity (the real photon pole) for $\hat{s} \to 0$. An upper cut on x_γ can here be chosen to fit the experimental setup. The latter is a soft photon singularity, which is to be compensated in the no-radiation cross section. A requirement $x_\gamma > 0.01$ has therefore been chosen so that the hard-photon fraction is smaller than unity. In the total cross section, effects from photons with $x_\gamma < 0.01$ are taken into account, together with vertex and vacuum polarization corrections (hadronic vacuum polarizations using a simple parameterization of the more complicated formulae of ref. [Ber82]).

The hard photon spectrum can be integrated analytically, for the full γ^*/Z^0 structure including interference terms, provided that no new flavour thresholds are crossed and that the R_{QCD} term in the cross section can be approximated by a constant over the range of allowed $\hat{s}$ values. In fact, threshold effects can be taken into account by standard rejection techniques, at the price of not obtaining the exact cross section analytically, but only by an effective Monte Carlo integration taking place in parallel with the ordinary event generation. In addition to x_γ, the polar angle θ_γ and azimuthal angle φ_γ of the photons are also to be chosen. Further, for the orientation of the hadronic system, a choice has to be made whether the photon is to be considered as having been radiated from the e^+ or from the e^-.

Final-state photon radiation, as well as interference between initial- and final-state radiation, has been left out of this treatment. The formulae for $e^+e^- \to \mu^+\mu^-$ cannot be simply taken over for the case of outgoing quarks, since the quarks as such only live for a short while before turning into hadrons. Another simplification in our treatment is that effects of incoming polarized $e^\pm$ beams have been completely neglected, i.e. neither the effective shift in azimuthal distribution of photons nor the reduction in polarization is included. The polarization parameters of the program are to be thought of as the effective polarization surviving after initial-state radiation.

6.1.9 Alternative matrix elements

The program contains two sets of 'toy model' matrix elements, one for an Abelian vector gluon model and one for a scalar gluon model. Clearly both of these alternatives are already excluded by data, and are anyway not viable alternatives for a consistent theory of strong interactions. They are therefore included more as references to show how well the characteristic features of QCD can be measured experimentally.

Second-order matrix elements are available for the Abelian vector gluon model. These are easily obtained from the standard QCD matrix elements by a substitution of the Casimir group factors: $C_F = 4/3 \to 1$, $N_C = 3 \to 0$, and $T_R = n_\mathrm{f}/2 \to 3n_\mathrm{f}$. First-order matrix elements contain only C_F; therefore the standard first-order QCD results may be recovered by a rescaling of α_s by a factor $4/3$. In second order the change of N_C to 0 means that $\mathrm{g} \to \mathrm{gg}$ couplings are absent from the Abelian model, while the change of T_R corresponds to an enhancement of the $\mathrm{g} \to \mathrm{q}'\overline{\mathrm{q}}'$ coupling, i.e. to an enhancement of the $\mathrm{q}\overline{\mathrm{q}}\mathrm{q}'\overline{\mathrm{q}}'$ 4-jet event rate.

The second-order corrections to the 3-jet rate turn out to be strongly negative — if α_s is fitted to get about the right rate of 4-jet events, the predicted differential 3-jet rate is negative almost everywhere in the (x_1, x_2) plane. Whether this unphysical behaviour would be saved by higher orders is unclear. It has been pointed out that the rate can be made positive by a suitable choice of scale, since α_s runs in opposite directions in an Abelian model and in QCD [Bet89]. This may be seen directly from eq. (38), where the term $33 = 11N_C$ is absent in the Abelian model, and therefore the scale-dependent term changes sign. In the program, optimized scales have not been implemented for this toy model. Therefore the alternatives provided for you are either to generate only 4-jet events, or to neglect second-order corrections to the 3-jet rate, or to have the total 3-jet rate set vanishing (so that only 2- and 4-jet events are generated). Normally we would expect the former to be the one of most interest, since it is in angular (and flavour) distributions of 4-jet events that the structure of QCD can be tested. Also note that the 'correct' running of α_s is not included; you are expected to use the option where α_s is just given as a constant number.

The scalar gluon model is even more excluded than the Abelian vector one, since differences appear already in the 3-jet matrix element [Lae80]:

$$\frac{\mathrm{d}\sigma}{\mathrm{d}x_1\,\mathrm{d}x_2} \propto \frac{x_3^2}{(1-x_1)(1-x_2)} \tag{42}$$

when only γ exchange is included. The axial part of the Z^0 gives a slightly different shape; this is included in the program but does not make much difference. The angular orientation does include the full γ^*/Z^0 interference [Lae80], but the main interest is in the 3-jet topology as such [Ell79]. No higher-order corrections are included. It is recommended to use the option of a fixed α_s also here, since the correct running is not available.

6.2 Decays of Onia Resonances

Many different possibilities are open for the decay of heavy $J^{PC} = 1^{--}$ onia resonances. Of special interest are the decays into three gluons or two gluons plus a photon, since these offer unique possibilities to study a 'pure sample' of gluon jets. A routine for this purpose is included in the program. It was written at a time where the expectations were to find toponium at PETRA energies. Given the large value of the top mass, weak decays dominate, to the extent that the top quark decays weakly even before a bound toponium state is formed, and thus the routine will be of no use for top. The charm system, on the other hand, is far too low in mass for a jet language to be of any use. The only application is therefore likely to be for Υ, which unfortunately also is on the low side in mass.

The matrix element for $q\overline{q} \to ggg$ is (in lowest order) [Kol78]

$$\frac{1}{\sigma_{ggg}} \frac{d\sigma_{ggg}}{dx_1\, dx_2} = \frac{1}{\pi^2 - 9} \left\{ \left(\frac{1-x_1}{x_2 x_3} \right)^2 + \left(\frac{1-x_2}{x_1 x_3} \right)^2 + \left(\frac{1-x_3}{x_1 x_2} \right)^2 \right\} , \tag{43}$$

where, as before, $x_i = 2E_i/E_{cm}$ in the c.m. frame of the event. This is a well-defined expression, without the kind of singularities encountered in the $q\overline{q}g$ matrix elements. In principle, no cuts at all would be necessary, but for reasons of numerical simplicity we implement a y cut as for continuum jet production, with all events not fulfilling this cut considered as (effective) gg events. For ggg events, each gg invariant mass is required to be at least 2 GeV.

Another process is $q\overline{q} \to \gamma gg$, obtained by replacing a gluon in $q\overline{q} \to ggg$ by a photon. This process has the same normalized cross section as the one above, if e.g. x_1 is taken to refer to the photon. The relative rate is [Kol78]

$$\frac{\sigma_{\gamma gg}}{\sigma_{ggg}} = \frac{36}{5} \frac{e_q^2 \alpha_{em}}{\alpha_s(Q^2)} . \tag{44}$$

Here e_q is the charge of the heavy quark, and the scale in α_s has been chosen as the mass of the onium state. If the mass of the recoiling gg system is lower than some cut-off (by default 2 GeV), the event is rejected.

In the present implementation the angular orientation of the ggg and γgg events is given for the $e^+e^- \to \gamma^* \to$ onium case [Kol78] (optionally with beam polarization effects included), i.e. weak effects have not been included, since they are negligible at around 10 GeV.

It is possible to start a perturbative shower evolution from either of the two states above. However, for Υ the phase space for additional evolution is so constrained that not much is to be gained from that. We therefore do not recommend this possibility. The shower generation machinery, when starting up from a γgg configuration, is constructed such that the photon energy is not changed. This means that there is currently no possibility to use showers to bring the theoretical photon spectrum in better agreement with the experimental one.

In string fragmentation language, a ggg state corresponds to a closed string triangle with the three gluons at the corners. As the partons move apart from a common origin, the string triangle expands. Since the photon does not take part in the fragmentation, the γgg state corresponds to a double string running between the two gluons.

6.3 Routines and Common Block Variables

6.3.1 e^+e^- continuum event generation

The only routine a normal user will call to generate e^+e^- continuum events is `PYEEVT`. The other routines listed below, as well as `PYSHOW` (see section 10.4), are called by `PYEEVT`.

```
CALL PYEEVT(KFL,ECM)
```

Purpose: to generate a complete event $e^+e^- \to \gamma^*/Z^0 \to q\overline{q} \to$ parton shower $\to$ hadrons according to QFD and QCD cross sections. As an alternative to parton showers, second-order matrix elements are available for $q\overline{q} + q\overline{q}g + q\overline{q}gg + q\overline{q}q'\overline{q}'$ production.

`KFL` : flavour of events generated.
- `= 0` : mixture of all allowed flavours according to relevant probabilities.
- `= 1 - 8` : primary quarks are only of the specified flavour `KFL`.

`ECM` : total c.m. energy of system.

Remark: Each call generates one event, which is independent of preceding ones, with one exception, as follows. If radiative corrections are included, the shape of the hard

photon spectrum is recalculated only with each PYXTEE call, which normally is done only if KFL, ECM or MSTJ(102) is changed. A change of e.g. the Z^0 mass in mid-run has to be followed either by a user call to PYXTEE or by an internal call forced e.g. by putting MSTJ(116)=3.

SUBROUTINE PYXTEE(KFL,ECM,XTOT) : to calculate the total hadronic cross section, including quark thresholds, weak, beam polarization, and QCD effects and radiative corrections. In the process, variables necessary for the treatment of hard photon radiation are calculated and stored.

KFL, ECM : as for PYEEVT.

XTOT : the calculated total cross section in nb.

SUBROUTINE PYRADK(ECM,MK,PAK,THEK,PHIK,ALPK) : to describe initial-state hard γ radiation.

SUBROUTINE PYXKFL(KFL,ECM,ECMC,KFLC) : to generate the primary quark flavour in case this is not specified by you.

SUBROUTINE PYXJET(ECM,NJET,CUT) : to determine the number of jets (2, 3 or 4) to be generated within the kinematically allowed region (characterized by CUT $= y_{\rm cut}$) in the matrix-element approach; to be chosen such that all probabilities are between 0 and 1.

SUBROUTINE PYX3JT(NJET,CUT,KFL,ECM,X1,X2) : to generate the internal momentum variables of a 3-jet event, $q\bar{q}g$, according to first- or second-order QCD matrix elements.

SUBROUTINE PYX4JT(NJET,CUT,KFL,ECM,KFLN,X1,X2,X4,X12,X14) : to generate the internal momentum variables for a 4-jet event, $q\bar{q}gg$ or $q\bar{q}q'\bar{q}'$, according to second-order QCD matrix elements.

SUBROUTINE PYXDIF(NC,NJET,KFL,ECM,CHI,THE,PHI) : to describe the angular orientation of the jets. In first-order QCD the complete QED or QFD formulae are used; in second order 3-jets are assumed to have the same orientation as in first, and 4-jets are approximated by 3-jets.

6.3.2 A routine for onium decay

In PYONIA we have implemented the decays of heavy onia resonances into three gluons or two gluons plus a photon, which are the dominant non-background-like decays of Υ.

```
CALL PYONIA(KFL,ECM)
```

Purpose: to simulate the process $e^+e^- \to \gamma^* \to 1^{--}$ onium resonance $\to$ (ggg or ggγ) $\to$ shower $\to$ hadrons.

KFL : the flavour of the quark giving rise to the resonance.

= 0 : generate ggg events alone.

= 1 - 8 : generate ggg and ggγ events in mixture determined by the squared charge of flavour KFL, see eq. (44). Normally KFL= 5.

ECM : total c.m. energy of system.

6.3.3 Common block variables

The status codes and parameters relevant for the e^+e^- routines are found in the common block PYDAT1. This common block also contains more general status codes and parameters, described elsewhere.

```
COMMON/PYDAT1/MSTU(200),PARU(200),MSTJ(200),PARJ(200)
```

Purpose: to give access to a number of status codes and parameters regulating the performance of the e^+e^- event generation routines.

`MSTJ(101)` : (D=5) gives the type of QCD corrections used for continuum events.

- `= 0` : only $q\bar{q}$ events are generated.
- `= 1` : $q\bar{q} + q\bar{q}g$ events are generated according to first-order QCD.
- `= 2` : $q\bar{q} + q\bar{q}g + q\bar{q}gg + q\bar{q}q'\bar{q}'$ events are generated according to second-order QCD.
- `= 3` : $q\bar{q} + q\bar{q}g + q\bar{q}gg + q\bar{q}q'\bar{q}'$ events are generated, but without second-order corrections to the 3-jet rate.
- `= 5` : a parton shower is allowed to develop from an original $q\bar{q}$ pair, see `MSTJ(38) - MSTJ(50)` for details.
- `= -1` : only $q\bar{q}g$ events are generated (within same matrix-element cuts as for `=1`). Since the change in flavour composition from mass cuts or radiative corrections is not taken into account, this option is not intended for quantitative studies.
- `= -2` : only $q\bar{q}gg$ and $q\bar{q}q'\bar{q}'$ events are generated (as for `=2`). The same warning as for `=-1` applies.
- `= -3` : only $q\bar{q}gg$ events are generated (as for `=2`). The same warning as for `=-1` applies.
- `= -4` : only $q\bar{q}q'\bar{q}'$ events are generated (as for `=2`). The same warning as for `=-1` applies.

Note 1: `MSTJ(101)` is also used in `PYONIA`, with

- $\leq$ `4` : $ggg + \gamma gg$ events are generated according to lowest-order matrix elements.
- $\geq$ `5` : a parton shower is allowed to develop from the original ggg or $gg\gamma$ configuration, see `MSTJ(38) - MSTJ(50)` for details.

Note 2: The default values of fragmentation parameters have been chosen to work well with the default parton-shower approach above. If any of the other options are used, or if the parton shower is used in non-default mode, it is normally necessary to retune fragmentation parameters. As an example, we note that the second-order matrix-element approach (`MSTJ(101)=2`) at PETRA/PEP energies gives a better description when the a and b parameters of the symmetric fragmentation function are set to $a =$`PARJ(41)=1`, $b =$`PARJ(42)=0.7`, and the width of the transverse momentum distribution to $\sigma =$`PARJ(21)=0.40`. In principle, one also ought to change the joining parameter to `PARJ(33)=PARJ(35)=1.1` to preserve a flat rapidity plateau, but if this should be forgotten, it does not make too much difference. For applications at TRISTAN or LEP, one has to change the matrix-element approach parameters even more, to make up for additional soft gluon effects not covered in this approach.

`MSTJ(102)` : (D=2) inclusion of weak effects (Z^0 exchange) for flavour production, angular orientation, cross sections and initial-state photon radiation in continuum events.

- `= 1` : QED, i.e. no weak effects are included.
- `= 2` : QFD, i.e. including weak effects.
- `= 3` : as `=2`, but at initialization in `PYXTEE` the Z^0 width is calculated from $\sin^2\theta_W$, α_{em} and Z^0 and quark masses (including bottom and top threshold factors for `MSTJ(103)` odd), assuming three full generations, and the result is stored in `PARJ(124)`.

`MSTJ(103)` : (D=7) mass effects in continuum matrix elements, in the form `MSTJ(103)` $= M_1 + 2M_2 + 4M_3$, where $M_i = 0$ if no mass effects and $M_i = 1$ if mass effects should be included. Here;

- M_1 : threshold factor for new flavour production according to QFD result;
- M_2 : gluon emission probability (only applies for `|MSTJ(101)|`≤ 1, otherwise no

mass effects anyhow);

M_3 : angular orientation of event (only applies for |MSTJ(101)|$\leq$ 1 and MSTJ(102)=1, otherwise no mass effects anyhow).

MSTJ(104) : (D=5) number of allowed flavours, i.e. flavours that can be produced in a continuum event if the energy is enough. A change to 6 makes top production allowed above the threshold, etc. Note that in $q\overline{q}q'\overline{q}'$ events only the first five flavours are allowed in the secondary pair, produced by a gluon breakup.

MSTJ(105) : (D=1) fragmentation and decay in PYEEVT and PYONIA calls.

= 0 : no PYEXEC calls, i.e. only matrix-element and/or parton-shower treatment, and collapse of small jet systems into one or two particles (in PYPREP).

= 1 : PYEXEC calls are made to generate fragmentation and decay chain.

= -1 : no PYEXEC calls and no collapse of small jet systems into one or two particles (in PYPREP).

MSTJ(106) : (D=1) angular orientation in PYEEVT and PYONIA.

= 0 : standard orientation of events, i.e. q along $+z$ axis and $\overline{q}$ along $-z$ axis or in xz plane with $p_x > 0$ for continuum events, and $g_1g_2g_3$ or γg_2g_3 in xz plane with g_1 or γ along the $+z$ axis for onium events.

= 1 : random orientation according to matrix elements.

MSTJ(107) : (D=0) radiative corrections to continuum events.

= 0 : no radiative corrections.

= 1 : initial-state radiative corrections (including weak effects for MSTJ(102)= 2 or 3).

MSTJ(108) : (D=2) calculation of α_s for matrix-element alternatives. The MSTU(111) and PARU(112) values are automatically overwritten in PYEEVT or PYONIA calls accordingly.

= 0 : fixed α_s value as given in PARU(111).

= 1 : first-order formula is always used, with Λ_{QCD} given by PARJ(121).

= 2 : first- or second-order formula is used, depending on value of MSTJ(101), with Λ_{QCD} given by PARJ(121) or PARJ(122).

MSTJ(109) : (D=0) gives a possibility to switch from QCD matrix elements to some alternative toy models. Is not relevant for shower evolution, MSTJ(101)=5, where one can use MSTJ(49) instead.

= 0 : standard QCD scenario.

= 1 : a scalar gluon model. Since no second-order corrections are available in this scenario, one can only use this with MSTJ(101) = 1 or -1. Also note that the event-as-a-whole angular distribution is for photon exchange only (i.e. no weak effects), and that no higher-order corrections to the total cross section are included.

= 2 : an Abelian vector gluon theory, with the colour factors $C_F = 1$ ($= 4/3$ in QCD), $N_C = 0$ ($= 3$ in QCD) and $T_R = 3n_f$ ($= n_f/2$ in QCD). If one selects $\alpha_{\rm Abelian} = (4/3)\alpha_{\rm QCD}$, the 3-jet cross section will agree with the QCD one, and differences are to be found only in 4-jets. The MSTJ(109)=2 option has to be run with MSTJ(110)=1 and MSTJ(111)=0; if need be, the latter variables will be overwritten by the program.

Warning: second-order corrections give a large negative contribution to the 3-jet cross section, so large that the whole scenario is of doubtful use. In order to make the second-order options work at all, the 3-jet cross section is here by hand set exactly equal to zero for MSTJ(101)=2. It is here probably better to use the option MSTJ(101)=3, although this is not a consistent procedure either.

MSTJ(110) : (D=2) choice of second-order contributions to the 3-jet rate.

= 1 : the GKS second-order matrix elements.

= 2 : the Zhu parameterization of the ERT matrix elements, based on the program

of Kunszt and Ali, i.e. in historical sequence ERT/Kunszt/Ali/Zhu. The parameterization is available for $y = 0.01$, 0.02, 0.03, 0.04 and 0.05. Values outside this range are put at the nearest border, while those inside it are given by a linear interpolation between the two nearest points. Since this procedure is rather primitive, one should try to work at one of the values given above. Note that no Abelian QCD parameterization is available for this option.

`MSTJ(111)` : (D=0) use of optimized perturbation theory for second-order matrix elements (it can also be used for first-order matrix elements, but here it only corresponds to a trivial rescaling of the α_s argument).

- `= 0` : no optimization procedure; i.e. $Q^2 = E_{cm}^2$.
- `= 1` : an optimized Q^2 scale is chosen as $Q^2 = fE_{cm}^2$, where f =`PARJ(128)` for the total cross section R factor, while f =`PARJ(129)` for the 3- and 4-jet rates. This f value enters via the α_s, and also via a term proportional to $\alpha_s^2 \ln f$. Some constraints are imposed; thus the optimized '3-jet' contribution to R is assumed to be positive (for `PARJ(128)`), the total 3-jet rate is not allowed to be negative (for `PARJ(129)`), etc. However, there is no guarantee that the differential 3-jet cross section is not negative (and truncated to 0) somewhere (this can also happen with $f = 1$, but is then less frequent). The actually obtained f values are stored in `PARJ(168)` and `PARJ(169)`, respectively. If an optimized Q^2 scale is used, then the Λ_{QCD} (and α_s) should also be changed. With the value $f = 0.002$, it has been shown [Bet89] that a $\Lambda_{QCD} = 0.100$ GeV gives a reasonable agreement; the parameter to be changed is `PARJ(122)` for a second-order running α_s. Note that, since the optimized Q^2 scale is sometimes below the charm threshold, the effective number of flavours used in α_s may well be 4 only. If one feels that it is still appropriate to use 5 flavours (one choice might be as good as the other), it is necessary to put `MSTU(113)=5`.

`MSTJ(115)` : (D=1) documentation of continuum or onium events, in increasing order of completeness.

- `= 0` : only the parton shower, the fragmenting partons and the generated hadronic system are stored in the `PYJETS` common block.
- `= 1` : also a radiative photon is stored (for continuum events).
- `= 2` : also the original e^+e^- are stored (with `K(I,1)=21`).
- `= 3` : also the γ or γ^*/Z^0 exchanged for continuum events, the onium state for resonance events is stored (with `K(I,1)=21`).

`MSTJ(116)` : (D=1) initialization of total cross section and radiative photon spectrum in `PYEEVT` calls.

- `= 0` : never; cannot be used together with radiative corrections.
- `= 1` : calculated at first call and then whenever `KFL` or `MSTJ(102)` is changed or `ECM` is changed by more than `PARJ(139)`.
- `= 2` : calculated at each call.
- `= 3` : everything is re-initialized in the next call, but `MSTJ(116)` is afterwards automatically put `=1` for use in subsequent calls.

`MSTJ(119)` : (I) check on need to re-initialize `PYXTEE`.

`MSTJ(120)` : (R) type of continuum event generated with the matrix-element option (with the shower one, the result is always `=1`).

- `= 1` : $q\bar{q}$.
- `= 2` : $q\bar{q}g$.
- `= 3` : $q\bar{q}gg$ from Abelian (QED-like) graphs in matrix element.
- `= 4` : $q\bar{q}gg$ from non-Abelian (i.e. containing triple-gluon coupling) graphs in matrix element.
- `= 5` : $q\bar{q}q'\bar{q}'$.

`MSTJ(121)` : (R) flag set if a negative differential cross section was encountered in the latest `PYX3JT` call. Events are still generated, but maybe not quite according to the distri-

bution one would like (the rate is set to zero in the regions of negative cross section, and the differential rate in the regions of positive cross section is rescaled to give the 'correct' total 3-jet rate).

`PARJ(121)` : (D=1.0 GeV) Λ value used in first-order calculation of α_s in the matrix-element alternative.

`PARJ(122)` : (D=0.25 GeV) Λ values used in second-order calculation of α_s in the matrix-element alternative.

`PARJ(123)` : (D=91.187 GeV) mass of Z^0 as used in propagators for the QFD case.

`PARJ(124)` : (D=2.489 GeV) width of Z^0 as used in propagators for the QFD case. Overwritten at initialization if `MSTJ(102)=3`.

`PARJ(125)` : (D=0.01) $y_{\rm cut}$, minimum squared scaled invariant mass of any two partons in 3- or 4-jet events; the main user-controlled matrix-element cut. `PARJ(126)` provides an additional constraint. For each new event, it is additionally checked that the total 3- plus 4-jet fraction does not exceed unity; if so the effective y cut will be dynamically increased. The actual y-cut value is stored in `PARJ(150)`, event by event.

`PARJ(126)` : (D=2. GeV) minimum invariant mass of any two partons in 3- or 4-jet events; a cut in addition to the one above, mainly for the case of a radiative photon lowering the hadronic c.m. energy significantly.

`PARJ(127)` : (D=1. GeV) is used as a safety margin for small colour-singlet jet systems, cf. `PARJ(32)`, specifically $q\overline{q}'$ masses in $q\overline{q}q'\overline{q}'$ 4-jet events and gg mass in onium γgg events.

`PARJ(128)` : (D=0.25) optimized Q^2 scale for the QCD R (total rate) factor for the `MSTJ(111)=1` option is given by $Q^2 = fE_{\rm cm}^2$, where $f =$`PARJ(128)`. For various reasons the actually used f value may be increased compared with the nominal one; while `PARJ(128)` gives the nominal value, `PARJ(168)` gives the actual one for the current event.

`PARJ(129)` : (D=0.002) optimized Q^2 scale for the 3- and 4-jet rate for the `MSTJ(111)=1` option is given by $Q^2 = fE_{\rm cm}^2$, where $f =$`PARJ(129)`. For various reasons the actually used f value may be increased compared with the nominal one; while `PARJ(129)` gives the nominal value, `PARJ(169)` gives the actual one for the current event. The default value is in agreement with the studies of Bethke [Bet89].

`PARJ(131)`, `PARJ(132)` : (D=2*0.) longitudinal polarizations $P_{\rm L}^+$ and $P_{\rm L}^-$ of incoming e^+ and e^-.

`PARJ(133)` : (D=0.) transverse polarization $P_{\rm T} = \sqrt{P_{\rm T}^+ P_{\rm T}^-}$, with $P_{\rm T}^+$ and $P_{\rm T}^-$ transverse polarizations of incoming e^+ and e^-.

`PARJ(134)` : (D=0.) mean of transverse polarization directions of incoming e^+ and e^-, $\Delta\varphi = (\varphi^+ + \varphi^-)/2$, with φ the azimuthal angle of polarization, leading to a shift in the φ distribution of jets by $\Delta\varphi$.

`PARJ(135)` : (D=0.01) minimum photon energy fraction (of beam energy) in initial-state radiation; should normally never be changed (if lowered too much, the fraction of events containing a radiative photon will exceed unity, leading to problems).

`PARJ(136)` : (D=0.99) maximum photon energy fraction (of beam energy) in initial-state radiation; may be changed to reflect actual trigger conditions of a detector (but must always be larger than `PARJ(135)`).

`PARJ(139)` : (D=0.2 GeV) maximum deviation of $E_{\rm cm}$ from the corresponding value at last `PYXTEE` call, above which a new call is made if `MSTJ(116)=1`.

`PARJ(141)` : (R) value of R, the ratio of continuum cross section to the lowest-order muon pair production cross section, as given in massless QED (i.e. three times the sum of active quark squared charges, possibly modified for polarization).

`PARJ(142)` : (R) value of R including quark-mass effects (for `MSTJ(102)=1`) and/or weak propagator effects (for `MSTJ(102)=2`).

`PARJ(143)` : (R) value of R as `PARJ(142)`, but including QCD corrections as given by `MSTJ(101)`.

`PARJ(144)` : (R) value of R as `PARJ(143)`, but additionally including corrections from initial-state photon radiation (if `MSTJ(107)=1`). Since the effects of heavy flavour thresholds are not simply integrable, the initial value of `PARJ(144)` is updated during the course of the run to improve accuracy.

`PARJ(145) - PARJ(148)` : (R) absolute cross sections in nb as for the cases `PARJ(141) - PARJ(144)` above.

`PARJ(150)` : (R) current effective matrix element cut-off $y_{\rm cut}$, as given by `PARJ(125)`, `PARJ(126)` and the requirements of having non-negative cross sections for 2-, 3- and 4-jet events. Not used in parton showers.

`PARJ(151)` : (R) value of c.m. energy `ECM` at last `PYXTEE` call.

`PARJ(152)` : (R) current first-order contribution to the 3-jet fraction; modified by mass effects. Not used in parton showers.

`PARJ(153)` : (R) current second-order contribution to the 3-jet fraction; modified by mass effects. Not used in parton showers.

`PARJ(154)` : (R) current second-order contribution to the 4-jet fraction; modified by mass effects. Not used in parton showers.

`PARJ(155)` : (R) current fraction of 4-jet rate attributable to $\mathrm{q\overline{q}q'\overline{q}'}$ events rather than $\mathrm{q\overline{q}gg}$ ones; modified by mass effects. Not used in parton showers.

`PARJ(156)` : (R) has two functions when using second-order QCD. For a 3-jet event, it gives the ratio of the second-order to the total 3-jet cross section in the given kinematical point. For a 4-jet event, it gives the ratio of the modified 4-jet cross section, obtained when neglecting interference terms whose colour flow is not well defined, to the full unmodified one, all evaluated in the given kinematical point. Not used in parton showers.

`PARJ(157) - PARJ(159)` : (I) used for cross-section calculations to include mass threshold effects to radiative photon cross section. What is stored is basic cross section, number of events generated and number that passed cuts.

`PARJ(160)` : (R) nominal fraction of events that should contain a radiative photon.

`PARJ(161) - PARJ(164)` : (I) give shape of radiative photon spectrum including weak effects.

`PARJ(168)` : (R) actual f value of current event in optimized perturbation theory for R; see `MSTJ(111)` and `PARJ(128)`.

`PARJ(169)` : (R) actual f value of current event in optimized perturbation theory for 3- and 4-jet rate; see `MSTJ(111)` and `PARJ(129)`.

`PARJ(171)` : (R) fraction of cross section corresponding to the axial coupling of quark pair to the intermediate γ^*/Z^0 state; needed for the Abelian gluon model 3-jet matrix element.

6.4 Examples

An ordinary $\mathrm{e^+e^-}$ annihilation event in the continuum, at a c.m. energy of 91 GeV, may be generated with

```
CALL PYEEVT(0,91D0)
```

In this case a $\mathrm{q\overline{q}}$ event is generated, including weak effects, followed by parton-shower evolution and fragmentation/decay treatment. Before a call to `PYEEVT`, however, a number of default values may be changed, e.g. `MSTJ(101)=2` to use second-order QCD matrix elements, giving a mixture of $\mathrm{q\overline{q}}$, $\mathrm{q\overline{q}g}$, $\mathrm{q\overline{q}gg}$, and $\mathrm{q\overline{q}q'\overline{q}'}$ events, `MSTJ(102)=1` to have QED only, `MSTJ(104)=6` to allow $\mathrm{t\overline{t}}$ production as well, `MSTJ(107)=1` to include initial-state photon radiation (including a treatment of the Z^0 pole), `PARJ(123)=92.0` to change the Z^0 mass, `PARJ(81)=0.3` to change

the parton-shower Λ value, or `PARJ(82)=1.5` to change the parton-shower cut-off. If initial-state photon radiation is used, some restrictions apply to how one can alternate the generation of events at different energies or with different Z^0 mass, etc. These restrictions are not there for efficiency reasons (the extra time for recalculating the extra constants every time is small), but because it ties in with the cross-section calculations (see `PARJ(144)`).

Most parameters can be changed independently of each other. However, if just one or a few parameters/switches are changed, one should not be surprised to find a rather bad agreement with the data, like e.g. a too low or high average hadron multiplicity. It is therefore usually necessary to retune one parameter related to the perturbative QCD description, like α_s or Λ, one of the two parameters *a* and *b* of the Lund symmetric fragmentation function (since they are so strongly correlated, it is often not necessary to retune both of them), and the average fragmentation transverse momentum — see Note 2 of the `MSTJ(101)` description for an example. For very detailed studies it may be necessary to retune even more parameters.

The three-gluon and gluon–gluon–photon decays of Υ may be simulated by a call

```
CALL PYONIA(5,9.46D0)
```

A typical program for analysis of e^+e^- annihilation events at 200 GeV might look something like

```
      IMPLICIT DOUBLE PRECISION(A-H, O-Z)
      IMPLICIT INTEGER(I-N)
      INTEGER PYK,PYCHGE,PYCOMP
      COMMON/PYJETS/N,NPAD,K(4000,5),P(4000,5),V(4000,5)
      COMMON/PYDAT1/MSTU(200),PARU(200),MSTJ(200),PARJ(200)
      COMMON/PYDAT2/KCHG(500,4),PMAS(500,4),PARF(2000),VCKM(4,4)
      COMMON/PYDAT3/MDCY(500,3),MDME(8000,2),BRAT(8000),KFDP(8000,5)
      MDCY(PYCOMP(111),1)=0           ! put pi0 stable
      MSTJ(107)=1                     ! include initial-state radiation
      PARU(41)=1D0                    ! use linear sphericity
      .....                           ! other desired changes
      CALL PYTABU(10)                 ! initialize analysis statistics
      DO 100 IEV=1,1000               ! loop over events
        CALL PYEEVT(0,200D0)          ! generate new event
        IF(IEV.EQ.1) CALL PYLIST(2)   ! list first event
        CALL PYTABU(11)               ! save particle composition
                                      !   statistics
        CALL PYEDIT(2)                ! remove decayed particles
        CALL PYSPHE(SPH,APL)          ! linear sphericity analysis
        IF(SPH.LT.0D0) GOTO 100       ! too few particles in event for
                                      !   PYSPHE to work on it (unusual)
        CALL PYEDIT(31)               ! orient event along axes above
        IF(IEV.EQ.1) CALL PYLIST(2)   ! list first treated event
        .....                         ! fill analysis statistics
        CALL PYTHRU(THR,OBL)          ! now do thrust analysis
        .....                         ! more analysis statistics
  100 CONTINUE                        !
      CALL PYTABU(12)                 ! print particle composition
                                      !   statistics
      .....                           ! print analysis statistics
      END
```

7 Process Generation

Much can be said about the processes in PYTHIA and the way they are generated. Therefore the material has been split into three sections. In the current one the philosophy underlying the event generation scheme is presented. Here we provide a generic description, where some special cases are swept under the carpet. In the next section, the existing processes are enumerated, with some comments about applications and limitations. Finally, in the third section the generation routines and common block switches are described.

The section starts with a survey of parton distributions, followed by a detailed description of the simple $2 \to 2$ and $2 \to 1$ hard subprocess generation schemes, including pairs of resonances. This is followed by a few comments on more complicated configurations, and on nonperturbative processes.

7.1 Parton Distributions

The parton distribution function $f_i^a(x, Q^2)$ parameterizes the probability to find a parton i with a fraction x of the beam energy when the beam particle a is probed by a hard scattering at virtuality scale Q^2. Usually the momentum-weighted combination $x f_i^a(x, Q^2)$ is used, for which the normalization condition $\sum_i \int_0^1 dx\, x f_i^a(x, Q^2) \equiv 1$ normally applies. The Q^2 dependence of parton distributions is perturbatively calculable, see section 10.3.1.

The parton distributions in PYTHIA come in many shapes, as shown in the following.

7.1.1 Baryons

For protons, many sets exist on the market. These are obtained by fits to experimental data, constrained so that the Q^2 dependence is in accordance with the standard QCD evolution equations. The current default in PYTHIA is GRV 94L [Glü95], a simple leading-order fit. Several other sets are found in PYTHIA. The complete list is:

- EHLQ sets 1 and 2 [Eic84];
- DO sets 1 and 2 [Duk82];
- the GRV 92L (updated version) fit [Glü92];
- the CTEQ 3L, CTEQ 3M and CTEQ 3D fits [Lai95];
- the GRV 94L, GRV 94M and GRV 94D fits [Glü95]; and
- the CTEQ 5L and CTEQ 5M1 fits [Lai00].

Of these, EHLQ, DO, GRV 92L, CTEQ 3L, GRV94L and CTEQ5L are leading-order parton distributions, while CTEQ 3D and GRV94D are in the next-to-leading-order DIS scheme and the rest in the next-to-leading order $\overline{\mathrm{MS}}$ scheme. The EHLQ and DO sets are by now rather old, and are kept mainly for backwards compatibility. Since only Born-level matrix elements are included in the program, there is no particular reason to use higher-order parton distributions — the resulting combination is anyway only good to leading-order accuracy. (Some higher-order corrections are effectively included by the parton-shower treatment, but there is no exact match.)

There is a steady flow of new parton-distribution sets on the market. To keep track of all of them is a major work on its own. Therefore PYTHIA contains an interface to an external library of parton distribution functions, PDFLIB [Plo93]. This is a truly encyclopedic collection of almost all proton, pion and photon parton distributions proposed since the late 70's. Three dummy routines come with the PYTHIA package, so as to avoid problems with unresolved external references if PDFLIB is not linked. One should also note that PYTHIA does not check the results, but assumes that sensible answers will be returned, also outside the nominal (x, Q^2) range of a set. Only the sets that come with PYTHIA have been suitably modified to provide reasonable answers outside their nominal domain of validity.

From the proton parton distributions, those of the neutron are obtained by isospin conjugation, i.e. $f_{\mathrm{u}}^{\mathrm{n}} = f_{\mathrm{d}}^{\mathrm{p}}$ and $f_{\mathrm{d}}^{\mathrm{n}} = f_{\mathrm{u}}^{\mathrm{p}}$.

The program does allow for incoming beams of a number of hyperons: Λ^0, $\Sigma^{-,0,+}$, $\Xi^{-,0}$ and Ω^-. Here one has essentially no experimental information. One could imagine to construct models in which valence s quarks are found at larger average x values than valence u and d ones, because of the larger s-quark mass. However, hyperon beams is a little-used part of the program, included only for a few specific studies. Therefore a simple approach has been taken, in which an average valence quark distribution is constructed as $f_{\mathrm{val}} = (f_{\mathrm{u,val}}^{\mathrm{p}} + f_{\mathrm{d,val}}^{\mathrm{p}})/3$, according to which each valence quark in a hyperon is assumed to be distributed. Sea-quark and gluon distributions are taken as in the proton. Any proton parton distribution set may be used with this procedure.

7.1.2 Mesons and photons

Data on meson parton distributions are scarce, so only very few sets have been constructed, and only for the $\pi^{\pm}$. PYTHIA contains the Owens set 1 and 2 parton distributions [Owe84], which for a long time were essentially the only sets on the market, and the more recent dynamically generated GRV LO (updated version) [Glü92a]. The latter one is the default in PYTHIA. Further sets are found in PDFLIB and can therefore be used by PYTHIA, just as described above for protons.

Like the proton was used as a template for simple hyperon sets, so also the pion is used to derive a crude ansatz for $\mathrm{K}^{\pm}/\mathrm{K}_{\mathrm{S}}^0/\mathrm{K}_{\mathrm{L}}^0$. The procedure is the same, except that now $f_{\mathrm{val}} = (f_{\mathrm{u,val}}^{\pi^+} + f_{\bar{\mathrm{d}},\mathrm{val}}^{\pi^+})/2$.

Sets of photon parton distributions have been obtained as for hadrons; an additional complication comes from the necessity to handle the matching of the vector meson dominance (VMD) and the perturbative pieces in a consistent manner. New sets have been produced where this division is explicit and therefore especially well suited for applications to event generation[Sch95]. The Schuler and Sjöstand set 1D is the default. Although the vector-meson philosophy is at the base, the details of the fits do not rely on pion data, but only on F_2^{γ} data. Here follows a brief summary of relevant details.

Real photons obey a set of inhomogeneous evolution equations, where the inhomogeneous term is induced by $\gamma \to \mathrm{q}\bar{\mathrm{q}}$ branchings. The solution can be written as the sum of two terms,

$$f_a^{\gamma}(x, Q^2) = f_a^{\gamma,\mathrm{NP}}(x, Q^2; Q_0^2) + f_a^{\gamma,\mathrm{PT}}(x, Q^2; Q_0^2) \ , \tag{45}$$

where the former term is a solution to the homogeneous evolution with a (nonperturbative) input at $Q = Q_0$ and the latter is a solution to the full inhomogeneous equation with boundary condition $f_a^{\gamma,\mathrm{PT}}(x, Q_0^2; Q_0^2) \equiv 0$. One possible physics interpretation is to let $f_a^{\gamma,\mathrm{NP}}$ correspond to $\gamma \leftrightarrow V$ fluctuations, where $V = \rho^0, \omega, \phi, \ldots$ is a set of vector mesons, and let $f_a^{\gamma,\mathrm{PT}}$ correspond to perturbative ('anomalous') $\gamma \leftrightarrow \mathrm{q}\bar{\mathrm{q}}$ fluctuations. The discrete spectrum of vector mesons can be combined with the continuous (in virtuality k^2) spectrum of $\mathrm{q}\bar{\mathrm{q}}$ fluctuations, to give

$$f_a^{\gamma}(x, Q^2) = \sum_V \frac{4\pi\alpha_{\mathrm{em}}}{f_V^2} f_a^{\gamma,V}(x, Q^2) + \frac{\alpha_{\mathrm{em}}}{2\pi} \sum_{\mathrm{q}} 2e_{\mathrm{q}}^2 \int_{Q_0^2}^{Q^2} \frac{\mathrm{d}k^2}{k^2} f_a^{\gamma,\mathrm{q}\bar{\mathrm{q}}}(x, Q^2; k^2) \ , \tag{46}$$

where each component $f^{\gamma,V}$ and $f^{\gamma,\mathrm{q}\bar{\mathrm{q}}}$ obeys a unit momentum sum rule.

In sets 1 the Q_0 scale is picked at a low value, 0.6 GeV, where an identification of the nonperturbative component with a set of low-lying mesons appear natural, while sets 2 use a higher value, 2 GeV, where the validity of perturbation theory is better established. The data are not good enough to allow a precise determination of Λ_{QCD}. Therefore we use a fixed value $\Lambda^{(4)} = 200$ MeV, in agreement with conventional results for proton distributions. In the VMD component the ρ^0 and ω have been added coherently, so that $\mathrm{u}\bar{\mathrm{u}} : \mathrm{d}\bar{\mathrm{d}} = 4 : 1$ at Q_0.

Unlike the p, the γ has a direct component where the photon acts as an unresolved probe. In the definition of F_2^γ this adds a component C^γ, symbolically

$$F_2^\gamma(x,Q^2) = \sum_{\mathrm{q}} e_{\mathrm{q}}^2 \left[f_{\mathrm{q}}^\gamma + f_{\overline{\mathrm{q}}}^\gamma \right] \otimes C_{\mathrm{q}} + f_{\mathrm{g}}^\gamma \otimes C_{\mathrm{g}} + C^\gamma \,. \tag{47}$$

Since $C^\gamma \equiv 0$ in leading order, and since we stay with leading-order fits, it is permissible to neglect this complication. Numerically, however, it makes a non-negligible difference. We therefore make two kinds of fits, one DIS type with $C^\gamma = 0$ and one $\overline{\mathrm{MS}}$ type including the universal part of C^γ.

When jet production is studied for real incoming photons, the standard evolution approach is reasonable also for heavy flavours, i.e. predominantly the c, but with a lower cut-off $Q_0 \approx m_{\mathrm{c}}$ for $\gamma \to \mathrm{c}\overline{\mathrm{c}}$. Moving to Deeply Inelastic Scattering, $\mathrm{e}\gamma \to \mathrm{e}X$, there is an extra kinematical constraint: $W^2 = Q^2(1-x)/x > 4m_{\mathrm{c}}^2$. It is here better to use the 'Bethe-Heitler' cross section for $\gamma^*\gamma \to \mathrm{c}\overline{\mathrm{c}}$. Therefore each distribution appears in two variants. For applications to real γ's the parton distributions are calculated as the sum of a vector-meson part and an anomalous part including all five flavours. For applications to DIS, the sum runs over the same vector-meson part, an anomalous part and possibly a C^γ part for the three light flavours, and a Bethe-Heitler part for c and b.

In version 2 of the SaS distributions, which are the ones found here, the extension from real to virtual photons was improved, and further options made available [Sch96]. The resolved components of the photon are dampened by phenomenologically motivated virtuality-dependent dipole factors, while the direct ones are explicitly calculable. Thus eq. (46) generalizes to

$$\begin{aligned} f_a^{\gamma^*}(x,Q^2,P^2) &= \sum_V \frac{4\pi\alpha_{\mathrm{em}}}{f_V^2} \left(\frac{m_V^2}{m_V^2+P^2} \right)^2 f_a^{\gamma,V}(x,Q^2;\tilde{Q}_0^2) \\ &+ \frac{\alpha_{\mathrm{em}}}{2\pi} \sum_{\mathrm{q}} 2e_{\mathrm{q}}^2 \int_{Q_0^2}^{Q^2} \frac{\mathrm{d}k^2}{k^2} \left(\frac{k^2}{k^2+P^2} \right)^2 f_a^{\gamma,\mathrm{q}\overline{\mathrm{q}}}(x,Q^2;k^2) \,. \end{aligned} \tag{48}$$

In addition to the introduction of the dipole form factors, note that the lower input scale for the VMD states is here shifted from Q_0^2 to some $\tilde{Q}_0^2 \geq Q_0^2$. This is based on a study of the evolution equation [Bor93] that shows that the evolution effectively starts 'later' in Q^2 for a virtual photon. Equation (48) is one possible answer. It depends on both Q^2 and P^2 in a non-trivial way, however, so that results are only obtained by a time-consuming numerical integration rather than as a simple parametrization. Therefore several other alternatives are offered, that are in some sense equivalent, but can be given in simpler form.

In addition to the SaS sets, PYTHIA also contains the Drees–Grassie set of parton distributions [Dre85] and, as for the proton, there is an interface to the PDFLIB library [Plo93]. These calls are made with photon virtuality P^2 below the hard process scale Q^2. Further author-recommended constrains are implemented in the interface to the GRS set [Glü99] which, along with SaS, is among the few also to define parton distributions of virtual photons. However, these sets do not allow a subdivision of the photon parton distributions into one VMD part and one anomalous part. This subdivision is necessary a sophisticated modelling of γp and $\gamma\gamma$ events, see above and section 7.7.2. As an alternative, for the VMD part alone, the ρ^0 parton distribution can be found from the assumed equality

$$f_i^{\rho^0} = f_i^{\pi^0} = \frac{1}{2}\left(f_i^{\pi^+} + f_i^{\pi^-} \right) \,. \tag{49}$$

Thus any π^+ parton distribution set, from any library, can be turned into a VMD ρ^0 set. The ω parton distribution is assumed the same, while the ϕ and J/ψ ones are handled in the very crude approximation $f_{\mathrm{s,val}}^\phi = f_{\mathrm{u,val}}^{\pi^+}$ and $f_{\mathrm{sea}}^\phi = f_{\mathrm{sea}}^{\pi^+}$. The VMD part needs to be complemented

by an anomalous part to make up a full photon distribution. The latter is fully perturbatively calculable, given the lower cut-off scale Q_0. The SaS parameterization of the anomalous part is therefore used throughout for this purpose. The Q_0 scale can be set freely in the `PARP(15)` parameter.

The $f_i^{\gamma,\mathrm{anom}}$ distribution can be further decomposed, by the flavour and the $p_\perp$ of the original branching $\gamma \to \mathrm{q}\overline{\mathrm{q}}$. The flavour is distributed according to squared charge (plus flavour thresholds for heavy flavours) and the $p_\perp$ according to $\mathrm{d}p_\perp^2/p_\perp^2$ in the range $Q_0 < p_\perp < Q$. At the branching scale, the photon only consists of a $\mathrm{q}\overline{\mathrm{q}}$ pair, with x distribution $\propto x^2 + (1-x)^2$. A component $f_a^{\gamma,\mathrm{q}\overline{\mathrm{q}}}(x,Q^2;k^2)$, characterized by its $k \approx p_\perp$ and flavour, then is evolved homogeneously from $p_\perp$ to Q. For theoretical studies it is convenient to be able to access a specific component of this kind. Therefore also leading-order parameterizations of these decomposed distributions are available [Sch95].

7.1.3 Leptons

Contrary to the hadron case, there is no necessity to introduce the parton-distribution function concept for leptons. A lepton can be considered as a point-like particle, with initial-state radiation handled by higher-order matrix elements. However, the parton distribution function approach offers a slightly simplified but very economical description of initial-state radiation effects for any hard process, also those for which higher-order corrections are not yet calculated.

Parton distributions for electrons have been introduced in PYTHIA, and are used also for muons and taus, with a trivial substitution of masses. Alternatively, one is free to use a simple 'unresolved' e, $f_\mathrm{e}^\mathrm{e}(x,Q^2) = \delta(x-1)$, where the e retains the full original momentum.

Electron parton distributions are calculable entirely from first principles, but different levels of approximation may be used. The parton-distribution formulae in PYTHIA are based on a next-to-leading-order exponentiated description, see ref. [Kle89], p. 34. The approximate behaviour is

$$f_\mathrm{e}^\mathrm{e}(x,Q^2) \approx \frac{\beta}{2}(1-x)^{\beta/2-1} \; ;$$
$$\beta = \frac{2\alpha_\mathrm{em}}{\pi}\left(\ln\frac{Q^2}{m_\mathrm{e}^2} - 1\right) \; . \tag{50}$$

The form is divergent but integrable for $x \to 1$, i.e. the electron likes to keep most of the energy. To handle the numerical precision problems for x very close to unity, the parton distribution is set, by hand, to zero for $x > 1-10^{-10}$, and is rescaled upwards in the range $1-10^{-7} < x < 1-10^{-10}$, in such a way that the total area under the parton distribution is preserved:

$$\left(f_\mathrm{e}^\mathrm{e}(x,Q^2)\right)_\mathrm{mod} = \begin{cases} f_\mathrm{e}^\mathrm{e}(x,Q^2) & 0 \le x \le 1-10^{-7} \\ \dfrac{1000^{\beta/2}}{1000^{\beta/2}-1}\, f_\mathrm{e}^\mathrm{e}(x,Q^2) & 1-10^{-7} < x < 1-10^{-10} \\ 0 & x > 1-10^{-10} \; . \end{cases} \tag{51}$$

A separate issue is that electron beams may not be monochromatic, more so than for other particles because of the small electron mass. In storage rings the main mechanism is synchrotron radiation. For future high-luminosity linear colliders, the beam–beam interactions at the collision vertex may induce a quite significant energy loss — 'beamstrahlung'. Note that neither of these are associated with any off-shellness of the electrons, i.e. the resulting spectrum only depends on x and not Q^2. Examples of beamstrahlung spectra are provided by the CIRCE program [Ohl97], with a sample interface on the PYTHIA webpages.

The branchings $\mathrm{e} \to \mathrm{e}\gamma$, which are responsible for the softening of the f_e^e parton distribution, also gives rise to a flow of photons. In photon-induced hard processes, the f_γ^e parton distribution can be used to describe the equivalent flow of photons. In the next section, a complete

differential photon flux machinery is introduced. Here some simpler first-order expressions are introduced, for the flux integrated up to a hard interaction scale Q^2. There is some ambiguity in the choice of Q^2 range over which emissions should be included. The naïve (default) choice is

$$f_\gamma^{\mathrm{e}}(x,Q^2) = \frac{\alpha_{\mathrm{em}}}{2\pi}\,\frac{1+(1-x)^2}{x}\,\ln\left(\frac{Q^2}{m_{\mathrm{e}}^2}\right) \ . \tag{52}$$

Here it is assumed that only one scale enters the problem, namely that of the hard interaction, and that the scale of the branching $\mathrm{e} \to \mathrm{e}\gamma$ is bounded from above by the hard interaction scale. For a pure QCD or pure QED shower this is an appropriate procedure, cf. section 10.1.3, but in other cases it may not be optimal. In particular, for photoproduction the alternative that is probably most appropriate is [Ali88]:

$$f_\gamma^{\mathrm{e}}(x,Q^2) = \frac{\alpha_{\mathrm{em}}}{2\pi}\,\frac{1+(1-x)^2}{x}\,\ln\left(\frac{Q^2_{\mathrm{max}}(1-x)}{m_{\mathrm{e}}^2 x^2}\right) \ . \tag{53}$$

Here Q^2_{max} is a user-defined cut for the range of scattered electron kinematics that is counted as photoproduction. Note that we now deal with two different Q^2 scales, one related to the hard subprocess itself, which appears as the argument of the parton distribution, and the other related to the scattering of the electron, which is reflected in Q^2_{max}.

Also other sources of photons should be mentioned. One is the beamstahlung photons mentioned above, where again CIRCE provides sample parameterizations. Another, particularly interesting one, is laser backscattering, wherein an intense laser pulse is shot at an incoming high-energy electron bunch. By Compton backscattering this gives rise to a photon energy spectrum with a peak at a sigificant fraction of the original electron energy [Gin82]. Both of these sources produce real photons, which can be considered as photon beams of variable energy (see subsection 9.8), decoupled from the production process proper.

In resolved photoproduction or resolved $\gamma\gamma$ interactions, one has to include the parton distributions for quarks and gluons inside the photon inside the electron. This is best done with the machinery of the next section. However, as an older and simpler alternative, $f_{\mathrm{q,g}}^{\mathrm{e}}$ can be obtained by a numerical convolution according to

$$f_{\mathrm{q,g}}^{\mathrm{e}}(x,Q^2) = \int_x^1 \frac{dx_\gamma}{x_\gamma}\, f_\gamma^{\mathrm{e}}(x_\gamma,Q^2)\, f_{\mathrm{q,g}}^{\gamma}\left(\frac{x}{x_\gamma},Q^2\right) \ , \tag{54}$$

with f_γ^{e} as discussed above. The necessity for numerical convolution makes this parton distribution evaluation rather slow compared with the others; one should therefore only have it switched on for resolved photoproduction studies.

One can obtain the positron distribution inside an electron, which is also the electron sea parton distribution, by a convolution of the two branchings $\mathrm{e} \to \mathrm{e}\gamma$ and $\gamma \to \mathrm{e}^+\mathrm{e}^-$; the result is [Che75]

$$f_{\mathrm{e}^+}^{\mathrm{e}^-}(x,Q^2) = \frac{1}{2}\left\{\frac{\alpha_{\mathrm{em}}}{2\pi}\left(\ln\frac{Q^2}{m_{\mathrm{e}}^2}-1\right)\right\}^2 \frac{1}{x}\left(\frac{4}{3}-x^2-\frac{4}{3}x^3+2x(1+x)\ln x\right) \ . \tag{55}$$

Finally, the program also contains the distribution of a transverse W^- inside an electron

$$f_{\mathrm{W}}^{\mathrm{e}}(x,Q^2) = \frac{\alpha_{\mathrm{em}}}{2\pi}\,\frac{1}{4\sin^2\theta_W}\,\frac{1+(1-x)^2}{x}\,\ln\left(1+\frac{Q^2}{m_{\mathrm{W}}^2}\right) \ . \tag{56}$$

7.1.4 Equivalent photon flux in leptons

With the `'gamma/`*lepton*`'` option of a `PYINIT` call, an ep or e^+e^- event (or corresponding processes with muons) is factorized into the flux of virtual photons and the subsequent interactions of such photons. While real photons always are transverse, the virtual photons also allow a longitudinal component. This corresponds to cross sections

$$d\sigma(ep \to e\mathbf{X}) = \sum_{\xi=T,L} \int\int dy\, dQ^2\; f^{\xi}_{\gamma/e}(y,Q^2)\; d\sigma(\gamma^*_{\xi} p \to \mathbf{X}) \tag{57}$$

and

$$d\sigma(ee \to ee\mathbf{X}) = \sum_{\xi_1,\xi_2=T,L} \int\int\int dy_1\, dQ_1^2\, dy_2\, dQ_2^2\; f^{\xi_1}_{\gamma/e}(y_1,Q_1^2)\, f^{\xi_2}_{\gamma/e}(y_2,Q_2^2)\; d\sigma(\gamma^*_{\xi_1}\gamma^*_{\xi_2} \to \mathbf{X})\ . \tag{58}$$

For ep events, this factorized ansatz is perfectly general, so long as azimuthal distributions in the final state are not studied in detail. In e^+e^- events, it is not a good approximation when the virtualities Q_1^2 and Q_2^2 of both photons become of the order of the squared invariant mass W^2 of the colliding photons [Sch98]. In this region the cross section have terms that depend on the relative azimuthal angle of the scattered leptons, and the transverse and longitudinal polarizations are non-trivially mixed. However, these terms are of order $Q_1^2 Q_2^2/W^2$ and can be neglected whenever at least one of the photons has low virtuality compared to W^2.

When Q^2/W^2 is small, one can derive [Bon73, Bud75, Sch98]

$$f^{T}_{\gamma/l}(y,Q^2) = \frac{\alpha_{em}}{2\pi}\left(\frac{(1+(1-y)^2}{y}\frac{1}{Q^2} - \frac{2m_l^2 y}{Q^4}\right)\ , \tag{59}$$

$$f^{L}_{\gamma/l}(y,Q^2) = \frac{\alpha_{em}}{2\pi}\frac{2(1-y)}{y}\frac{1}{Q^2}\ , \tag{60}$$

where $l = e^{\pm},\ \mu^{\pm}$ or $\tau^{\pm}$. In $f^{T}_{\gamma/l}$ the second term, proportional to m_l^2/Q^4, is not leading log and is therefore often omitted. Clearly it is irrelevant at large Q^2, but around the lower cut-off Q^2_{min} it significantly dampens the small-y rise of the first term. (Note that Q^2_{min} is y-dependent, so properly the dampening is in a region of the (y,Q^2) plane.) Overall, under realistic conditions, it reduces event rates by 5–10% [Sch98, Fri93].

The y variable is defined as the light-cone fraction the photon takes of the incoming lepton momentum. For instance, for l^+l^- events,

$$y_i = \frac{q_i k_j}{k_i k_j}\ , \qquad j = 2(1) \text{ for } i = 1(2)\ , \tag{61}$$

where the k_i are the incoming lepton four-momenta and the q_i the four-momenta of the virtual photons.

Alternatively, the energy fraction the photon takes in the rest frame of the collision can be used,

$$x_i = \frac{q_i(k_1+k_2)}{k_i(k_1+k_2)}\ , \qquad i = 1,2\ . \tag{62}$$

The two are simply related,

$$y_i = x_i + \frac{Q_i^2}{s}\ , \tag{63}$$

with $s = (k_1+k_2)^2$. (Here and in the following formulae we have omitted the lepton and hadron mass terms when it is not of importance for the argumentation.) Since the Jacobian

$\mathrm{d}(y_i, Q_i^2)/\mathrm{d}(x_i, Q_i^2) = 1$, either variable would be an equally valid choice for covering the phase space. Small x_i values will be of less interest for us, since they lead to small W^2, so $y_i/x_i \approx 1$ except in the high-Q^2 tail, and often the two are used interchangeably. Unless special Q^2 cuts are imposed, cross sections obtained with $f_{\gamma/l}^{\mathrm{T,L}}(x, Q^2)\,\mathrm{d}x$ rather than $f_{\gamma/l}^{\mathrm{T,L}}(y, Q^2)\,\mathrm{d}y$ differ only at the per mil level. For comparisons with experimental cuts, it is sometimes relevant to know which of the two is being used in an analysis.

In the ep kinematics, the x and y definitions give that

$$W^2 = xs = ys - Q^2 \ . \tag{64}$$

The W^2 expression for l^+l^- is more complicated, especially because of the dependence on the relative azimuthal angle of the scattered leptons, $\varphi_{12} = \varphi_1 - \varphi_2$:

$$\begin{aligned} W^2 &= x_1 x_2 s + \frac{2Q_1^2 Q_2^2}{s} - 2\sqrt{1 - x_1 - \frac{Q_1^2}{s}}\sqrt{1 - x_2 - \frac{Q_2^2}{s}}\, Q_1 Q_2 \cos\varphi_{12} \\ &= y_1 y_2 s - Q_1^2 - Q_2^2 + \frac{Q_1^2 Q_2^2}{s} - 2\sqrt{1 - y_1}\sqrt{1 - y_2}\, Q_1 Q_2 \cos\varphi_{12} \ . \end{aligned} \tag{65}$$

The lepton scattering angle θ_i is related to Q_i^2 as

$$Q_i^2 = \frac{x_i^2}{1 - x_i} m_i^2 + (1 - x_i)\left(s - \frac{2}{(1 - x_i)^2} m_i^2 - 2m_j^2\right)\sin^2(\theta_i/2) \ , \tag{66}$$

with $m_i^2 = k_i^2 = k'^2_i$ and terms of $O(m^4)$ neglected. The kinematical limits thus are

$$(Q_i^2)_{\min} \approx \frac{x_i^2}{1 - x_i} m_i^2 \ , \tag{67}$$

$$(Q_i^2)_{\max} \approx (1 - x_i)s \ , \tag{68}$$

unless experimental conditions reduce the θ_i ranges.

In summary, we will allow the possibility of experimental cuts in the x_i, y_i, Q_i^2, θ_i and W^2 variables. Within the allowed region, the phase space is Monte Carlo sampled according to $\prod_i (\mathrm{d}Q_i^2/Q_i^2)\,(\mathrm{d}x_i/x_i)\,\mathrm{d}\varphi_i$, with the remaining flux factors combined with the cross section factors to give the event weight used for eventual acceptance or rejection. This cross section in its turn can contain the parton densities of a resolved virtual photon, thus offering an effective convolution that gives partons inside photons inside electrons.

7.2 Kinematics and Cross Section for a $2 \to 2$ Process

In this section we begin the description of kinematics selection and cross-section calculation. The example is for the case of a $2 \to 2$ process, with final-state masses assumed to be vanishing. Later on we will expand to finite fixed masses, and to resonances.

Consider two incoming beam particles in their c.m. frame, each with energy E_{beam}. The total squared c.m. energy is then $s = 4E_{\mathrm{beam}}^2$. The two partons that enter the hard interaction do not carry the total beam momentum, but only fractions x_1 and x_2, respectively, i.e. they have four-momenta

$$\begin{aligned} p_1 &= E_{\mathrm{beam}}(x_1; 0, 0, x_1) \ , \\ p_2 &= E_{\mathrm{beam}}(x_2; 0, 0, -x_2) \ . \end{aligned} \tag{69}$$

There is no reason to put the incoming partons on the mass shell, i.e. to have time-like incoming four-vectors, since partons inside a particle are always virtual and thus space-like. These space-like virtualities are introduced as part of the initial-state parton-shower description, see section 10.3.3, but do not affect the formalism of this section, wherefore massless incoming partons is a sensible ansatz. The one example where it would be appropriate to put a parton on the mass shell is for an incoming lepton beam, but even here the massless kinematics description is adequate as long as the c.m. energy is correctly calculated with masses.

The squared invariant mass of the two partons is defined as

$$\hat{s} = (p_1 + p_2)^2 = x_1 x_2 s \; . \tag{70}$$

Instead of x_1 and x_2, it is often customary to use τ and either y or x_F:

$$\tau = x_1 x_2 = \frac{\hat{s}}{s} \; ; \tag{71}$$

$$y = \frac{1}{2} \ln \frac{x_1}{x_2} \; ; \tag{72}$$

$$x_F = x_1 - x_2 \; . \tag{73}$$

In addition to x_1 and x_2, two additional variables are needed to describe the kinematics of a scattering $1+2 \to 3+4$. One corresponds to the azimuthal angle φ of the scattering plane around the beam axis. This angle is always isotropically distributed for unpolarized incoming beam particles, and so need not be considered further. The other variable can be picked as $\hat{\theta}$, the polar angle of parton 3 in the c.m. frame of the hard scattering. The conventional choice is to use the variable

$$\hat{t} = (p_1 - p_3)^2 = (p_2 - p_4)^2 = -\frac{\hat{s}}{2}(1 - \cos\hat{\theta}) \; , \tag{74}$$

with $\hat{\theta}$ defined as above. In the following, we will make use of both $\hat{t}$ and $\hat{\theta}$. It is also customary to define $\hat{u}$,

$$\hat{u} = (p_1 - p_4)^2 = (p_2 - p_3)^2 = -\frac{\hat{s}}{2}(1 + \cos\hat{\theta}) \; , \tag{75}$$

but $\hat{u}$ is not an independent variable since

$$\hat{s} + \hat{t} + \hat{u} = 0 \; . \tag{76}$$

If the two outgoing particles have masses m_3 and m_4, respectively, then the four-momenta in the c.m. frame of the hard interaction are given by

$$\hat{p}_{3,4} = \left(\frac{\hat{s} \pm (m_3^2 - m_4^2)}{2\sqrt{\hat{s}}}, \pm \frac{\sqrt{\hat{s}}}{2} \beta_{34} \sin\hat{\theta}, 0, \pm \frac{\sqrt{\hat{s}}}{2} \beta_{34} \cos\hat{\theta} \right) \; , \tag{77}$$

where

$$\beta_{34} = \sqrt{\left(1 - \frac{m_3^2}{\hat{s}} - \frac{m_4^2}{\hat{s}}\right)^2 - 4 \frac{m_3^2}{\hat{s}} \frac{m_4^2}{\hat{s}}} \; . \tag{78}$$

Then $\hat{t}$ and $\hat{u}$ are modified to

$$\hat{t}, \hat{u} = -\frac{1}{2} \left\{ (\hat{s} - m_3^2 - m_4^2) \mp \hat{s} \beta_{34} \cos\hat{\theta} \right\} \; , \tag{79}$$

with

$$\hat{s} + \hat{t} + \hat{u} = m_3^2 + m_4^2 \; . \tag{80}$$

The cross section for the process $1+2 \to 3+4$ may be written as

$$\begin{aligned} \sigma &= \int\int\int \mathrm{d}x_1\,\mathrm{d}x_2\,\mathrm{d}\hat{t}\; f_1(x_1,Q^2)\; f_2(x_2,Q^2)\,\frac{\mathrm{d}\hat{\sigma}}{\mathrm{d}\hat{t}} \\ &= \int\int\int \frac{\mathrm{d}\tau}{\tau}\,\mathrm{d}y\,\mathrm{d}\hat{t}\, x_1 f_1(x_1,Q^2)\, x_2 f_2(x_2,Q^2)\,\frac{\mathrm{d}\hat{\sigma}}{\mathrm{d}\hat{t}}\;. \end{aligned} \tag{81}$$

The choice of Q^2 scale is ambiguous, and several alternatives are available in the program. For massless outgoing particles the default is the squared transverse momentum

$$Q^2 = \hat{p}_\perp^2 = \frac{\hat{s}}{4}\sin^2\hat{\theta} = \frac{\hat{t}\hat{u}}{\hat{s}}\;, \tag{82}$$

which is modified to

$$Q^2 = \frac{1}{2}(m_{\perp 3}^2 + m_{\perp 4}^2) = \frac{1}{2}(m_3^2+m_4^2) + \hat{p}_\perp^2 = \frac{1}{2}(m_3^2+m_4^2) + \frac{\hat{t}\hat{u} - m_3^2 m_4^2}{\hat{s}} \tag{83}$$

when masses are introduced in the final state. The mass term is selected such that, for $m_3 = m_4 = m$, the expression reduces to the squared transverse mass, $Q^2 = \hat{m}_\perp^2 = m^2 + \hat{p}_\perp^2$. For cases with spacelike virtual incoming photons, of virtuality $Q_i^2 = -m_i^2 = |p_i^2|$, a further generalization to

$$Q^2 = \frac{1}{2}(Q_1^2 + Q_2^2 + m_3^2 + m_4^2) + \hat{p}_\perp^2 \tag{84}$$

is offered.

The $\mathrm{d}\hat{\sigma}/\mathrm{d}\hat{t}$ expresses the differential cross section for a scattering, as a function of the kinematical quantities $\hat{s}$, $\hat{t}$ and $\hat{u}$. It is in this function that the physics of a given process resides.

The performance of a machine is measured in terms of its luminosity $\mathcal{L}$, which is directly proportional to the number of particles in each bunch and to the bunch crossing frequency, and inversely proportional to the area of the bunches at the collision point. For a process with a σ as given by eq. (81), the differential event rate is given by $\sigma\mathcal{L}$, and the number of events collected over a given period of time

$$N = \sigma \int \mathcal{L}\,\mathrm{d}t\;. \tag{85}$$

The program does not calculate the number of events, but only the integrated cross sections.

7.3 Resonance Production

The simplest way to produce a resonance is by a $2 \to 1$ process. If the decay of the resonance is not considered, the cross-section formula does not depend on $\hat{t}$, but takes the form

$$\sigma = \int\int \frac{\mathrm{d}\tau}{\tau}\,\mathrm{d}y\, x_1 f_1(x_1,Q^2)\, x_2 f_2(x_2,Q^2)\,\hat{\sigma}(\hat{s})\;. \tag{86}$$

Here the physics is contained in the cross section $\hat{\sigma}(\hat{s})$. The Q^2 scale is usually taken to be $Q^2 = \hat{s}$.

In published formulae, cross sections are often given in the zero-width approximation, i.e. $\hat{\sigma}(\hat{s}) \propto \delta(\hat{s} - m_R^2)$, where m_R is the mass of the resonance. Introducing the scaled mass $\tau_R = m_R^2/s$, this corresponds to a delta function $\delta(\tau - \tau_R)$, which can be used to eliminate the integral over τ.

However, what we normally want to do is replace the δ function by the appropriate Breit–Wigner shape. For a resonance width Γ_R this is achieved by the replacement

$$\delta(\tau - \tau_R) \to \frac{s}{\pi}\,\frac{m_R\Gamma_R}{(s\tau - m_R^2)^2 + m_R^2\Gamma_R^2}\;. \tag{87}$$

In this formula the resonance width Γ_R is a constant.

An improved description of resonance shapes is obtained if the width is made $\hat{s}$-dependent (occasionally also referred to as mass-dependent width, since $\hat{s}$ is not always the resonance mass), see e.g. [Ber89]. To first approximation, this means that the expression $m_R\Gamma_R$ is to be replaced by $\hat{s}\Gamma_R/m_R$, both in the numerator and the denominator. An intermediate step is to perform this replacement only in the numerator. This is convenient when not only s-channel resonance production is simulated but also non-resonance t- or u-channel graphs are involved, since mass-dependent widths in the denominator here may give an imperfect cancellation of divergences. (More about this below.)

To be more precise, in the program the quantity $H_R(\hat{s})$ is introduced, and the Breit–Wigner is written as

$$\delta(\tau-\tau_R) \to \frac{s}{\pi}\,\frac{H_R(s\tau)}{(s\tau-m_R^2)^2+H_R^2(s\tau)}\ . \tag{88}$$

The H_R factor is evaluated as a sum over all possible final-state channels, $H_R=\sum_f H_R^{(f)}$. Each decay channel may have its own $\hat{s}$ dependence, as follows.

A decay to a fermion pair, $R\to \mathrm{f}\bar{\mathrm{f}}$, gives no contribution below threshold, i.e. for $\hat{s}<4m_\mathrm{f}^2$. Above threshold, $H_R^{(f)}$ is proportional to $\hat{s}$, multiplied by a threshold factor $\beta(3-\beta^2)/2$ for the vector part of a spin 1 resonance, by β^3 for the axial vector part, by β^3 for a scalar resonance and by β for a pseudoscalar one. Here $\beta=\sqrt{1-4m_\mathrm{f}^2/\hat{s}}$. For the decay into unequal masses, e.g. of the W^+, corresponding but more complicated expressions are used.

For decays into a quark pair, a first-order strong correction factor $1+\alpha_\mathrm{s}(\hat{s})/\pi$ is included in $H_R^{(f)}$. This is the correct choice for all spin 1 colourless resonances, but is here used for all resonances where no better knowledge is available. Currently the major exception is top decay, where the factor $1-2.5\,\alpha_\mathrm{s}(\hat{s})/\pi$ is used to approximate loop corrections [Jeż89]. The second-order corrections are often known, but then are specific to each resonance, and are not included. An option exists for the $\gamma/\mathrm{Z}^0/\mathrm{Z}'^0$ resonances, where threshold effects due to $\mathrm{q}\bar{\mathrm{q}}$ bound-state formation are taken into account in a smeared-out, average sense, see eq. (137).

For other decay channels, not into fermion pairs, the $\hat{s}$ dependence is typically more complicated. An example would be the decay $\mathrm{h}^0\to\mathrm{W}^+\mathrm{W}^-$, with a nontrivial threshold and a subtle energy dependence above that [Sey95a]. Since a Higgs with $m_\mathrm{h}<2m_\mathrm{W}$ could still decay in this channel, it is in fact necessary to perform a two-dimensional integral over the $W^\pm$ Breit–Wigner mass distributions to obtain the correct result (and this has to be done numerically, at least in part). Fortunately, a Higgs particle lighter than $2m_\mathrm{W}$ is sufficiently narrow that the integral only needs to be performed once and for all at initialization (whereas most other partial widths are recalculated whenever needed). Channels that proceed via loops, such as $\mathrm{h}\to\mathrm{gg}$, also display complicated threshold behaviours.

The coupling structure within the electroweak sector is usually (re)expressed in terms of gauge boson masses, α_em and $\sin^2\theta_W$, i.e. factors of G_F are replaced according to

$$\sqrt{2}G_\mathrm{F}=\frac{\pi\,\alpha_\mathrm{em}}{\sin^2\theta_W\,m_\mathrm{W}^2}\ . \tag{89}$$

Having done that, α_em is allowed to run [Kle89], and is evaluated at the $\hat{s}$ scale. Thereby the relevant electroweak loop correction factors are recovered at the $m_\mathrm{W}/m_\mathrm{Z}$ scale. However, the option exists to go the other way and eliminate α_em in favour of G_F. Currently $\sin^2\theta_W$ is not allowed to run.

For Higgs particles and technipions, fermion masses enter not only in the kinematics but also as couplings. The latter kind of quark masses (but not the former, at least not in the program) are running with the scale of the process, i.e. normally the resonance mass. The expression used

is [Car96]

$$m(Q^2) = m_0 \left(\frac{\ln(k^2 m_0^2/\Lambda^2)}{\ln(Q^2/\Lambda^2)} \right)^{12/(33-2n_\mathrm{f})} . \tag{90}$$

Here m_0 is the input mass at a reference scale km_0, defined in the $\overline{\mathrm{MS}}$ scheme. Typical choices are either $k = 1$ or $k = 2$; the latter would be relevant if the reference scale is chosen at the $\mathrm{Q\overline{Q}}$ threshold. Both Λ and n_f are as given in α_s.

In summary, we see that an $\hat{s}$ dependence may enter several different ways into the $H_R^{(f)}$ expressions from which the total H_R is built up.

When only decays to a specific final state f are considered, the H_R in the denominator remains the sum over all allowed decay channels, but the numerator only contains the $H_R^{(f)}$ term of the final state considered.

If the combined production and decay process $i \to R \to f$ is considered, the same $\hat{s}$ dependence is implicit in the coupling structure of $i \to R$ as one would have had in $R \to i$, i.e. to first approximation there is a symmetry between couplings of a resonance to the initial and to the final state. The cross section $\hat{\sigma}$ is therefore, in the program, written in the form

$$\hat{\sigma}_{i\to R\to f}(\hat{s}) \propto \frac{\pi}{\hat{s}} \frac{H_R^{(i)}(\hat{s})\, H_R^{(f)}(\hat{s})}{(\hat{s}-m_R^2)^2 + H_R^2(\hat{s})} . \tag{91}$$

As a simple example, the cross section for the process $\mathrm{e}^-\overline{\nu}_\mathrm{e} \to \mathrm{W}^- \to \mu^-\overline{\nu}_\mu$ can be written as

$$\hat{\sigma}(\hat{s}) = 12\, \frac{\pi}{\hat{s}} \frac{H_\mathrm{W}^{(i)}(\hat{s})\, H_\mathrm{W}^{(f)}(\hat{s})}{(\hat{s}-m_\mathrm{W}^2)^2 + H_\mathrm{W}^2(\hat{s})} , \tag{92}$$

where

$$H_\mathrm{W}^{(i)}(\hat{s}) = H_\mathrm{W}^{(f)}(\hat{s}) = \frac{\alpha_\mathrm{em}(\hat{s})}{24 \sin^2\theta_W}\, \hat{s} . \tag{93}$$

If the effects of several initial and/or final states are studied, it is straightforward to introduce an appropriate summation in the numerator.

The analogy between the $H_R^{(f)}$ and $H_R^{(i)}$ cannot be pushed too far, however. The two differ in several important aspects. Firstly, colour factors appear reversed: the decay $R \to \mathrm{q\overline{q}}$ contains a colour factor $N_C = 3$ enhancement, while $\mathrm{q\overline{q}} \to R$ is instead suppressed by a factor $1/N_C = 1/3$. Secondly, the $1 + \alpha_\mathrm{s}(\hat{s})/\pi$ first-order correction factor for the final state has to be replaced by a more complicated K factor for the initial state. This factor is not known usually, or it is known (to first non-trivial order) but too lengthy to be included in the program. Thirdly, incoming partons as a rule are space-like. All the threshold suppression factors of the final-state expressions are therefore irrelevant when production is considered. In sum, the analogy between $H_R^{(f)}$ and $H_R^{(i)}$ is mainly useful as a consistency cross-check, while the two usually are calculated separately. Exceptions include the rather messy loop structure involved in $\mathrm{gg} \to \mathrm{h}^0$ and $\mathrm{h}^0 \to \mathrm{gg}$, which is only coded once.

It is of some interest to consider the observable resonance shape when the effects of parton distributions are included. In a hadron collider, to first approximation, parton distributions tend to have a behaviour roughly like $f(x) \propto 1/x$ for small x — this is why $f(x)$ is replaced by $xf(x)$ in eq. (81). Instead, the basic parton-distribution behaviour is shifted into the factor of $1/\tau$ in the integration phase space $\mathrm{d}\tau/\tau$, cf. eq. (86). When convoluted with the Breit–Wigner shape, two effects appear. One is that the overall resonance is tilted: the low-mass tail is enhanced and the high-mass one suppressed. The other is that an extremely long tail develops on the low-mass side of the resonance: when $\tau \to 0$, eq. (91) with $H_R(\hat{s}) \propto \hat{s}$ gives a $\hat{\sigma}(\hat{s}) \propto \hat{s} \propto \tau$, which exactly cancels the $1/\tau$ factor mentioned above. Naïvely, the integral over y, $\int \mathrm{d}y = -\ln\tau$, therefore gives a net logarithmic divergence of the resonance shape when $\tau \to 0$. Clearly, it

is then necessary to consider the shape of the parton distributions in more detail. At not-too-small Q^2, the evolution equations in fact lead to parton distributions more strongly peaked than $1/x$, typically with $xf(x) \propto x^{-0.3}$, and therefore a divergence like $\tau^{-0.3}$ in the cross-section expression. Eventually this divergence is regularized by a closing of the phase space, i.e. that $H_R(\hat{s})$ vanishes faster than $\hat{s}$, and by a less drastic small-x parton-distribution behaviour when $Q^2 \approx \hat{s} \to 0$.

The secondary peak at small τ may give a rather high cross section, which can even rival that of the ordinary peak around the nominal mass. This is the case, for instance, with W production. Such a peak has never been observed experimentally, but this is not surprising, since the background from other processes is overwhelming at low $\hat{s}$. Thus a lepton of one or a few GeV of transverse momentum is far more likely to come from the decay of a charm or bottom hadron than from an extremely off-shell W of a mass of a few GeV. When resonance production is studied, it is therefore important to set limits on the mass of the resonance, so as to agree with the experimental definition, at least to first approximation. If not, cross-section information given by the program may be very confusing.

Another problem is that often the matrix elements really are valid only in the resonance region. The reason is that one usually includes only the simplest s-channel graph in the calculation. It is this 'signal' graph that has a peak at the position of the resonance, where it (usually) gives much larger cross sections than the other 'background' graphs. Away from the resonance position, 'signal' and 'background' may be of comparable order, or the 'background' may even dominate. There is a quantum mechanical interference when some of the 'signal' and 'background' graphs have the same initial and final state, and this interference may be destructive or constructive. When the interference is non-negligible, it is no longer meaningful to speak of a 'signal' cross section. As an example, consider the scattering of longitudinal W's, $\mathrm{W_L^+W_L^-} \to \mathrm{W_L^+W_L^-}$, where the 'signal' process is s-channel exchange of a Higgs. This graph by itself is ill-behaved away from the resonance region. Destructive interference with 'background' graphs such as t-channel exchange of a Higgs and s- and t-channel exchange of a γ/Z is required to save unitarity at large energies.

In $\mathrm{e^+e^-}$ colliders, the f_e^e parton distribution is peaked at $x = 1$ rather than at $x = 0$. The situation therefore is the opposite, if one considers e.g. Z^0 production in a machine running at energies above m_Z: the tail towards lower masses is suppressed and the one towards higher masses enhanced, with a sharp secondary peak at around the nominal energy of the machine. Also in this case, an appropriate definition of cross sections therefore is necessary — with additional complications due to the interference between γ^* and Z^0. When other processes are considered, problems of interference with background appears also here. Numerically the problems may be less pressing, however, since the secondary peak is occurring in a high-mass region, rather than in a more complicated low-mass one. Further, in $\mathrm{e^+e^-}$ there is little uncertainty from the shape of the parton distributions.

In $2 \to 2$ processes where a pair of resonances are produced, e.g. $\mathrm{e^+e^-} \to \mathrm{Z^0h^0}$, cross section are almost always given in the zero-width approximation for the resonances. Here two substitutions of the type

$$1 = \int \delta(m^2 - m_R^2)\, dm^2 \to \int \frac{1}{\pi} \frac{m_R \Gamma_R}{(m^2 - m_R^2)^2 + m_R^2 \Gamma_R^2}\, dm^2 \tag{94}$$

are used to introduce mass distributions for the two resonance masses, i.e. m_3^2 and m_4^2. In the formula, m_R is the nominal mass and m the actually selected one. The phase-space integral over x_1, x_1 and $\hat{t}$ in eq. (81) is then extended to involve also m_3^2 and m_4^2. The effects of the mass-dependent width is only partly taken into account, by replacing the nominal masses m_3^2 and m_4^2 in the $\mathrm{d}\hat{\sigma}/\mathrm{d}\hat{t}$ expression by the actually generated ones (also e.g. in the relation between $\hat{t}$ and $\cos\hat{\theta}$), while the widths are evaluated at the nominal masses. This is the equivalent of a simple replacement of $m_R\Gamma_R$ by $\hat{s}\Gamma_R/m_R$ in the numerator of eq. (87), but not in the denominator. In

addition, the full threshold dependence of the widths, i.e. the velocity-dependent factors, is not reproduced.

There is no particular reason why the full mass-dependence could not be introduced, except for the extra work and time consumption needed for each process. In fact, the matrix elements for several γ^*/Z^0 and $W^\pm$ production processes do contain the full expressions. On the other hand, the matrix elements given in the literature are often valid only when the resonances are almost on the mass shell, since some graphs have been omitted. As an example, the process $q\overline{q} \to e^-\overline{\nu}_e\mu^+\nu_\mu$ is dominated by $q\overline{q} \to W^-W^+$ when each of the two lepton pairs is close to m_W in mass, but in general also receives contributions e.g. from $q\overline{q} \to Z^0 \to e^+e^-$, followed by $e^+ \to \overline{\nu}_e W^+$ and $W^+ \to \mu^+\nu_\mu$. The latter contributions are neglected in cross sections given in the zero-width approximation.

Widths may induce gauge invariance problems, in particular when the s-channel graph interferes with t- or u-channels. Then there may be an imperfect cancellation of contributions at high energies, leading to an incorrect cross section behaviour. The underlying reason is that a Breit-Wigner corresponds to a resummation of terms of different orders in coupling constants, and that therefore effectively the s-channel contributions are calculated to higher orders than the t- or u-channel ones, including interference contributions. A specific example is $e^+e^- \to W^+W^-$, where s-channel γ^*/Z^* exchange interferes with t-channel ν_e exchange. In such cases, a fixed width is used in the denominator. One could also introduce procedures whereby the width is made to vanish completely at high energies, and theoretically this is the cleanest, but the fixed-width approach appears good enough in practice.

Another gauge invariance issue is when two particles of the same kind are produced in a pair, e.g. $gg \to t\overline{t}$. Matrix elements are then often calculated for one common m_t mass, even though in real life the masses $m_3 \neq m_4$. The proper gauge invariant procedure to handle this would be to study the full six-fermion state obtained after the two $t \to bW \to bf_i\overline{f}_j$ decays, but that may be overkill if indeed the t's are close to mass shell. Even when only equal-mass matrix elements are available, Breit-Wigners are therefore used to select two separate masses m_3 and m_4. From these two masses, an average mass $\overline{m}$ is constructed so that the β_{34} velocity factor of eq. (78) is retained,

$$\beta_{34}(\hat{s},\overline{m}^2,\overline{m}^2) = \beta_{34}(\hat{s},m_3^2,m_4^2) \quad \Rightarrow \quad \overline{m}^2 = \frac{m_3^2+m_4^2}{2} - \frac{(m_3^2-m_4^2)^2}{4\hat{s}}. \tag{95}$$

This choice certainly is not unique, but normally should provide a sensible behaviour, also around threshold. The approach may well break down when either or both particles are far away from mass shell. Furthermore, the preliminary choice of scattering angle $\hat{\theta}$ is also retained. Instead of the correct $\hat{t}$ and $\hat{u}$ of eq. (79), modified

$$\overline{\hat{t}},\overline{\hat{u}} = -\frac{1}{2}\left\{(\hat{s}-2\overline{m}^2) \mp \hat{s}\beta_{34}\cos\hat{\theta}\right\} = (\hat{t},\hat{u}) - \frac{(m_3^2-m_4^2)^2}{4\hat{s}} \tag{96}$$

can then be obtained. The $\overline{m}^2$, $\overline{\hat{t}}$ and $\overline{\hat{u}}$ are now used in the matrix elements to decide whether to retain the event or not.

Processes with one final-state resonance and another ordinary final-state product, e.g. $qg \to W^+q'$, are treated in the same spirit as the $2 \to 2$ processes with two resonances, except that only one mass need be selected according to a Breit–Wigner.

7.4 Cross-section Calculations

In the program, the variables used in the generation of a $2 \to 2$ process are τ, y and $z = \cos\hat{\theta}$. For a $2 \to 1$ process, the z variable can be integrated out, and need therefore not be generated as part of the hard process, except when the allowed angular range of decays is restricted. In unresolved lepton beams, i.e. when $f_e^e(x) = \delta(x-1)$, the variables τ and/or y may be integrated out. We will cover all these special cases towards the end of the section, and here concentrate on 'standard' $2 \to 2$ and $2 \to 1$ processes.

7.4.1 The simple $2 \to 2$ processes

In the spirit of section 4.1, we want to select simple functions such that the true τ, y and z dependence of the cross sections is approximately modelled. In particular, (almost) all conceivable kinematical peaks should be represented by separate terms in the approximate formulae. If this can be achieved, the ratio of the correct to the approximate cross sections will not fluctuate too much, but allow reasonable Monte Carlo efficiency.

Therefore the variables are generated according to the distributions $h_\tau(\tau)$, $h_y(y)$ and $h_z(z)$, where normally

$$\begin{aligned} h_\tau(\tau) &= \frac{c_1}{I_1}\frac{1}{\tau} + \frac{c_2}{I_2}\frac{1}{\tau^2} + \frac{c_3}{I_3}\frac{1}{\tau(\tau+\tau_R)} + \frac{c_4}{I_4}\frac{1}{(s\tau - m_R^2)^2 + m_R^2\Gamma_R^2} \\ &\quad + \frac{c_5}{I_5}\frac{1}{\tau(\tau+\tau_{R'})} + \frac{c_6}{I_6}\frac{1}{(s\tau - m_{R'}^2)^2 + m_{R'}^2\Gamma_{R'}^2} \,, \end{aligned} \tag{97}$$

$$h_y(y) = \frac{c_1}{I_1}(y - y_{\min}) + \frac{c_2}{I_2}(y_{\max} - y) + \frac{c_3}{I_3}\frac{1}{\cosh y} \,, \tag{98}$$

$$h_z(z) = \frac{c_1}{I_1} + \frac{c_2}{I_2}\frac{1}{a-z} + \frac{c_3}{I_3}\frac{1}{a+z} + \frac{c_4}{I_4}\frac{1}{(a-z)^2} + \frac{c_5}{I_5}\frac{1}{(a+z)^2} \,. \tag{99}$$

Here each term is separately integrable, with an invertible primitive function, such that generation of τ, y and z separately is a standard task, as described in section 4.1. In the following we describe the details of the scheme, including the meaning of the coefficients c_i and I_i, which are separate for τ, y and z.

The first variable to be selected is τ. The range of allowed values, $\tau_{\min} \le \tau \le \tau_{\max}$, is generally constrained by a number of user-defined requirements. A cut on the allowed mass range is directly reflected in τ, a cut on the $p_\perp$ range indirectly. The first two terms of h_τ are intended to represent a smooth τ dependence, as generally obtained in processes which do not receive contributions from s-channel resonances. Also s-channel exchange of essentially massless particles (γ, g, light quarks and leptons) are accounted for, since these do not produce any separate peaks at non-vanishing τ. The last four terms of h_τ are there to catch the peaks in the cross section from resonance production. These terms are only included when needed. Each resonance is represented by two pieces, a first to cover the interference with graphs which peak at $\tau = 0$, plus the variation of parton distributions, and a second to approximate the Breit–Wigner shape of the resonance itself. The subscripts R and R' denote values pertaining to the two resonances, with $\tau_R = m_R^2/s$. Currently there is only one process where the full structure with two resonances is used, namely $\mathrm{f\bar{f}} \to \gamma^*/\mathrm{Z}^0/\mathrm{Z}'^0$. Otherwise either one or no resonance peak is taken into account.

The kinematically allowed range of y values is constrained by τ, $|y| \le -\frac{1}{2}\ln\tau$, and you may impose additional cuts. Therefore the allowed range $y_{\min} \le y \le y_{\max}$ is only constructed after τ has been selected. The first two terms of h_y give a fairly flat y dependence — for processes which are symmetric in $y \leftrightarrow -y$, they will add to give a completely flat y spectrum between the allowed limits. In principle, the natural subdivision would have been one term flat in y and one forward–backward asymmetric, i.e. proportional to y. The latter is disallowed by the requirement of positivity, however. The $y - y_{\min}$ and $y_{\max} - y$ terms actually used give the same amount of freedom, but respect positivity. The third term is peaked at around $y = 0$, and represents the bias of parton distributions towards this region.

The allowed $z = \cos\hat{\theta}$ range is naïvely $-1 \le z \le 1$. However, most cross sections are divergent for $z \to \pm 1$, so some kind of regularization is necessary. Normally one requires $p_\perp \ge p_{\perp\min}$, which translates into $z^2 \le 1 - 4p_{\perp\min}^2/(\tau s)$ for massless outgoing particles. Since again the limits depend on τ, the selection of z is done after that of τ. Additional requirements may constrain the range further. In particular, a $p_{\perp\max}$ constraint may split the allowed z range into two, i.e. $z_{-\min} \le z \le z_{-\max}$ or $z_{+\min} \le z \le z_{+\max}$. An un-split range is represented by

$z_{-\max} = z_{+\min} = 0$. For massless outgoing particles the parameter $a = 1$ in h_z, such that the five terms represent a piece flat in angle and pieces peaked as $1/\hat{t}$, $1/\hat{u}$, $1/\hat{t}^2$, and $1/\hat{u}^2$, respectively. For non-vanishing masses one has $a = 1 + 2m_3^2 m_4^2/\hat{s}^2$. In this case, the full range $-1 \leq z \leq 1$ is therefore available — physically, the standard $\hat{t}$ and $\hat{u}$ singularities are regularized by the masses m_3 and m_4.

For each of the terms, the I_i coefficients represent the integral over the quantity multiplying the coefficient c_i; thus, for instance:

$$\begin{aligned}
h_\tau: \quad & I_1 = \int \frac{\mathrm{d}\tau}{\tau} = \ln\left(\frac{\tau_{\max}}{\tau_{\min}}\right) , \\
& I_2 = \int \frac{\mathrm{d}\tau}{\tau^2} = \frac{1}{\tau_{\min}} - \frac{1}{\tau_{\max}} ; \\
h_y: \quad & I_1 = \int (y - y_{\min})\,\mathrm{d}y = \frac{1}{2}(y_{\max} - y_{\min})^2 ; \\
h_z: \quad & I_1 = \int \mathrm{d}z = (z_{-\max} - z_{-\min}) + (z_{+\max} - z_{+\min}), \\
& I_2 = \int \frac{\mathrm{d}z}{a - z} = \ln\left(\frac{(a - z_{-\min})(a - z_{+\min})}{(a - z_{-\max})(a - z_{-\min})}\right) . \qquad (100)
\end{aligned}$$

The c_i coefficients are normalized to unit sum for h_τ, h_y and h_z separately. They have a simple interpretation, as the probability for each of the terms to be used in the preliminary selection of τ, y and z, respectively. The variation of the cross section over the allowed phase space is explored in the initialization procedure of a PYTHIA run, and based on this knowledge the c_i are optimized so as to give functions h_τ, h_y and h_z that closely follow the general behaviour of the true cross section. For instance, the coefficient c_4 in h_τ is to be made larger the more the total cross section is dominated by the region around the resonance mass.

The phase-space points tested at initialization are put on a grid, with the number of points in each dimension given by the number of terms in the respective h expression, and with the position of each point given by the median value of the distribution of one of the terms. For instance, the $\mathrm{d}\tau/\tau$ distribution gives a median point at $\sqrt{\tau_{\min}\tau_{\max}}$, and $\mathrm{d}\tau/\tau^2$ has the median $2\tau_{\min}\tau_{\max}/(\tau_{\min} + \tau_{\max})$. Since the allowed y and z ranges depend on the τ value selected, then so do the median points defined for these two variables.

With only a limited set of phase-space points studied at the initialization, the 'optimal' set of coefficients is not uniquely defined. To be on the safe side, 40% of the total weight is therefore assigned evenly between all allowed c_i, whereas the remaining 60% are assigned according to the relative importance surmised, under the constraint that no coefficient is allowed to receive a negative contribution from this second piece.

After a preliminary choice has been made of τ, y and z, it is necessary to find the weight of the event, which is to be used to determine whether to keep it or generate another one. Using the relation $\mathrm{d}\hat{t} = \hat{s}\beta_{34}\,\mathrm{d}z/2$, eq. (81) may be rewritten as

$$\begin{aligned}
\sigma &= \int\int\int \frac{\mathrm{d}\tau}{\tau}\mathrm{d}y\,\frac{\hat{s}\beta_{34}}{2}\mathrm{d}z\, x_1 f_1(x_1, Q^2)\, x_2 f_2(x_2, Q^2)\,\frac{\mathrm{d}\hat{\sigma}}{\mathrm{d}\hat{t}} \\
&= \frac{\pi}{s}\int h_\tau(\tau)\,\mathrm{d}\tau \int h_y(y)\,\mathrm{d}y \int h_z(z)\,\mathrm{d}z\,\beta_{34}\,\frac{x_1 f_1(x_1, Q^2)\, x_2 f_2(x_2, Q^2)}{\tau^2 h_\tau(\tau)\, h_y(y)\, 2h_z(z)}\,\frac{\hat{s}^2}{\pi}\frac{\mathrm{d}\hat{\sigma}}{\mathrm{d}\hat{t}} \\
&= \left\langle \frac{\pi}{s}\,\frac{\beta_{34}}{\tau^2 h_\tau(\tau)\, h_y(y)\, 2h_z(z)}\, x_1 f_1(x_1, Q^2)\, x_2 f_2(x_2, Q^2)\,\frac{\hat{s}^2}{\pi}\frac{\mathrm{d}\hat{\sigma}}{\mathrm{d}\hat{t}} \right\rangle . \qquad (101)
\end{aligned}$$

In the middle line, a factor of $1 = h_\tau/h_\tau$ has been introduced to rewrite the τ integral in terms of a phase space of unit volume: $\int h_\tau(\tau)\,\mathrm{d}\tau = 1$ according to the relations above. Correspondingly for the y and z integrals. In addition, factors of $1 = \hat{s}/(\tau s)$ and $1 = \pi/\pi$ are used to isolate the

dimensionless cross section $(\hat{s}^2/\pi)\,\mathrm{d}\hat{\sigma}/\mathrm{d}\hat{t}$. The content of the last line is that, with τ, y and z selected according to the expressions $h_\tau(\tau)$, $h_y(y)$ and $h_z(z)$, respectively, the cross section is obtained as the average of the final expression over all events. Since the h's have been picked to give unit volume, there is no need to multiply by the total phase-space volume.

As can be seen, the cross section for a given Monte Carlo event is given as the product of four factors, as follows:

1. The π/s factor, which is common to all events, gives the overall dimensions of the cross section, in GeV^{-2}. Since the final cross section is given in units of mb, the conversion factor of $1\ \mathrm{GeV}^{-2} = 0.3894$ mb is also included here.
2. Next comes the Jacobian, which compensates for the change from the original to the final phase-space volume.
3. The parton-distribution function weight is obtained by making use of the parton distribution libraries in PYTHIA or externally. The x_1 and x_2 values are obtained from τ and y via the relations $x_{1,2} = \sqrt{\tau}\exp(\pm y)$.
4. Finally, the dimensionless cross section $(\hat{s}^2/\pi)\,\mathrm{d}\hat{\sigma}/\mathrm{d}\hat{t}$ is the quantity that has to be coded for each process separately, and where the physics content is found.

Of course, the expression in the last line is not strictly necessary to obtain the cross section by Monte Carlo integration. One could also have used eq. (81) directly, selecting phase-space points evenly in τ, y and $\hat{t}$, and averaging over those Monte Carlo weights. Clearly this would be much simpler, but the price to be paid is that the weights of individual events could fluctuate wildly. For instance, if the cross section contains a narrow resonance, the few phase-space points that are generated in the resonance region obtain large weights, while the rest do not. With our procedure, a resonance would be included in the $h_\tau(\tau)$ factor, so that more events would be generated at around the appropriate τ_R value (owing to the h_τ numerator in the phase-space expression), but with these events assigned a lower, more normal weight (owing to the factor $1/h_\tau$ in the weight expression). Since the weights fluctuate less, fewer phase-space points need be selected to get a reasonable cross-section estimate.

In the program, the cross section is obtained as the average over all phase-space points generated. The events actually handed on to you should have unit weight, however (an option with weighted events exists, but does not represent the mainstream usage). At initialization, after the c_i coefficients have been determined, a search inside the allowed phase-space volume is therefore made to find the maximum of the weight expression in the last line of eq. (101). In the subsequent generation of events, a selected phase-space point is then retained with a probability equal to the weight in the point divided by the maximum weight. Only the retained phase-space points are considered further, and generated as complete events.

The search for the maximum is begun by evaluating the weight in the same grid of points as used to determine the c_i coefficients. The point with highest weight is used as starting point for a search towards the maximum. In unfortunate cases, the convergence could be towards a local maximum which is not the global one. To somewhat reduce this risk, also the grid point with second-highest weight is used for another search. After initialization, when events are generated, a warning message will be given by default at any time a phase-space point is selected where the weight is larger than the maximum, and thereafter the maximum weight is adjusted to reflect the new knowledge. This means that events generated before this time have a somewhat erroneous distribution in phase space, but if the maximum violation is rather modest the effects should be negligible. The estimation of the cross section is not affected by any of these considerations, since the maximum weight does not enter into eq. (101).

For $2 \to 2$ processes with identical final-state particles, the symmetrization factor of $1/2$ is explicitly included at the end of the $\mathrm{d}\hat{\sigma}/\mathrm{d}\hat{t}$ calculation. In the final cross section, a factor of 2 is retrieved because of integration over the full phase space (rather than only half of it). That way, no special provisions are needed in the phase-space integration machinery.

7.4.2 Resonance production

We have now covered the simple $2 \to 2$ case. In a $2 \to 1$ process, the $\hat{t}$ integral is absent, and the differential cross section $\mathrm{d}\hat{\sigma}/\mathrm{d}\hat{t}$ is replaced by $\hat{\sigma}(\hat{s})$. The cross section may now be written as

$$\begin{aligned}
\sigma &= \int\int \frac{\mathrm{d}\tau}{\tau}\,\mathrm{d}y\, x_1 f_1(x_1,Q^2)\, x_2 f_2(x_2,Q^2)\,\hat{\sigma}(\hat{s}) \\
&= \frac{\pi}{s}\int h_\tau(\tau)\,\mathrm{d}\tau \int h_y(y)\,\mathrm{d}y\, \frac{x_1 f_1(x_1,Q^2)\, x_2 f_2(x_2,Q^2)}{\tau^2 h_\tau(\tau)\, h_y(y)}\,\frac{\hat{s}}{\pi}\hat{\sigma}(\hat{s}) \\
&= \left\langle \frac{\pi}{s}\,\frac{1}{\tau^2 h_\tau(\tau)\, h_y(y)}\, x_1 f_1(x_1,Q^2)\, x_2 f_2(x_2,Q^2)\,\frac{\hat{s}}{\pi}\hat{\sigma}(\hat{s}) \right\rangle . \qquad (102)
\end{aligned}$$

The structure is thus exactly the same, but the z-related pieces are absent, and the rôle of the dimensionless cross section is played by $(\hat{s}/\pi)\hat{\sigma}(\hat{s})$.

If the range of allowed decay angles of the resonance is restricted, e.g. by requiring the decay products to have a minimum transverse momentum, effectively this translates into constraints on the $z = \cos\hat{\theta}$ variable of the $2 \to 2$ process. The difference is that the angular dependence of a resonance decay is trivial, and that therefore the z-dependent factor can be easily evaluated. For a spin-0 resonance, which decays isotropically, the relevant weight is simply $(z_{-\mathrm{max}} - z_{-\mathrm{min}})/2 + (z_{+\mathrm{max}} - z_{+\mathrm{min}})/2$. For a transversely polarized spin-1 resonance the expression is, instead,

$$\frac{3}{8}(z_{-\mathrm{max}} - z_{-\mathrm{min}}) + \frac{3}{8}(z_{+\mathrm{max}} - z_{+\mathrm{min}}) + \frac{1}{8}(z_{-\mathrm{max}} - z_{-\mathrm{min}})^3 + \frac{1}{8}(z_{+\mathrm{max}} - z_{+\mathrm{min}})^3 . \qquad (103)$$

Since the allowed z range could depend on τ and/or y (it does for a $p_\perp$ cut), the factor has to be evaluated for each individual phase-space point and included in the expression of eq. (102).

For $2 \to 2$ processes where either of the final-state particles is a resonance, or both, an additional choice has to be made for each resonance mass, eq. (94). Since the allowed τ, y and z ranges depend on m_3^2 and m_4^2, the selection of masses have to precede the choice of the other phase-space variables. Just as for the other variables, masses are not selected uniformly over the allowed range, but are rather distributed according to a function $h_m(m^2)\,dm^2$, with a compensating factor $1/h_m(m^2)$ in the Jacobian. The functional form picked is normally

$$h_m(m^2) = \frac{c_1}{I_1}\frac{1}{\pi}\frac{m_R\Gamma_R}{(m^2 - m_R^2)^2 + m_R^2\Gamma_R^2} + \frac{c_2}{I_2} + \frac{c_3}{I_3}\frac{1}{m^2} + \frac{c_4}{I_4}\frac{1}{m^4} . \qquad (104)$$

The definition of the I_i integrals is analogous to the one before. The c_i coefficients are not found by optimization, but predetermined, normally to $c_1 = 0.8$, $c_2 = c_3 = 0.1$, $c_4 = 0$. Clearly, had the phase space and the cross section been independent of m_3^2 and m_4^2, the optimal choice would have been to put $c_1 = 1$ and have all other c_i vanishing — then the $1/h_m$ factor of the Jacobian would exactly have cancelled the Breit–Wigner of eq. (94) in the cross section. The second and the third terms are there to cover the possibility that the cross section does not die away quite as fast as given by the naïve Breit–Wigner shape. In particular, the third term covers the possibility of a secondary peak at small m^2, in a spirit slightly similar to the one discussed for resonance production in $2 \to 1$ processes.

The fourth term is only used for processes involving γ^*/Z^0 production, where the γ propagator guarantees that the cross section does have a significant secondary peak for $m^2 \to 0$. Therefore here the choice is $c_1 = 0.4$, $c_2 = 0.05$, $c_3 = 0.3$ and $c_4 = 0.25$.

A few special tricks have been included to improve efficiency when the allowed mass range of resonances is constrained by kinematics or by user cuts. For instance, if a pair of equal or charge-conjugate resonances are produced, such as in $\mathrm{e}^+\mathrm{e}^- \to \mathrm{W}^+\mathrm{W}^-$, use is made of the constraint that the lighter of the two has to have a mass smaller than half the c.m. energy.

7.4.3 Lepton beams

Lepton beams have to be handled slightly differently from what has been described so far. One also has to distinguish between a lepton for which parton distributions are included and one which is treated as an unresolved point-like particle. The necessary modifications are the same for $2 \to 2$ and $2 \to 1$ processes, however, since the $\hat{t}$ degree of freedom is unaffected.

If one incoming beam is an unresolved lepton, the corresponding parton-distribution piece collapses to a δ function. This function can be used to integrate out the y variable: $\delta(x_{1,2} - 1) = \delta(y \pm (1/2)\ln\tau)$. It is therefore only necessary to select the τ and the z variables according to the proper distributions, with compensating weight factors, and only one set of parton distributions has to be evaluated explicitly.

If both incoming beams are unresolved leptons, both the τ and the y variables are trivially given: $\tau = 1$ and $y = 0$. Parton-distribution weights disappear completely. For a $2 \to 2$ process, only the z selection remains to be performed, while a $2 \to 1$ process is completely specified, i.e. the cross section is a simple number that only depends on the c.m. energy.

For a resolved electron, the $f_{\mathrm{e}}^{\mathrm{e}}$ parton distribution is strongly peaked towards $x = 1$. This affects both the τ and the y distributions, which are not well described by either of the pieces in $h_\tau(\tau)$ or $h_y(y)$ in processes with interacting $\mathrm{e}^\pm$. (Processes which involve e.g. the γ content of the e are still well simulated, since f_γ^{e} is peaked at small x.)

If both parton distributions are peaked close to 1, the $h_\tau(\tau)$ expression in eq. (99) is therefore increased with one additional term of the form $h_\tau(\tau) \propto 1/(1-\tau)$, with coefficients c_7 and I_7 determined as before. The divergence when $\tau \to 1$ is cut off by our regularization procedure for the $f_{\mathrm{e}}^{\mathrm{e}}$ parton distribution; therefore we only need consider $\tau < 1 - 2 \times 10^{-10}$.

Correspondingly, the $h_y(y)$ expression is expanded with a term $1/(1-\exp(y-y_0))$ when incoming beam number 1 consists of a resolved $\mathrm{e}^\pm$, and with a term $1/(1-\exp(-y-y_0))$ when incoming beam number 2 consists of a resolved $\mathrm{e}^\pm$. Both terms are present for an $\mathrm{e}^+\mathrm{e}^-$ collider, only one for an ep one. The coefficient $y_0 = -(1/2)\ln\tau$ is the naïve kinematical limit of the y range, $|y| < y_0$. From the definitions of y and y_0 it is easy to see that the two terms above correspond to $1/(1-x_1)$ and $1/(1-x_2)$, respectively, and thus are again regularized by our parton-distribution function cut-off. Therefore the integration ranges are $y < y_0 - 10^{-10}$ for the first term and $y > -y_0 + 10^{-10}$ for the second one.

7.4.4 Mixing processes

In the cross-section formulae given so far, we have deliberately suppressed a summation over the allowed incoming flavours. For instance, the process $\mathrm{f}\bar{\mathrm{f}} \to \mathrm{Z}^0$ in a hadron collider receives contributions from $\mathrm{u}\bar{\mathrm{u}} \to \mathrm{Z}^0$, $\mathrm{d}\bar{\mathrm{d}} \to \mathrm{Z}^0$, $\mathrm{s}\bar{\mathrm{s}} \to \mathrm{Z}^0$, and so on. These contributions share the same basic form, but differ in the parton-distribution weights and (usually) in a few coupling constants in the hard matrix elements. It it therefore convenient to generate the terms together, as follows:

1. A phase-space point is picked, and all common factors related to this choice are evaluated, i.e. the Jacobian and the common pieces of the matrix elements (e.g. for a Z^0 the basic Breit–Wigner shape, excluding couplings to the initial flavour).
2. The parton-distribution-function library is called to produce all the parton distributions, at the relevant x and Q^2 values, for the two incoming beams.
3. A loop is made over the two incoming flavours, one from each beam particle. For each allowed set of incoming flavours, the full matrix-element expression is constructed, using the common pieces and the flavour-dependent couplings. This is multiplied by the common factors and the parton-distribution weights to obtain a cross-section weight.
4. Each allowed flavour combination is stored as a separate entry in a table, together with its weight. In addition, a summed weight is calculated.
5. The phase-space point is kept or rejected, according to a comparison of the summed weight with the maximum weight obtained at initialization. Also the cross-section Monte Carlo integration is based on the summed weight.

6. If the point is retained, one of the allowed flavour combinations is picked according to the relative weights stored in the full table.

Generally, the flavours of the final state are either completely specified by those of the initial state, e.g. as in $qg \to qg$, or completely decoupled from them, e.g. as in $f\bar{f} \to Z^0 \to f'\bar{f}'$. In neither case need therefore the final-state flavours be specified in the cross-section calculation. It is only necessary, in the latter case, to include an overall weight factor, which takes into account the summed contribution of all final states that are to be simulated. For instance, if only the process $Z^0 \to e^+e^-$ is studied, the relevant weight factor is simply Γ_{ee}/Γ_{tot}. Once the kinematics and the incoming flavours have been selected, the outgoing flavours can be picked according to the appropriate relative probabilities.

In some processes, such as $gg \to gg$, several different colour flows are allowed, each with its own kinematical dependence of the matrix-element weight, see section 8.2.1. Each colour flow is then given as a separate entry in the table mentioned above, i.e. in total an entry is characterized by the two incoming flavours, a colour-flow index, and the weight. For an accepted phase-space point, the colour flow is selected in the same way as the incoming flavours.

The program can also allow the mixed generation of two or more completely different processes, such as $f\bar{f} \to Z^0$ and $q\bar{q} \to gg$. In that case, each process is initialized separately, with its own set of coefficients c_i and so on. The maxima obtained for the individual cross sections are all expressed in the same units, even when the dimensionality of the phase space is different. (This is because we always transform to a phase space of unit volume, $\int h_\tau(\tau)\, d\tau \equiv 1$, etc.) The above generation scheme need therefore only be generalized as follows:

1. One process is selected among the allowed ones, with a relative probability given by the maximum weight for this process.
2. A phase-space point is found, using the distributions $h_\tau(\tau)$ and so on, optimized for this particular process.
3. The total weight for the phase-space point is evaluated, again with Jacobians, matrix elements and allowed incoming flavour combinations that are specific to the process.
4. The point is retained with a probability given by the ratio of the actual to the maximum weight of the process. If the point is rejected, one has to go back to step 1 and pick a new process.
5. Once a phase-space point has been accepted, flavours may be selected, and the event generated in full.

It is clear why this works: although phase-space points are selected among the allowed processes according to relative probabilities given by the maximum weights, the probability that a point is accepted is proportional to the ratio of actual to maximum weight. In total, the probability for a given process to be retained is therefore only proportional to the average of the actual weights, and any dependence on the maximum weight is gone.

In γp and $\gamma\gamma$ physics, the different components of the photon give different final states, see section 7.7.2. Technically, this introduces a further level of administration, since each event class contains a set of (partly overlapping) processes. From an ideological point of view, however, it just represents one more choice to be made, that of event class, before the selection of process in step 1 above. When a weighting fails, both class and process have to be picked anew.

7.5 $2 \to 3$ and $2 \to 4$ Processes

The PYTHIA machinery to handle $2 \to 1$ and $2 \to 2$ processes is fairly sophisticated and generic. The same cannot be said about the generation of hard scattering processes with more than two final-state particles. The number of phase-space variables is larger, and it is therefore more difficult to find and transform away all possible peaks in the cross section by a suitably biased choice of phase-space points. In addition, matrix-element expressions for $2 \to 3$ processes are typically fairly lengthy. Therefore PYTHIA only contains a very limited number of $2 \to 3$ and $2 \to 4$ processes, and almost each process is a special case of its own. It is therefore less

interesting to discuss details, and we only give a very generic overview.

If the Higgs mass is not light, interactions among longitudinal W and Z gauge bosons are of interest. In the program, $2 \to 1$ processes such as $\mathrm{W}_\mathrm{L}^+\mathrm{W}_\mathrm{L}^- \to \mathrm{h}^0$ and $2 \to 2$ ones such as $\mathrm{W}_\mathrm{L}^+\mathrm{W}_\mathrm{L}^- \to \mathrm{Z}_\mathrm{L}^0\mathrm{Z}_\mathrm{L}^0$ are included. The former are for use when the h^0 still is reasonably narrow, such that a resonance description is applicable, while the latter are intended for high energies, where different contributions have to be added up. Since the program does not contain W_L or Z_L distributions inside hadrons, the basic hard scattering has to be convoluted with the $\mathrm{q} \to \mathrm{q}'\mathrm{W}_\mathrm{L}$ and $\mathrm{q} \to \mathrm{q}\mathrm{Z}_\mathrm{L}$ branchings, to yield effective $2 \to 3$ and $2 \to 4$ processes. However, it is possible to integrate out the scattering angles of the quarks analytically, as well as one energy-sharing variable [Cha85]. Only after an event has been accepted are these other kinematical variables selected. This involves further choices of random variables, according to a separate selection loop.

In total, it is therefore only necessary to introduce one additional variable in the basic phase-space selection, which is chosen to be $\hat{s}'$, the squared invariant mass of the full $2 \to 3$ or $2 \to 4$ process, while $\hat{s}$ is used for the squared invariant mass of the inner $2 \to 1$ or $2 \to 2$ process. The y variable is coupled to the full process, since parton-distribution weights have to be given for the original quarks at $x_{1,2} = \sqrt{\tau'}\exp(\pm y)$. The $\hat{t}$ variable is related to the inner process, and thus not needed for the $2 \to 3$ processes. The selection of the $\tau' = \hat{s}'/s$ variable is done after τ, but before y has been chosen. To improve the efficiency, the selection is made according to a weighted phase space of the form $\int h_{\tau'}(\tau')\,\mathrm{d}\tau'$, where

$$h_{\tau'}(\tau') = \frac{c_1}{I_1}\frac{1}{\tau'} + \frac{c_2}{I_2}\frac{(1-\tau/\tau')^3}{\tau'^2} + \frac{c_3}{I_3}\frac{1}{\tau'(1-\tau')} \;, \tag{105}$$

in conventional notation. The c_i coefficients are optimized at initialization. The c_3 term, peaked at $\tau' \approx 1$, is only used for $\mathrm{e}^+\mathrm{e}^-$ collisions. The choice of $h_{\tau'}$ is roughly matched to the longitudinal gauge-boson flux factor, which is of the form

$$\left(1+\frac{\tau}{\tau'}\right)\ln\left(\frac{\tau}{\tau'}\right) - 2\left(1-\frac{\tau}{\tau'}\right) \;. \tag{106}$$

For a light h the effective W approximation above breaks down, and it is necessary to include the full structure of the $\mathrm{qq}' \to \mathrm{qq}'\mathrm{h}^0$ (i.e. ZZ fusion) and $\mathrm{qq}' \to \mathrm{q}''\mathrm{q}'''\mathrm{h}^0$ (i.e. WW fusion) matrix elements. The τ', τ and y variables are here retained, and selected according to standard procedures. The Higgs mass is represented by the τ choice; normally the h^0 is so narrow that the τ distribution effectively collapses to a δ function. In addition, the three-body final-state phase space is rewritten as

$$\left(\prod_{i=3}^{5}\frac{1}{(2\pi)^3}\frac{\mathrm{d}^3p_i}{2E_i}\right)(2\pi)^4\delta^{(4)}(p_3+p_4+p_5-p_1-p_2) = \frac{1}{(2\pi)^5}\frac{\pi^2}{4\sqrt{\lambda_{\perp 34}}}\,\mathrm{d}p_{\perp 3}^2\,\frac{\mathrm{d}\varphi_3}{2\pi}\,\mathrm{d}p_{\perp 4}^2\,\frac{\mathrm{d}\varphi_4}{2\pi}\,\mathrm{d}y_5 \;, \tag{107}$$

where $\lambda_{\perp 34} = (m_{\perp 34}^2 - m_{\perp 3}^2 - m_{\perp 4}^2)^2 - 4m_{\perp 3}^2 m_{\perp 4}^2$. The outgoing quarks are labelled 3 and 4, and the outgoing Higgs 5. The φ angles are selected isotropically, while the two transverse momenta are picked, with some foreknowledge of the shape of the W/Z propagators in the cross sections, according to $h_\perp(p_\perp^2)\,\mathrm{d}p_\perp^2$, where

$$h_\perp(p_\perp^2) = \frac{c_1}{I_1} + \frac{c_2}{I_2}\frac{1}{m_R^2+p_\perp^2} + \frac{c_3}{I_3}\frac{1}{(m_R^2+p_\perp^2)^2} \;, \tag{108}$$

with m_R the W or Z mass, depending on process, and $c_1 = c_2 = 0.05$, $c_3 = 0.9$. Within the limits given by the other variable choices, the rapidity y_5 is chosen uniformly. A final choice remains to be made, which comes from a twofold ambiguity of exchanging the longitudinal momenta of partons 3 and 4 (with minor modifications if they are massive). Here the relative weight can be obtained exactly from the form of the matrix element itself.

7.6 Resonance Decays

Resonances (see section 2.1.2) can be made to decay in two different routines. One is the standard decay treatment (in PYDECY) that can be used for any unstable particle, where decay channels are chosen according to fixed probabilities, and decay angles usually are picked isotropically in the rest frame of the resonance, see section 13.3. The more sophisticated treatment (in PYRESD) is the default one for resonances produced in PYTHIA, and is described here. The ground rule is that everything in mass up to and including b hadrons is decayed with the simpler PYDECY routine, while heavier particles are handled with PYRESD. This also includes the γ^*/Z^0, even though here the mass in principle could be below the b threshold. Other resonances include, e.g., t, $\mathrm{W}^\pm$, h^0, Z'^0, $\mathrm{W}'^\pm$, H^0, A^0, $\mathrm{H}^\pm$, and technicolor and supersymmetric particles.

7.6.1 The decay scheme

In the beginning of the decay treatment, either one or two resonances may be present, the former represented by processes such as $\mathrm{q}\overline{\mathrm{q}}' \to \mathrm{W}^+$ and $\mathrm{qg} \to \mathrm{W}^+\mathrm{q}'$, the latter by $\mathrm{q}\overline{\mathrm{q}} \to \mathrm{W}^+\mathrm{W}^-$. If the latter is the case, the decay of the two resonances is considered in parallel (unlike PYDECY, where one particle at a time is made to decay).

First the decay channel of each resonance is selected according to the relative weights $H_R^{(f)}$, as described above, evaluated at the actual mass of the resonance, rather than at the nominal one. Threshold factors are therefore fully taken into account, with channels automatically switched off below the threshold. Normally the masses of the decay products are well-defined, but e.g. in decays like $\mathrm{h}^0 \to \mathrm{W}^+\mathrm{W}^-$ it is also necessary to select the decay product masses. This is done according to two Breit–Wigners of the type in eq. (94), multiplied by the threshold factor, which depends on both masses.

Next the decay angles of the resonance are selected isotropically in its rest frame. Normally the full range of decay angles is available, but in $2 \to 1$ processes the decay angles of the original resonance may be restrained by user cuts, e.g. on the $p_\perp$ of the decay products. Based on the angles, the four-momenta of the decay products are constructed and boosted to the correct frame. As a rule, matrix elements are given with quark and lepton masses assumed vanishing. Therefore the four-momentum vectors constructed at this stage are actually massless for all quarks and leptons.

The matrix elements may now be evaluated. For a process such as $\mathrm{q}\overline{\mathrm{q}} \to \mathrm{W}^+\mathrm{W}^- \to \mathrm{e}^+\nu_\mathrm{e}\mu^-\overline{\nu}_\mu$, the matrix element is a function of the four-momenta of the two incoming fermions and of the four outgoing ones. An upper limit for the event weight can be constructed from the cross section for the basic process $\mathrm{q}\overline{\mathrm{q}} \to \mathrm{W}^+\mathrm{W}^-$, as already used to select the two W momenta. If the weighting fails, new resonance decay angles are picked and the procedure is iterated until acceptance.

Based on the accepted set of angles, the correct decay product four-momenta are constructed, including previously neglected fermion masses. Quarks and, optionally, leptons are allowed to radiate, using the standard final-state showering machinery, with maximum virtuality given by the resonance mass.

In some decays new resonances are produced, and these are then subsequently allowed to decay. Normally only one resonance pair is considered at a time, with the possibility of full correlations. In a few cases triplets can also appear, but such configurations currently are considered without inclusion of correlations. Also note that, in a process like $\mathrm{q}\overline{\mathrm{q}} \to \mathrm{Z}^0\mathrm{h}^0 \to \mathrm{Z}^0\mathrm{W}^+\mathrm{W}^- \to 6$ fermions, the spinless nature of the h^0 ensures that the $\mathrm{W}^\pm$ decays are decoupled from that of the Z^0 (but not from each other).

7.6.2 Cross-section considerations

The cross section for a process which involves the production of one or several resonances is always reduced to take into account channels not allowed by user flags. This is trivial for a single

s-channel resonance, cf. eq. (91), but can also be included approximately if several layers of resonance decays are involved. At initialization, the ratio between the user-allowed width and the nominally possible one is evaluated and stored, starting from the lightest resonances and moving upwards. As an example, one first finds the reduction factors for W^+ and for W^- decays, which need not be the same if e.g. W^+ is allowed to decay only to quarks and W^- only to leptons. These factors enter together as a weight for the $h^0 \to W^+W^-$ channel, which is thus reduced in importance compared with other possible Higgs decay channels. This is also reflected in the weight factor of the h^0 itself, where some channels are open in full, others completely closed, and finally some (like the one above) open but with reduced weight. Finally, the weight for the process $q\overline{q} \to Z^0h^0$ is evaluated as the product of the Z^0 weight factor and the h^0 one. The standard cross section of the process is multiplied with this weight.

Since the restriction on allowed decay modes is already included in the hard process cross section, mixing of different event types is greatly simplified, and the selection of decay channel chains is straightforward. There is a price to be paid, however. The reduction factors evaluated at initialization all refer to resonances at their nominal masses. For instance, the W reduction factor is evaluated at the nominal W mass, even when that factor is used, later on, in the description of the decay of a 120 GeV Higgs, where at least one W would be produced below this mass. We know of no case where this approximation has any serious consequences, however.

The weighting procedure works because the number of resonances to be produced, directly or in subsequent decays, can be derived recursively already from the start. It does not work for particles which could also be produced at later stages, such as the parton-shower evolution and the fragmentation. For instance, D^0 mesons can be produced fairly late in the event generation chain, in unknown numbers, and so weights could not be introduced to compensate, e.g. for the forcing of decays only into π^+K^-.

One should note that this reduction factor is separate from the description of the resonance shape itself, where the full width of the resonance has to be used. This width is based on the sum of all possible decay modes, not just the simulated ones. PYTHIA does allow the possibility to change also the underlying physics scenario, e.g. to include the decay of a Z^0 into a fourth-generation neutrino.

Normally the evaluation of the reduction factors is straightforward. However, for decays into a pair of equal or charge-conjugate resonances, such as Z^0Z^0 or W^+W^-, it is possible to pick combinations in such a way that the weight of the pair does not factorize into a product of the weight of each resonance itself. To be precise, any decay channel can be given seven different status codes:

- -1: a non-existent decay mode, completely switched off and of no concern to us;
- 0: an existing decay channel, which is switched off;
- 1: a channel which is switched on;
- 2: a channel switched on for particles, but off for antiparticles;
- 3: a channel switched on for antiparticles, but off for particles;
- 4: a channel switched on for one of the particles or antiparticles, but not for both;
- 5: a channel switched on for the other of the particles or antiparticles, but not for both.

The meaning of possibilities 4 and 5 is exemplified by the statement 'in a W^+W^- pair, one W decays hadronically and the other leptonically', which thus covers the cases where either W^+ or W^- decays hadronically.

Neglecting non-existing channels, each channel belongs to either of the classes above. If we denote the total branching ratio into channels of type i by r_i, this then translates into the requirement $r_0 + r_1 + r_2 + r_3 + r_4 + r_5 = 1$. For a single particle the weight factor is $r_1 + r_2 + r_4$, and for a single antiparticle $r_1 + r_3 + r_4$. For a pair of identical resonances, the joint weight is instead

$$(r_1 + r_2)^2 + 2(r_1 + r_2)(r_4 + r_5) + 2r_4r_5 \ , \tag{109}$$

and for a resonance–antiresonance pair

$$(r_1 + r_2)(r_1 + r_3) + (2r_1 + r_2 + r_3)(r_4 + r_5) + 2r_4r_5 \ . \tag{110}$$

If some channels come with a reduced weight because of restrictions on subsequent decay chains, this may be described in terms of properly reduced r_i, so that the sum is less than unity. For instance, in a $\mathrm{t\bar{t}} \to \mathrm{bW^+\bar{b}W^-}$ process, the W decay modes may be restricted to $\mathrm{W^+} \to \mathrm{q\bar{q}}$ and $\mathrm{W^-} \to \mathrm{e^-\bar{\nu}_e}$, in which case $(\sum r_i)_{\mathrm{t}} \approx 2/3$ and $(\sum r_i)_{\bar{\mathrm{t}}} \approx 1/9$. With index $\pm$ denoting resonance/antiresonance, eq. (110) then generalizes to

$$(r_1+r_2)^+(r_1+r_3)^- + (r_1+r_2)^+(r_4+r_5)^- + (r_4+r_5)^+(r_1+r_3)^- + r_4^+ r_5^- + r_5^+ r_4^- \ . \qquad (111)$$

7.7 Nonperturbative Processes

A few processes are not covered by the discussion so far. These are the ones that depend on the details of hadronic wave functions, and therefore are not strictly calculable perturbatively (although perturbation theory may often provide some guidance). What we have primarily in mind is elastic scattering, diffractive scattering and low-$p_\perp$ 'minimum-bias' events in hadron–hadron collisions, but one can also find corresponding processes in γp and $\gamma\gamma$ interactions. The description of these processes is rather differently structured from that of the other ones, as is explained below. Models for 'minimum-bias' events are discussed in detail in section 11.2, to which we refer for details on this part of the program.

7.7.1 Hadron–hadron interactions

In hadron–hadron interactions, the total hadronic cross section for $AB \to$ anything, $\sigma_{\mathrm{tot}}^{AB}$, is calculated using the parameterization of Donnachie and Landshoff [Don92]. In this approach, each cross section appears as the sum of one pomeron term and one reggeon one

$$\sigma_{\mathrm{tot}}^{AB}(s) = X^{AB} s^{\varepsilon} + Y^{AB} s^{-\eta} \ , \qquad (112)$$

where $s = E_{\mathrm{cm}}^2$. The powers $\varepsilon = 0.0808$ and $\eta = 0.4525$ are expected to be universal, whereas the coefficients X^{AB} and Y^{AB} are specific to each initial state. (In fact, the high-energy behaviour given by the pomeron term is expected to be the same for particle and antiparticle interactions, i.e. $X^{\overline{A}B} = X^{AB}$.) Parameterizations not provided in [Don92] have been calculated in the same spirit, making use of quark counting rules [Sch93a].

The total cross section is subdivided according to

$$\sigma_{\mathrm{tot}}^{AB}(s) = \sigma_{\mathrm{el}}^{AB}(s) + \sigma_{\mathrm{sd}(XB)}^{AB}(s) + \sigma_{\mathrm{sd}(AX)}^{AB}(s) + \sigma_{\mathrm{dd}}^{AB}(s) + \sigma_{\mathrm{nd}}^{AB}(s) \ . \qquad (113)$$

Here 'el' is the elastic process $AB \to AB$, 'sd(XB)' the single diffractive $AB \to XB$, 'sd(AX)' the single diffractive $AB \to AX$, 'dd' the double diffractive $AB \to X_1X_2$, and 'nd' the non-diffractive ones. Higher diffractive topologies, such as central diffraction, are currently neglected. In the following, the elastic and diffractive cross sections and event characteristics are described, as given in the model by Schuler and Sjöstrand [Sch94, Sch93a]. The non-diffractive component is identified with the 'minimum bias' physics already mentioned, a practical but not unambiguous choice. Its cross section is given by 'whatever is left' according to eq. (113), and its properties are discussed in section 11.2.

At not too large squared momentum transfers t, the elastic cross section can be approximated by a simple exponential fall-off. If one neglects the small real part of the cross section, the optical theorem then gives

$$\frac{\mathrm{d}\sigma_{\mathrm{el}}}{\mathrm{d}t} = \frac{\sigma_{\mathrm{tot}}^2}{16\pi} \exp(B_{\mathrm{el}} t) \ , \qquad (114)$$

and $\sigma_{\mathrm{el}} = \sigma_{\mathrm{tot}}^2/16\pi B_{\mathrm{el}}$. The elastic slope parameter is parameterized by

$$B_{\mathrm{el}} = B_{\mathrm{el}}^{AB}(s) = 2b_A + 2b_B + 4s^{\varepsilon} - 4.2 \ , \qquad (115)$$

with s given in units of GeV and $B_{\rm el}$ in GeV^{-2}. The constants $b_{A,B}$ are $b_{\rm p} = 2.3$, $b_{\pi,\rho,\omega,\phi} = 1.4$, $b_{\rm J/\psi} = 0.23$. The increase of the slope parameter with c.m. energy is faster than the logarithmically one conventionally assumed; that way the ratio $\sigma_{\rm el}/\sigma_{\rm tot}$ remains well-behaved at large energies.

The diffractive cross sections are given by

$$\begin{aligned}
\frac{{\rm d}\sigma_{{\rm sd}(XB)}(s)}{{\rm d}t\,{\rm d}M^2} &= \frac{g_{3\mathbb{P}}}{16\pi}\beta_{A\mathbb{P}}\,\beta_{B\mathbb{P}}^2\,\frac{1}{M^2}\,\exp(B_{{\rm sd}(XB)}t)\,F_{\rm sd}\ , \\
\frac{{\rm d}\sigma_{{\rm sd}(AX)}(s)}{{\rm d}t\,{\rm d}M^2} &= \frac{g_{3\mathbb{P}}}{16\pi}\beta_{A\mathbb{P}}^2\,\beta_{B\mathbb{P}}\,\frac{1}{M^2}\,\exp(B_{{\rm sd}(AX)}t)\,F_{\rm sd}\ , \\
\frac{{\rm d}\sigma_{\rm dd}(s)}{{\rm d}t\,{\rm d}M_1^2\,{\rm d}M_2^2} &= \frac{g_{3\mathbb{P}}^2}{16\pi}\beta_{A\mathbb{P}}\,\beta_{B\mathbb{P}}\,\frac{1}{M_1^2}\,\frac{1}{M_2^2}\,\exp(B_{\rm dd}t)\,F_{\rm dd}\ .
\end{aligned} \tag{116}$$

The couplings $\beta_{A\mathbb{P}}$ are related to the pomeron term $X^{AB}s^{\varepsilon}$ of the total cross section parameterization, eq. (112). Picking a reference scale $\sqrt{s_{\rm ref}} = 20$ GeV, the couplings are given by $\beta_{A\mathbb{P}}\beta_{B\mathbb{P}} = X^{AB}s_{\rm ref}^{\varepsilon}$. The triple-pomeron coupling is determined from single-diffractive data to be $g_{3\mathbb{P}} \approx 0.318$ mb$^{1/2}$; within the context of the formulae in this section.

The spectrum of diffractive masses M is taken to begin 0.28 GeV $\approx 2m_\pi$ above the mass of the respective incoming particle and extend to the kinematical limit. The simple ${\rm d}M^2/M^2$ form is modified by the mass-dependence in the diffractive slopes and in the $F_{\rm sd}$ and $F_{\rm dd}$ factors (see below).

The slope parameters are assumed to be

$$\begin{aligned}
B_{{\rm sd}(XB)}(s) &= 2b_B + 2\alpha'\ln\left(\frac{s}{M^2}\right)\ , \\
B_{{\rm sd}(AX)}(s) &= 2b_A + 2\alpha'\ln\left(\frac{s}{M^2}\right)\ , \\
B_{\rm dd}(s) &= 2\alpha'\ln\left(e^4 + \frac{ss_0}{M_1^2M_2^2}\right)\ .
\end{aligned} \tag{117}$$

Here $\alpha' = 0.25$ GeV^{-2} and conventionally s_0 is picked as $s_0 = 1/\alpha'$. The term e^4 in $B_{\rm dd}$ is added by hand to avoid a breakdown of the standard expression for large values of $M_1^2M_2^2$. The $b_{A,B}$ terms protect $B_{\rm sd}$ from breaking down; however a minimum value of 2 GeV^{-2} is still explicitly required for $B_{\rm sd}$, which comes into play e.g. for a J/ψ state (as part of a VMD photon beam).

The kinematical range in t depends on all the masses of the problem. In terms of the scaled variables $\mu_1 = m_A^2/s$, $\mu_2 = m_B^2/s$, $\mu_3 = M_{(1)}^2/s$ ($= m_A^2/s$ when A scatters elastically), $\mu_4 = M_{(2)}^2/s$ ($= m_B^2/s$ when B scatters elastically), and the combinations

$$\begin{aligned}
C_1 &= 1-(\mu_1+\mu_2+\mu_3+\mu_4)+(\mu_1-\mu_2)(\mu_3-\mu_4)\ , \\
C_2 &= \sqrt{(1-\mu_1-\mu_2)^2-4\mu_1\mu_2}\,\sqrt{(1-\mu_3-\mu_4)^2-4\mu_3\mu_4}\ , \\
C_3 &= (\mu_3-\mu_1)(\mu_4-\mu_2)+(\mu_1+\mu_4-\mu_2-\mu_3)(\mu_1\mu_4-\mu_2\mu_3)\ ,
\end{aligned} \tag{118}$$

one has $t_{\rm min} < t < t_{\rm max}$ with

$$\begin{aligned}
t_{\rm min} &= -\frac{s}{2}(C_1+C_2)\ , \\
t_{\rm max} &= -\frac{s}{2}(C_1-C_2) = -\frac{s}{2}\,\frac{4C_3}{C_1+C_2} = \frac{s^2C_3}{t_{\rm min}}\ .
\end{aligned} \tag{119}$$

The Regge formulae above for single- and double-diffractive events are supposed to hold in certain asymptotic regions of the total phase space. Of course, there will be diffraction also

outside these restrictive regions. Lacking a theory which predicts differential cross sections at arbitrary t and M^2 values, the Regge formulae are used everywhere, but fudge factors are introduced in order to obtain 'sensible' behaviour in the full phase space. These factors are:

$$\begin{aligned} F_{\mathrm{sd}} &= \left(1-\frac{M^2}{s}\right)\left(1+\frac{c_{\mathrm{res}}M_{\mathrm{res}}^2}{M_{\mathrm{res}}^2+M^2}\right) , \\ F_{\mathrm{dd}} &= \left(1-\frac{(M_1+M_2)^2}{s}\right)\left(\frac{s\,m_{\mathrm{p}}^2}{s\,m_{\mathrm{p}}^2+M_1^2M_2^2}\right) \\ &\times \left(1+\frac{c_{\mathrm{res}}M_{\mathrm{res}}^2}{M_{\mathrm{res}}^2+M_1^2}\right)\left(1+\frac{c_{\mathrm{res}}M_{\mathrm{res}}^2}{M_{\mathrm{res}}^2+M_2^2}\right) . \end{aligned} \tag{120}$$

The first factor in either expression suppresses production close to the kinematical limit. The second factor in F_{dd} suppresses configurations where the two diffractive systems overlap in rapidity space. The final factors give an enhancement of the low-mass region, where a resonance structure is observed in the data. Clearly a more detailed modelling would have to be based on a set of exclusive states rather than on this smeared-out averaging procedure. A reasonable fit to $\mathrm{pp}/\overline{\mathrm{p}}\mathrm{p}$ data is obtained for $c_{\mathrm{res}} = 2$ and $M_{\mathrm{res}} = 2$ GeV, for an arbitrary particle A which is diffractively excited we use $M_{\mathrm{res}}^A = m_A - m_{\mathrm{p}} + 2$ GeV.

The diffractive cross-section formulae above have been integrated for a set of c.m. energies, starting at 10 GeV, and the results have been parameterized. The form of these parameterizations is given in ref. [Sch94], with explicit numbers for the $\mathrm{pp}/\overline{\mathrm{p}}\mathrm{p}$ case. PYTHIA also contains similar parameterizations for $\pi\mathrm{p}$ (assumed to be same as $\rho\mathrm{p}$ and $\omega\mathrm{p}$), $\phi\mathrm{p}$, $\mathrm{J}/\psi\mathrm{p}$, $\rho\rho$ ($\pi\pi$ etc.), $\rho\phi$, $\rho\mathrm{J}/\psi$, $\phi\phi$, $\phi\mathrm{J}/\psi$ and $\mathrm{J}/\psi\mathrm{J}/\psi$.

The processes above do not obey the ordinary event mixing strategy. First of all, since their total cross sections are known, it is possible to pick the appropriate process from the start, and then remain with that choice. In other words, if the selection of kinematical variables fails, one would not go back and pick a new process, the way it was done in section 7.4.4. Second, it is not possible to impose any cuts or restrain allowed incoming or outgoing flavours: if not additional information were to be provided, it would make the whole scenario ill-defined. Third, it is not recommended to mix generation of these processes with that of any of the other ones: normally the other processes have so small cross sections that they would almost never be generated anyway. (We here exclude the cases of 'underlying events' and 'pile-up events', where mixing is provided for, and even is a central part of the formalism, see sections 11.2 and 11.3.)

Once the cross-section parameterizations has been used to pick one of the processes, the variables t and M are selected according to the formulae given above.

A ρ^0 formed by $\gamma \to \rho^0$ in elastic or diffractive scattering is polarized, and therefore its decay angular distribution in $\rho^0 \to \pi^+\pi^-$ is taken to be proportional to $\sin^2\theta$, where the reference axis is given by the ρ^0 direction of motion.

A light diffractive system, with a mass less than 1 GeV above the mass of the incoming particle, is allowed to decay isotropically into a two-body state. Single-resonance diffractive states, such as a Δ^+, are therefore not explicitly generated, but are assumed described in an average, smeared-out sense.

A more massive diffractive system is subsequently treated as a string with the quantum numbers of the original hadron. Since the exact nature of the pomeron exchanged between the hadrons is unknown, two alternatives are included. In the first, the pomeron is assumed to couple to (valence) quarks, so that the string is stretched directly between the struck quark and the remnant diquark (antiquark) of the diffractive state. In the second, the interaction is rather with a gluon, giving rise to a 'hairpin' configuration in which the string is stretched from a quark to a gluon and then back to a diquark (antiquark). Both of these scenarios could be present in the data; the default choice is to mix them in equal proportions.

There is experimental support for more complicated scenarios [Ing85], wherein the pomeron has a partonic substructure, which e.g. can lead to high-$p_\perp$ jet production in the diffractive

system. The full machinery, wherein a pomeron spectrum is convoluted with a pomeron-proton hard interaction, is not available in PYTHIA. (But is found in the POMPYT program [Bru96].)

7.7.2 Photoproduction and $\gamma\gamma$ physics

The photon physics machinery in PYTHIA has been largely expanded in recent years. Historically, the model was first developed for photoproduction, i.e. a real photon on a hadron target [Sch93, Sch93a]. Thereafter $\gamma\gamma$ physics was added in the same spirit [Sch94a, Sch97]. Only recently also virtual photons have been added to the description [Fri00], including the nontrivial transition region between real photons and Deeply Inelastic Scattering (DIS). In this section we partly trace this evolution towards more complex configurations.

The total γp and $\gamma\gamma$ cross sections can again be parameterized in a form like eq. (112), which is not so obvious since the photon has more complicated structure than an ordinary hadron. In fact, the structure is still not so well understood. The model we outline is the one studied by Schuler and Sjöstrand [Sch93, Sch93a], and further updated in [Fri00]. In this model the physical photon is represented by

$$|\gamma\rangle = \sqrt{Z_3}\,|\gamma_B\rangle + \sum_{V=\rho^0,\omega,\phi,J/\psi} \frac{e}{f_V}\,|V\rangle + \sum_{q} \frac{e}{f_{q\overline{q}}}\,|q\overline{q}\rangle + \sum_{\ell=e,\mu,\tau} \frac{e}{f_{\ell\ell}}\,|\ell^+\ell^-\rangle\ . \tag{121}$$

By virtue of this superposition, one is led to a model of γp interactions, where three different kinds of events may be distinguished:

- Direct events, wherein the bare photon $|\gamma_B\rangle$ interacts directly with a parton from the proton. The process is perturbatively calculable, and no parton distributions of the photon are involved. The typical event structure is two high-$p_\perp$ jets and a proton remnant, while the photon does not leave behind any remnant.
- VMD events, in which the photon fluctuates into a vector meson, predominantly a ρ^0. All the event classes known from ordinary hadron–hadron interactions may thus occur here, such as elastic, diffractive, low-$p_\perp$ and high-$p_\perp$ events. For the latter, one may define (VMD) parton distributions of the photon, and the photon also leaves behind a beam remnant. This remnant is smeared in transverse momentum by a typical 'primordial $k_\perp$' of a few hundred MeV.
- Anomalous or GVMD (Generalized VMD) events, in which the photon fluctuates into a $q\overline{q}$ pair of larger virtuality than in the VMD class. The initial parton distribution is perturbatively calculable, as is the subsequent QCD evolution. It gives rise to the so-called anomalous part of the parton distributions of the photon, whence one name for the class. As long as only real photons were considered, it made sense to define the cross section of this event class to be completely perturbatively calculable, given some lower $p_\perp$ cut-off. Thus only high-$p_\perp$ events could occur. However, alternatively, one may view these states as excited higher resonances (ρ' etc.), thus the GVMD name. In this case one is lead to a picture which also allows a low-$p_\perp$ cross section, uncalculable in perturbation theory. The reality may well interpolate between these two extreme alternatives, but the current framework more leans towards the latter point of view. Either the q or the $\overline{q}$ plays the rôle of a beam remnant, but this remnant has a larger $p_\perp$ than in the VMD case, related to the virtuality of the $\gamma \leftrightarrow q\overline{q}$ fluctuation.

The $|\ell^+\ell^-\rangle$ states can only interact strongly with partons inside the hadron at higher orders, and can therefore be neglected in the study of hadronic final states.

In order that the above classification is smooth and free of double counting, one has to introduce scales that separate the three components. The main one is k_0, which separates the low-mass vector meson region from the high-mass $|q\overline{q}\rangle$ one, $k_0 \approx m_\phi/2 \approx 0.5$ GeV. Given this dividing line to VMD states, the anomalous parton distributions are perturbatively calculable. The total cross section of a state is not, however, since this involves aspects of soft physics and eikonalization of jet rates. Therefore an ansatz is chosen where the total cross section of

a state scales like $k_V^2/k_\perp^2$, where the adjustable parameter $k_V \approx m_\rho/2$ for light quarks. The $k_\perp$ scale is roughly equated with half the mass of the GVMD state. The spectrum of GVMD states is taken to extend over a range $k_0 < k_\perp < k_1$, where k_1 is identified with the $p_{\perp\min}(s)$ cut-off of the perturbative jet spectrum in hadronic interactions, $p_{\perp\min}(s) \approx 1.5$ GeV at typical energies, see section 11.2 and especially eq. (189). Above that range, the states are assumed to be sufficiently weakly interacting that no eikonalization procedure is required, so that cross sections can be calculated perturbatively without any recourse to pomeron phenomenology. There is some arbitrariness in that choice, and some simplifications are required in order to obtain a manageable description.

The VMD and GVMD/anomalous events are together called resolved ones. In terms of high-$p_\perp$ jet production, the VMD and anomalous contributions can be combined into a total resolved one, and the same for parton-distribution functions. However, the two classes differ in the structure of the underlying event and possibly in the appearance of soft processes.

In terms of cross sections, eq. (121) corresponds to

$$\sigma_{\rm tot}^{\gamma p}(s) = \sigma_{\rm dir}^{\gamma p}(s) + \sigma_{\rm VMD}^{\gamma p}(s) + \sigma_{\rm anom}^{\gamma p}(s) \ . \tag{122}$$

The direct cross section is, to lowest order, the perturbative cross section for the two processes $\gamma q \to qg$ and $\gamma g \to q\bar{q}$, with a lower cut-off $p_\perp > k_1$, in order to avoid double-counting with the interactions of the GVMD states. Properly speaking, this should be multiplied by the Z_3 coefficient,

$$Z_3 = 1 - \sum_{V=\rho^0,\omega,\phi,J/\psi} \left(\frac{e}{f_V}\right)^2 - \sum_q \left(\frac{e}{f_{q\bar{q}}}\right)^2 - \sum_{\ell=e,\mu,\tau} \left(\frac{e}{f_{\ell\ell}}\right)^2 \ , \tag{123}$$

but normally Z_3 is so close to unity as to make no difference.

The VMD factor $(e/f_V)^2 = 4\pi\alpha_{\rm em}/f_V^2$ gives the probability for the transition $\gamma \to V$. The coefficients $f_V^2/4\pi$ are determined from data to be (with a non-negligible amount of uncertainty) 2.20 for ρ^0, 23.6 for ω, 18.4 for ϕ and 11.5 for J/ψ. Together these numbers imply that the photon can be found in a VMD state about 0.4% of the time, dominated by the ρ^0 contribution. All the properties of the VMD interactions can be obtained by appropriately scaling down Vp physics predictions. Thus the whole machinery developed in the previous section for hadron–hadron interactions is directly applicable. Also parton distributions of the VMD component inside the photon are obtained by suitable rescaling.

The contribution from the 'anomalous' high-mass fluctuations to the total cross section is obtained by a convolution of the fluctuation rate

$$\sum_q \left(\frac{e}{f_{q\bar{q}}}\right)^2 \approx \frac{\alpha_{\rm em}}{2\pi} \left(2\sum_q e_q^2\right) \int_{k_0}^{k_1} \frac{dk_\perp^2}{k_\perp^2} \ , \tag{124}$$

which is to be multiplied by the abovementioned reduction factor $k_V^2/k_\perp^2$ for the total cross section, and all scaled by the assumed real vector meson cross section.

As an illustration of this scenario, the phase space of γp events may be represented by a $(k_\perp, p_\perp)$ plane. Two transverse momentum scales are distinguished: the photon resolution scale $k_\perp$ and the hard interaction scale $p_\perp$. Here $k_\perp$ is a measure of the virtuality of a fluctuation of the photon and $p_\perp$ corresponds to the most virtual rung of the ladder, possibly apart from $k_\perp$. As we have discussed above, the low-$k_\perp$ region corresponds to VMD and GVMD states that encompasses both perturbative high-$p_\perp$ and nonperturbative low-$p_\perp$ interactions. Above k_1, the region is split along the line $k_\perp = p_\perp$. When $p_\perp > k_\perp$ the photon is resolved by the hard interaction, as described by the anomalous part of the photon distribution function. This is as in the GVMD sector, except that we should (probably) not worry about multiple parton–parton interactions. In the complementary region $k_\perp > p_\perp$, the $p_\perp$ scale is just part of the traditional

evolution of the parton distributions of the proton up to the scale of $k_\perp$, and thus there is no need to introduce an internal structure of the photon. One could imagine the direct class of events as extending below k_1 and there being the low-$p_\perp$ part of the GVMD class, only appearing when a hard interaction at a larger $p_\perp$ scale would not preempt it. This possibility is implicit in the standard cross section framework.

In $\gamma\gamma$ physics [Sch94a, Sch97], the superposition in eq. (121) applies separately for each of the two incoming photons. In total there are therefore $3 \times 3 = 9$ combinations. However, trivial symmetry reduces this to six distinct classes, written in terms of the total cross section (cf. eq. (122)) as

$$\begin{aligned} \sigma_{\rm tot}^{\gamma\gamma}(s) &= \sigma_{\rm dir\times dir}^{\gamma\gamma}(s) + \sigma_{\rm VMD\times VMD}^{\gamma\gamma}(s) + \sigma_{\rm GVMD\times GVMD}^{\gamma\gamma}(s) \\ &+ 2\sigma_{\rm dir\times VMD}^{\gamma\gamma}(s) + 2\sigma_{\rm dir\times GVMD}^{\gamma\gamma}(s) + 2\sigma_{\rm VMD\times GVMD}^{\gamma\gamma}(s) \ . \end{aligned} \tag{125}$$

A parameterization of the total $\gamma\gamma$ cross section is found in [Sch94a, Sch97].

The six different kinds of $\gamma\gamma$ events are thus:

- The direct$\times$direct events, which correspond to the subprocess $\gamma\gamma \to q\overline{q}$ (or $\ell^+\ell^-$). The typical event structure is two high-$p_\perp$ jets and no beam remnants.
- The VMD$\times$VMD events, which have the same properties as the VMD γp events. There are four by four combinations of the two incoming vector mesons, with one VMD factor for each meson.
- The GVMD$\times$GVMD events, wherein each photon fluctuates into a $q\overline{q}$ pair of larger virtuality than in the VMD class. The 'anomalous' classification assumes that one parton of each pair gives a beam remnant, whereas the other (or a daughter parton thereof) participates in a high-$p_\perp$ scattering. The GVMD concept implies the presence also of low-$p_\perp$ events, like for VMD.
- The direct$\times$VMD events, which have the same properties as the direct γp events.
- The direct$\times$GVMD events, in which a bare photon interacts with a parton from the anomalous photon. The typical structure is then two high-$p_\perp$ jets and a beam remnant.
- The VMD$\times$GVMD events, which have the same properties as the GVMD γp events.

Like for photoproduction events, this can be illustrated in a parameter space, but now three-dimensional, with axes given by the $k_{\perp 1}$, $k_{\perp 2}$ and $p_\perp$ scales. Here each $k_{\perp i}$ is a measure of the virtuality of a fluctuation of a photon, and $p_\perp$ corresponds to the most virtual rung on the ladder between the two photons, possibly excepting the endpoint $k_{\perp i}$ ones. So, to first approximation, the coordinates along the $k_{\perp i}$ axes determine the characters of the interacting photons while $p_\perp$ determines the character of the interaction process. Double counting should be avoided by trying to impose a consistent classification. Thus, for instance, $p_\perp > k_{\perp i}$ with $k_{\perp 1} < k_0$ and $k_0 < k_{\perp 2} < k_1$ gives a hard interaction between a VMD and a GVMD photon, while $k_{\perp 1} > p_\perp > k_{\perp 2}$ with $k_{\perp 1} > k_1$ and $k_{\perp 2} < k_0$ is a single-resolved process (direct$\times$VMD; with $p_\perp$ now in the parton distribution evolution).

In much of the literature, where a coarser classification is used, our direct$\times$direct is called direct, our direct$\times$VMD and direct$\times$GVMD is called single-resolved since they both involve one resolved photon which gives a beam remnant, and the rest are called double-resolved since both photons are resolved and give beam remnants.

If the photon is virtual, it has a reduced probability to fluctuate into a vector meson state, and this state has a reduced interaction probability. This can be modelled by a traditional dipole factor $(m_V^2/(m_V^2+Q^2))^2$ for a photon of virtuality Q^2, where $m_V \to 2k_\perp$ for a GVMD state. Putting it all together, the cross section of the GVMD sector of photoproduction then scales like

$$\int_{k_0^2}^{k_1^2} \frac{{\rm d}k_\perp^2}{k_\perp^2} \frac{k_V^2}{k_\perp^2} \left(\frac{4k_\perp^2}{4k_\perp^2+Q^2} \right)^2 \ . \tag{126}$$

For a virtual photon the DIS process $\gamma^* q \to q$ is also possible, but by gauge invariance its cross section must vanish in the limit $Q^2 \to 0$. At large Q^2, the direct processes can be

considered as the $O(\alpha_s)$ correction to the lowest-order DIS process, but the direct ones survive for $Q^2 \to 0$. There is no unique prescription for a proper combination at all Q^2, but we have attempted an approach that gives the proper limits and minimizes double-counting. For large Q^2, the DIS $\gamma^* p$ cross section is proportional to the structure function $F_2(x,Q^2)$ with the Bjorken $x = Q^2/(Q^2+W^2)$. Since normal parton distribution parameterizations are frozen below some Q_0 scale and therefore do not obey the gauge invariance condition, an ad hoc factor $(Q^2/(Q^2+m_\rho^2))^2$ is introduced for the conversion from the parameterized $F_2(x,Q^2)$ to a $\sigma_{\rm DIS}^{\gamma^* p}$:

$$\sigma_{\rm DIS}^{\gamma^* p} \simeq \left(\frac{Q^2}{Q^2+m_\rho^2}\right)^2 \frac{4\pi^2\alpha_{\rm em}}{Q^2} F_2(x,Q^2) = \frac{4\pi^2\alpha_{\rm em}Q^2}{(Q^2+m_\rho^2)^2} \sum_q e_q^2 \left\{xq(x,Q^2)+x\overline{q}(x,Q^2)\right\} \ . \quad (127)$$

Here m_ρ is some nonperturbative hadronic mass parameter, for simplicity identified with the ρ mass. One of the $Q^2/(Q^2+m_\rho^2)$ factors is required already to give finite $\sigma_{\rm tot}^{\gamma p}$ for conventional parton distributions, and could be viewed as a screening of the individual partons at small Q^2. The second factor is chosen to give not only a finite but actually a vanishing $\sigma_{\rm DIS}^{\gamma^* p}$ for $Q^2 \to 0$ in order to retain the pure photoproduction description there. This latter factor thus is more a matter of convenience, and other approaches could have been pursued.

In order to avoid double-counting between DIS and direct events, a requirement $p_\perp > \max(k_1, Q)$ is imposed on direct events. In the remaining DIS ones, denoted lowest order (LO) DIS, thus $p_\perp < Q$. This would suggest a subdivision $\sigma_{\rm LO\,DIS}^{\gamma^* p} = \sigma_{\rm DIS}^{\gamma^* p} - \sigma_{\rm direct}^{\gamma^* p}$, with $\sigma_{\rm DIS}^{\gamma^* p}$ given by eq. (127) and $\sigma_{\rm direct}^{\gamma^* p}$ by the perturbative matrix elements. In the limit $Q^2 \to 0$, the DIS cross section is now constructed to vanish while the direct is not, so this would give $\sigma_{\rm LO\,DIS}^{\gamma^* p} < 0$. However, here we expect the correct answer not to be a negative number but an exponentially suppressed one, by a Sudakov form factor. This modifies the cross section:

$$\sigma_{\rm LO\,DIS}^{\gamma^* p} = \sigma_{\rm DIS}^{\gamma^* p} - \sigma_{\rm direct}^{\gamma^* p} \quad \longrightarrow \quad \sigma_{\rm DIS}^{\gamma^* p} \exp\left(-\frac{\sigma_{\rm direct}^{\gamma^* p}}{\sigma_{\rm DIS}^{\gamma^* p}}\right) \ . \quad (128)$$

Since we here are in a region where the DIS cross section is no longer the dominant one, this change of the total DIS cross section is not essential.

The overall picture, from a DIS perspective, now requires three scales to be kept track of. The traditional DIS region is the strongly ordered one, $Q^2 \gg k_\perp^2 \gg p_\perp^2$, where DGLAP-style evolution [Alt77, Gri72] is responsible for the event structure. As always, ideology wants strong ordering, while the actual classification is based on ordinary ordering $Q^2 > k_\perp^2 > p_\perp^2$. The region $k_\perp^2 > \max(Q^2, p_\perp^2)$ is also DIS, but of the $O(\alpha_s)$ direct kind. The region where $k_\perp$ is the smallest scale corresponds to non-ordered emissions, that then go beyond DGLAP validity, while the region $p_\perp^2 > k_\perp^2 > Q^2$ cover the interactions of a resolved virtual photon. Comparing with the plane of real photoproduction, we conclude that the whole region $p_\perp > k_\perp$ involves no double-counting, since we have made no attempt at a non-DGLAP DIS description but can choose to cover this region entirely by the VMD/GVMD descriptions. Actually, it is only in the corner $p_\perp < k_\perp < \min(k_1, Q)$ that an overlap can occur between the resolved and the DIS descriptions. Some further considerations show that usually either of the two is strongly suppressed in this region, except in the range of intermediate Q^2 and rather small W^2. Typically, this is the region where $x \approx Q^2/(Q^2+W^2)$ is not close to zero, and where F_2 is dominated by the valence-quark contribution. The latter behaves roughly $\propto (1-x)^n$, with an n of the order of 3 or 4. Therefore we will introduce a corresponding damping factor to the VMD/GVMD terms.

In total, we have now arrived at our ansatz for all Q^2:

$$\sigma_{\rm tot}^{\gamma^* p} = \sigma_{\rm DIS}^{\gamma^* p} \exp\left(-\frac{\sigma_{\rm direct}^{\gamma^* p}}{\sigma_{\rm DIS}^{\gamma^* p}}\right) + \sigma_{\rm direct}^{\gamma^* p} + \left(\frac{W^2}{Q^2+W^2}\right)^n \left(\sigma_{\rm VMD}^{\gamma^* p} + \sigma_{\rm GVMD}^{\gamma^* p}\right) \ , \quad (129)$$

with four main components. Most of these in their turn have a complicated internal structure, as we have seen.

Turning to $\gamma^*\gamma^*$ processes, finally, the parameter space is now five-dimensional: Q_1, Q_2, $k_{\perp 1}$, $k_{\perp 2}$ and $p_\perp$. As before, an effort is made to avoid double-counting, by having a unique classification of each region in the five-dimensional space. Remaining double-counting is dealt with as above. In total, our ansatz for $\gamma^*\gamma^*$ interactions at all Q^2 contains 13 components: 9 when two VMD, GVMD or direct photons interact, as is already allowed for real photons, plus a further 4 where a 'DIS photon' from either side interacts with a VMD or GVMD one. With the label resolved used to denote VMD and GVMD, one can write

$$\begin{aligned}
\sigma_{\rm tot}^{\gamma^*\gamma^*}(W^2,Q_1^2,Q_2^2) &= \sigma_{\rm DIS\times res}^{\gamma^*\gamma^*}\,\exp\left(-\frac{\sigma_{\rm dir\times res}^{\gamma^*\gamma^*}}{\sigma_{\rm DIS\times res}^{\gamma^*\gamma^*}}\right)+\sigma_{\rm dir\times res}^{\gamma^*\gamma^*} \\
&+ \sigma_{\rm res\times DIS}^{\gamma^*\gamma^*}\,\exp\left(-\frac{\sigma_{\rm res\times dir}^{\gamma^*\gamma^*}}{\sigma_{\rm res\times DIS}^{\gamma^*\gamma^*}}\right)+\sigma_{\rm res\times dir}^{\gamma^*\gamma^*} \\
&+ \sigma_{\rm dir\times dir}^{\gamma^*\gamma^*}+\left(\frac{W^2}{Q_1^2+Q_2^2+W^2}\right)^3\sigma_{\rm res\times res}^{\gamma^*\gamma^*}
\end{aligned} \tag{130}$$

Most of the 13 components in their turn have a complicated internal structure, as we have seen.

An important note is that the Q^2 dependence of the DIS and direct photon interactions is implemented in the matrix element expressions, i.e. in processes such as $\gamma^*\gamma^* \to q\overline{q}$ or $\gamma^* q \to qg$ the photon virtuality explicitly enters. This is different from VMD/GVMD, where dipole factors are used to reduce the total cross sections and the assumed flux of partons inside a virtual photon relative to those of a real one, but the matrix elements themselves contain no dependence on the virtuality either of the partons or of the photon itself. Typically results are obtained with the SaS 1D parton distributions for the virtual transverse photons [Sch95, Sch96], since these are well matched to our framework, e.g. allowing a separation of the VMD and GVMD/anomalous components. Parton distributions of virtual longitudinal photons are by default given by some Q^2-dependent factor times the transverse ones. The new set by Chýla [Chý00] allows more precise modelling here, but first indications are that many studies will not be sensitive to the detailed shape.

The photon physics machinery is of considerable complexity, and so the above is only a brief summary. Further details can be found in the literature quoted above. Some topics are also covered in other places in this manual, e.g. the flux of transverse and longitudinal photons in subsection 7.1.4, scale choices for parton density evaluation in subsection 7.2, and further aspects of the generation machinery and switches in subsection 8.3.

8 Physics Processes

In this section we enumerate the physics processes that are available in PYTHIA, introducing the ISUB code that can be used to select desired processes. A number of comments are made about the physics scenarios involved, in particular with respect to underlying assumptions and domain of validity. The section closes with a survey of interesting processes by machine.

8.1 The Process Classification Scheme

A wide selection of fundamental $2 \to 1$ and $2 \to 2$ tree processes of the Standard Model (electroweak and strong) has been included in PYTHIA, and slots are provided for many more, not yet implemented. In addition, 'minimum-bias'-type processes (like elastic scattering), loop graphs, box graphs, $2 \to 3$ tree graphs and many non-Standard Model processes are included. The classification is not always unique. A process that proceeds only via an *s*-channel state is classified as a $2 \to 1$ process (e.g. $q\overline{q} \to \gamma^*/Z^0 \to e^+e^-$), but a $2 \to 2$ cross section may well have contributions from *s*-channel diagrams ($gg \to gg$ obtains contributions from $gg \to g^* \to gg$). Also, in the program, $2 \to 1$ and $2 \to 2$ graphs may sometimes be convoluted with two $1 \to 2$ splittings to form effective $2 \to 3$ or $2 \to 4$ processes ($W^+W^- \to h^0$ is folded with $q \to q''W^+$ and $q' \to q'''W^-$ to give $qq' \to q''q'''h^0$).

The original classification and numbering scheme feels less relevant today than when originally conceived. In those days, the calculation of $2 \to 3$ or $2 \to 4$ matrix elements was sufficiently complicated that one would wish to avoid it if at all possible, e.g. by having in mind to define effective parton densities for all standard model particles, such as the $W^\pm$. Today, the improved computational techniques and increased computing power implies that people would be willing to include a branching $q \to qW$ as part of the hard process, i.e. not try to factor it off in some approximation. With the large top mass and large Higgs mass limits, there is also a natural subdivision, such that the b quark is the heaviest object for which the parton distribution concept makes sense. Therefore most of the prepared but empty slots are likely to remain empty, or be reclaimed for other processes.

It is possible to select a combination of subprocesses to simulate, and also afterwards to know which subprocess was actually selected in each event. For this purpose, all subprocesses are numbered according to an ISUB code. The list of possible codes is given in Tables 17, 18, 19, 20, 21, 22, 23 and 24. Only processes marked with a '+' sign in the first column have been implemented in the program to date. Although ISUB codes were originally designed in a logical fashion, we must admit that subsequent developments of the program have tended to obscure the structure. For instance, the process numbers for Higgs production are spread out, in part as a consequence of the original classification, in part because further production mechanisms have been added one at a time, in whatever free slots could be found. At some future date the subprocess list will therefore be re-organized. In the thematic descriptions that follow the main tables, the processes of interest are repeated in a more logical order. If you want to look for a specific process, it will be easier to find it there.

In the following, f_i represents a fundamental fermion of flavour *i*, i.e. d, u, s, c, b, t, b′, t′, e^-, ν_e, μ^-, ν_μ, τ^-, ν_τ, τ'^- or ν'_τ. A corresponding antifermion is denoted by $\overline{f}_i$. In several cases, some classes of fermions are explicitly excluded, since they do not couple to the g or γ (no $e^+e^- \to gg$, e.g.). When processes have only been included for quarks, while leptons might also have been possible, the notation q_i is used. A lepton is denoted by ℓ; in a few cases neutrinos are also lumped under this heading. In processes where fermion masses are explicitly included in the matrix elements, an F or Q is used to denote an arbitrary fermion or quark. Flavours appearing already in the initial state are denoted by indices *i* and *j*, whereas new flavours in the final state are denoted by *k* and *l*.

In supersymmetric processes, antiparticles of sfermions are denoted by *, i.e. $\tilde{t}^*$ rather than the more correct but cumbersome $\overline{\tilde{t}}$ or $\tilde{t}$.

Table 17: Subprocess codes, part 1. First column is '+' for processes implemented and blank for those that are only foreseen. Second is the subprocess number `ISUB`, and third the description of the process. The final column gives references from which the cross sections have been obtained. See text for further information.

In	No.	Subprocess	Reference
+	1	$f_i\bar{f}_i \to \gamma^*/Z^0$	[Eic84]
+	2	$f_i\bar{f}_j \to W^+$	[Eic84]
+	3	$f_i\bar{f}_i \to h^0$	[Eic84]
	4	$\gamma W^+ \to W^+$	
+	5	$Z^0Z^0 \to h^0$	[Eic84, Cha85]
	6	$Z^0W^+ \to W^+$	
	7	$W^+W^- \to Z^0$	
+	8	$W^+W^- \to h^0$	[Eic84, Cha85]
+	10	$f_if_j \to f_kf_l$ (QFD)	[Ing87a]
+	11	$f_if_j \to f_if_j$ (QCD)	[Com77, Ben84, Eic84, Chi90]
+	12	$f_i\bar{f}_i \to f_k\bar{f}_k$	[Com77, Ben84, Eic84, Chi90]
+	13	$f_i\bar{f}_i \to gg$	[Com77, Ben84]
+	14	$f_i\bar{f}_i \to g\gamma$	[Hal78, Ben84]
+	15	$f_i\bar{f}_i \to gZ^0$	[Eic84]
+	16	$f_i\bar{f}_j \to gW^+$	[Eic84]
	17	$f_i\bar{f}_i \to gh^0$	
+	18	$f_i\bar{f}_i \to \gamma\gamma$	[Ber84]
+	19	$f_i\bar{f}_i \to \gamma Z^0$	[Eic84]
+	20	$f_i\bar{f}_j \to \gamma W^+$	[Eic84, Sam91]
	21	$f_i\bar{f}_i \to \gamma h^0$	
+	22	$f_i\bar{f}_i \to Z^0Z^0$	[Eic84, Gun86]
+	23	$f_i\bar{f}_j \to Z^0W^+$	[Eic84, Gun86]
+	24	$f_i\bar{f}_i \to Z^0h^0$	[Ber85]
+	25	$f_i\bar{f}_i \to W^+W^-$	[Bar94, Gun86]
+	26	$f_i\bar{f}_j \to W^+h^0$	[Eic84]
	27	$f_i\bar{f}_i \to h^0h^0$	
+	28	$f_ig \to f_ig$	[Com77, Ben84]
+	29	$f_ig \to f_i\gamma$	[Hal78, Ben84]
+	30	$f_ig \to f_iZ^0$	[Eic84]
+	31	$f_ig \to f_kW^+$	[Eic84]
	32	$f_ig \to f_ih^0$	
+	33	$f_i\gamma \to f_ig$	[Duk82]
+	34	$f_i\gamma \to f_i\gamma$	[Duk82]
+	35	$f_i\gamma \to f_iZ^0$	[Gab86]
+	36	$f_i\gamma \to f_kW^+$	[Gab86]
	37	$f_i\gamma \to f_ih^0$	
	38	$f_iZ^0 \to f_ig$	
	39	$f_iZ^0 \to f_i\gamma$	

Table 18: Subprocess codes, part 2. Comments as before.

In	No.	Subprocess	Reference
	40	$f_iZ^0 \to f_iZ^0$	
	41	$f_iZ^0 \to f_kW^+$	
	42	$f_iZ^0 \to f_ih^0$	
	43	$f_iW^+ \to f_kg$	
	44	$f_iW^+ \to f_k\gamma$	
	45	$f_iW^+ \to f_kZ^0$	
	46	$f_iW^+ \to f_kW^+$	
	47	$f_iW^+ \to f_kh^0$	
	48	$f_ih^0 \to f_ig$	
	49	$f_ih^0 \to f_i\gamma$	
	50	$f_ih^0 \to f_iZ^0$	
	51	$f_ih^0 \to f_kW^+$	
	52	$f_ih^0 \to f_ih^0$	
+	53	$gg \to f_k\bar{f}_k$	[Com77, Ben84]
+	54	$g\gamma \to f_k\bar{f}_k$	[Duk82]
	55	$gZ^0 \to f_k\bar{f}_k$	
	56	$gW^+ \to f_k\bar{f}_l$	
	57	$gh^0 \to f_k\bar{f}_k$	
+	58	$\gamma\gamma \to f_k\bar{f}_k$	[Bar90]
	59	$\gamma Z^0 \to f_k\bar{f}_k$	
	60	$\gamma W^+ \to f_k\bar{f}_l$	
	61	$\gamma h^0 \to f_k\bar{f}_k$	
	62	$Z^0Z^0 \to f_k\bar{f}_k$	
	63	$Z^0W^+ \to f_k\bar{f}_l$	
	64	$Z^0h^0 \to f_k\bar{f}_k$	
	65	$W^+W^- \to f_k\bar{f}_k$	
	66	$W^+h^0 \to f_k\bar{f}_l$	
	67	$h^0h^0 \to f_k\bar{f}_k$	
+	68	$gg \to gg$	[Com77, Ben84]
+	69	$\gamma\gamma \to W^+W^-$	[Kat83]
+	70	$\gamma W^+ \to Z^0W^+$	[Kun87]
+	71	$Z^0Z^0 \to Z^0Z^0$ (longitudinal)	[Abb87]
+	72	$Z^0Z^0 \to W^+W^-$ (longitudinal)	[Abb87]
+	73	$Z^0W^+ \to Z^0W^+$ (longitudinal)	[Dob91]
	74	$Z^0h^0 \to Z^0h^0$	
	75	$W^+W^- \to \gamma\gamma$	
+	76	$W^+W^- \to Z^0Z^0$ (longitudinal)	[Ben87b]
+	77	$W^+W^\pm \to W^+W^\pm$ (longitudinal)	[Dun86, Bar90a]
	78	$W^+h^0 \to W^+h^0$	
	79	$h^0h^0 \to h^0h^0$	
+	80	$q_i\gamma \to q_k\pi^\pm$	[Bag82]

Table 19: Subprocess codes, part 3. Comments as before

In	No.	Subprocess	Reference
+	81	$f_i\bar{f}_i \to Q_k\overline{Q}_k$	[Com79]
+	82	$gg \to Q_k\overline{Q}_k$	[Com79]
+	83	$q_i f_j \to Q_k f_l$	[Dic86]
+	84	$g\gamma \to Q_k\overline{Q}_k$	[Fon81]
+	85	$\gamma\gamma \to F_k\overline{F}_k$	[Bar90]
+	86	$gg \to J/\psi g$	[Bai83]
+	87	$gg \to \chi_{0c} g$	[Gas87]
+	88	$gg \to \chi_{1c} g$	[Gas87]
+	89	$gg \to \chi_{2c} g$	[Gas87]
+	91	elastic scattering	[Sch94]
+	92	single diffraction ($AB \to XB$)	[Sch94]
+	93	single diffraction ($AB \to AX$)	[Sch94]
+	94	double diffraction	[Sch94]
+	95	low-$p_\perp$ production	[Sjö87a]
+	96	semihard QCD $2 \to 2$	[Sjö87a]
+	99	$\gamma^* q \to q$	
	101	$gg \to Z^0$	
+	102	$gg \to h^0$	[Eic84]
+	103	$\gamma\gamma \to h^0$	[Dre89]
+	104	$gg \to \chi_{0c}$	[Bai83]
+	105	$gg \to \chi_{2c}$	[Bai83]
+	106	$gg \to J/\psi\gamma$	[Dre91]
+	107	$g\gamma \to J/\psi g$	[Ber81]
+	108	$\gamma\gamma \to J/\psi\gamma$	[Jun97]
+	110	$f_i\bar{f}_i \to \gamma h^0$	[Ber85a]
+	111	$f_i\bar{f}_i \to g h^0$	[Ell88]
+	112	$f_i g \to f_i h^0$	[Ell88]
+	113	$gg \to g h^0$	[Ell88]
+	114	$gg \to \gamma\gamma$	[Con71, Ber84, Dic88]
+	115	$gg \to g\gamma$	[Con71, Ber84, Dic88]
	116	$gg \to \gamma Z^0$	
	117	$gg \to Z^0 Z^0$	
	118	$gg \to W^+W^-$	
	119	$\gamma\gamma \to gg$	
+	121	$gg \to Q_k\overline{Q}_k h^0$	[Kun84]
+	122	$q_i\bar{q}_i \to Q_k\overline{Q}_k h^0$	[Kun84]
+	123	$f_i f_j \to f_i f_j h^0$ (ZZ fusion)	[Cah84]
+	124	$f_i f_j \to f_k f_l h^0$ (W^+W^- fusion)	[Cah84]
+	131	$f_i\gamma^*_T \to f_i g$	[Alt78]
+	132	$f_i\gamma^*_L \to f_i g$	[Alt78]
+	133	$f_i\gamma^*_T \to f_i\gamma$	[Alt78]

Table 20: Subprocess codes, part 4. Comments as before.

In	No.	Subprocess	Reference
+	134	$f_i\gamma^*_L \to f_i\gamma$	[Alt78]
+	135	$g\gamma^*_T \to f_i\bar{f}_i$	[Alt78]
+	136	$g\gamma^*_L \to f_i\bar{f}_i$	[Alt78]
+	137	$\gamma^*_T\gamma^*_T \to f_i\bar{f}_i$	[Bai81]
+	138	$\gamma^*_T\gamma^*_L \to f_i\bar{f}_i$	[Bai81]
+	139	$\gamma^*_L\gamma^*_T \to f_i\bar{f}_i$	[Bai81]
+	140	$\gamma^*_L\gamma^*_L \to f_i\bar{f}_i$	[Bai81]
+	141	$f_i\bar{f}_i \to \gamma/Z^0/Z'^0$	[Alt89]
+	142	$f_i\bar{f}_j \to W'^+$	[Alt89]
+	143	$f_i\bar{f}_j \to H^+$	[Gun87]
+	144	$f_i\bar{f}_j \to R$	[Ben85a]
+	145	$q_i\ell_j \to L_Q$	[Wud86]
+	146	$e\gamma \to e^*$	[Bau90]
+	147	$dg \to d^*$	[Bau90]
+	148	$ug \to u^*$	[Bau90]
+	149	$gg \to \eta_{tc}$	[Eic84, App92]
+	151	$f_i\bar{f}_i \to H^0$	[Eic84]
+	152	$gg \to H^0$	[Eic84]
+	153	$\gamma\gamma \to H^0$	[Dre89]
+	156	$f_i\bar{f}_i \to A^0$	[Eic84]
+	157	$gg \to A^0$	[Eic84]
+	158	$\gamma\gamma \to A^0$	[Dre89]
+	161	$f_ig \to f_kH^+$	[Bar88]
+	162	$q_ig \to \ell_kL_Q$	[Hew88]
+	163	$gg \to L_Q\overline{L}_Q$	[Hew88, Eic84]
+	164	$q_i\overline{q}_i \to L_Q\overline{L}_Q$	[Hew88]
+	165	$f_i\bar{f}_i \to f_k\bar{f}_k$ (via γ^*/Z^0)	[Eic84, Lan91]
+	166	$f_i\bar{f}_j \to f_k\bar{f}_l$ (via $W^\pm$)	[Eic84, Lan91]
+	167	$q_iq_j \to q_kd^*$	[Bau90]
+	168	$q_iq_j \to q_ku^*$	[Bau90]
+	169	$q_i\overline{q}_i \to e^\pm e^{*\mp}$	[Bau90]
+	171	$f_i\bar{f}_i \to Z^0H^0$	[Eic84]
+	172	$f_i\bar{f}_j \to W^+H^0$	[Eic84]
+	173	$f_if_j \to f_if_jH^0$ (ZZ fusion)	[Cah84]
+	174	$f_if_j \to f_kf_lH^0$ (W^+W^- fusion)	[Cah84]
+	176	$f_i\bar{f}_i \to Z^0A^0$	[Eic84]
+	177	$f_i\bar{f}_j \to W^+A^0$	[Eic84]
+	178	$f_if_j \to f_if_jA^0$ (ZZ fusion)	[Cah84]
+	179	$f_if_j \to f_kf_lA^0$ (W^+W^- fusion)	[Cah84]
+	181	$gg \to Q_k\overline{Q}_kH^0$	[Kun84]
+	182	$q_i\overline{q}_i \to Q_k\overline{Q}_kH^0$	[Kun84]

Table 21: Subprocess codes, part 5. Comments as before.

In	No.	Subprocess	Reference
+	183	$f_i\bar{f}_i \to gH^0$	[Ell88]
+	184	$f_i g \to f_i H^0$	[Ell88]
+	185	$gg \to gH^0$	[Ell88]
+	186	$gg \to Q_k\overline{Q}_k A^0$	[Kun84]
+	187	$q_i\bar{q}_i \to Q_k\overline{Q}_k A^0$	[Kun84]
+	188	$f_i\bar{f}_i \to gA^0$	[Ell88]
+	189	$f_i g \to f_i A^0$	[Ell88]
+	190	$gg \to gA^0$	[Ell88]
+	191	$f_i\bar{f}_i \to \rho^0_{tc}$	[Eic96]
+	192	$f_i\bar{f}_j \to \rho^\pm_{tc}$	[Eic96]
+	193	$f_i\bar{f}_i \to \omega^0_{tc}$	[Eic96]
+	194	$f_i\bar{f}_i \to f_k\bar{f}_k$	[Eic96, Lan99]
+	195	$f_i\bar{f}_j \to f_k\bar{f}_l$	[Eic96, Lan99]
+	201	$f_i\bar{f}_i \to \tilde{e}_L\tilde{e}^*_L$	[Bar87, Daw85]
+	202	$f_i\bar{f}_i \to \tilde{e}_R\tilde{e}^*_R$	[Bar87, Daw85]
+	203	$f_i\bar{f}_i \to \tilde{e}_L\tilde{e}^*_R + \tilde{e}^*_L\tilde{e}_R$	[Bar87]
+	204	$f_i\bar{f}_i \to \tilde{\mu}_L\tilde{\mu}^*_L$	[Bar87, Daw85]
+	205	$f_i\bar{f}_i \to \tilde{\mu}_R\tilde{\mu}^*_R$	[Bar87, Daw85]
+	206	$f_i\bar{f}_i \to \tilde{\mu}_L\tilde{\mu}^*_R + \tilde{\mu}^*_L\tilde{\mu}_R$	[Bar87]
+	207	$f_i\bar{f}_i \to \tilde{\tau}_1\tilde{\tau}^*_1$	[Bar87, Daw85]
+	208	$f_i\bar{f}_i \to \tilde{\tau}_2\tilde{\tau}^*_2$	[Bar87, Daw85]
+	209	$f_i\bar{f}_i \to \tilde{\tau}_1\tilde{\tau}^*_2 + \tilde{\tau}^*_1\tilde{\tau}_2$	[Bar87]
+	210	$f_i\bar{f}_j \to \tilde{\ell}_L\tilde{\nu}^*_\ell + \tilde{\ell}^*_L\tilde{\nu}_\ell$	[Daw85]
+	211	$f_i\bar{f}_j \to \tilde{\tau}_1\tilde{\nu}^*_\tau + \tilde{\tau}^*_1\tilde{\nu}_\tau$	[Daw85]
+	212	$f_i\bar{f}_j \to \tilde{\tau}_2\tilde{\nu}_\tau{}^* + \tilde{\tau}^*_2\tilde{\nu}_\tau$	[Daw85]
+	213	$f_i\bar{f}_i \to \tilde{\nu}_\ell\tilde{\nu}^*_\ell$	[Bar87, Daw85]
+	214	$f_i\bar{f}_i \to \tilde{\nu}_\tau\tilde{\nu}^*_\tau$	[Bar87, Daw85]
+	216	$f_i\bar{f}_i \to \tilde{\chi}_1\tilde{\chi}_1$	[Bar86a]
+	217	$f_i\bar{f}_i \to \tilde{\chi}_2\tilde{\chi}_2$	[Bar86a]
+	218	$f_i\bar{f}_i \to \tilde{\chi}_3\tilde{\chi}_3$	[Bar86a]
+	219	$f_i\bar{f}_i \to \tilde{\chi}_4\tilde{\chi}_4$	[Bar86a]
+	220	$f_i\bar{f}_i \to \tilde{\chi}_1\tilde{\chi}_2$	[Bar86a]
+	221	$f_i\bar{f}_i \to \tilde{\chi}_1\tilde{\chi}_3$	[Bar86a]
+	222	$f_i\bar{f}_i \to \tilde{\chi}_1\tilde{\chi}_4$	[Bar86a]
+	223	$f_i\bar{f}_i \to \tilde{\chi}_2\tilde{\chi}_3$	[Bar86a]
+	224	$f_i\bar{f}_i \to \tilde{\chi}_2\tilde{\chi}_4$	[Bar86a]
+	225	$f_i\bar{f}_i \to \tilde{\chi}_3\tilde{\chi}_4$	[Bar86a]
+	226	$f_i\bar{f}_i \to \tilde{\chi}^\pm_1\tilde{\chi}^\mp_1$	[Bar86b]
+	227	$f_i\bar{f}_i \to \tilde{\chi}^\pm_2\tilde{\chi}^\mp_2$	[Bar86b]
+	228	$f_i\bar{f}_i \to \tilde{\chi}^\pm_1\tilde{\chi}^\mp_2$	[Bar86b]

Table 22: Subprocess codes, part 6. Comments as before.

In	No.	Subprocess	Reference
+	229	$f_i\bar{f}_j \to \tilde{\chi}_1\tilde{\chi}_1^{\pm}$	[Bar86a, Bar86b]
+	230	$f_i\bar{f}_j \to \tilde{\chi}_2\tilde{\chi}_1^{\pm}$	[Bar86a, Bar86b]
+	231	$f_i\bar{f}_j \to \tilde{\chi}_3\tilde{\chi}_1^{\pm}$	[Bar86a, Bar86b]
+	232	$f_i\bar{f}_j \to \tilde{\chi}_4\tilde{\chi}_1^{\pm}$	[Bar86a, Bar86b]
+	233	$f_i\bar{f}_j \to \tilde{\chi}_1\tilde{\chi}_2^{\pm}$	[Bar86a, Bar86b]
+	234	$f_i\bar{f}_j \to \tilde{\chi}_2\tilde{\chi}_2^{\pm}$	[Bar86a, Bar86b]
+	235	$f_i\bar{f}_j \to \tilde{\chi}_3\tilde{\chi}_2^{\pm}$	[Bar86a, Bar86b]
+	236	$f_i\bar{f}_j \to \tilde{\chi}_4\tilde{\chi}_2^{\pm}$	[Bar86a, Bar86b]
+	237	$f_i\bar{f}_i \to \tilde{g}\tilde{\chi}_1$	[Daw85]
+	238	$f_i\bar{f}_i \to \tilde{g}\tilde{\chi}_2$	[Daw85]
+	239	$f_i\bar{f}_i \to \tilde{g}\tilde{\chi}_3$	[Daw85]
+	240	$f_i\bar{f}_i \to \tilde{g}\tilde{\chi}_4$	[Daw85]
+	241	$f_i\bar{f}_j \to \tilde{g}\tilde{\chi}_1^{\pm}$	[Daw85]
+	242	$f_i\bar{f}_j \to \tilde{g}\tilde{\chi}_2^{\pm}$	[Daw85]
+	243	$f_i\bar{f}_i \to \tilde{g}\tilde{g}$	[Daw85]
+	244	$gg \to \tilde{g}\tilde{g}$	[Daw85]
+	246	$f_i g \to \tilde{q}_{iL}\tilde{\chi}_1$	[Daw85]
+	247	$f_i g \to \tilde{q}_{iR}\tilde{\chi}_1$	[Daw85]
+	248	$f_i g \to \tilde{q}_{iL}\tilde{\chi}_2$	[Daw85]
+	249	$f_i g \to \tilde{q}_{iR}\tilde{\chi}_2$	[Daw85]
+	250	$f_i g \to \tilde{q}_{iL}\tilde{\chi}_3$	[Daw85]
+	251	$f_i g \to \tilde{q}_{iR}\tilde{\chi}_3$	[Daw85]
+	252	$f_i g \to \tilde{q}_{iL}\tilde{\chi}_4$	[Daw85]
+	253	$f_i g \to \tilde{q}_{iR}\tilde{\chi}_4$	[Daw85]
+	254	$f_i g \to \tilde{q}_{jL}\tilde{\chi}_1^{\pm}$	[Daw85]
+	256	$f_i g \to \tilde{q}_{jL}\tilde{\chi}_2^{\pm}$	[Daw85]
+	258	$f_i g \to \tilde{q}_{iL}\tilde{g}$	[Daw85]
+	259	$f_i g \to \tilde{q}_{iR}\tilde{g}$	[Daw85]
+	261	$f_i\bar{f}_i \to \tilde{t}_1\tilde{t}_1^*$	[Daw85]
+	262	$f_i\bar{f}_i \to \tilde{t}_2\tilde{t}_2^*$	[Daw85]
+	263	$f_i\bar{f}_i \to \tilde{t}_1\tilde{t}_2^* + \tilde{t}_1^*\tilde{t}_2$	[Daw85]
+	264	$gg \to \tilde{t}_1\tilde{t}_1^*$	[Daw85]
+	265	$gg \to \tilde{t}_2\tilde{t}_2^*$	[Daw85]
+	271	$f_i f_j \to \tilde{q}_{iL}\tilde{q}_{jL}$	[Daw85]
+	272	$f_i f_j \to \tilde{q}_{iR}\tilde{q}_{jR}$	[Daw85]
+	273	$f_i f_j \to \tilde{q}_{iL}\tilde{q}_{jR} + \tilde{q}_{iR}\tilde{q}_{jL}$	[Daw85]
+	274	$f_i\bar{f}_j \to \tilde{q}_{iL}\tilde{q}_{jL}^*$	[Daw85]
+	275	$f_i\bar{f}_j \to \tilde{q}_{iR}\tilde{q}_{jR}^*$	[Daw85]
+	276	$f_i\bar{f}_j \to \tilde{q}_{iL}\tilde{q}_{jR}^* + \tilde{q}_{iR}\tilde{q}_{jL}^*$	[Daw85]
+	277	$f_i\bar{f}_i \to \tilde{q}_{jL}\tilde{q}_{jL}^*$	[Daw85]

Table 23: Subprocess codes, part 7. Comments as before.

In	No.	Subprocess	Reference
+	278	$f_i\bar{f}_i \to \tilde{q}_{jR}\tilde{q}^*_{jR}$	[Daw85]
+	279	$gg \to \tilde{q}_{iL}\tilde{q}^*_{iL}$	[Daw85]
+	280	$gg \to \tilde{q}_{iR}\tilde{q}^*_{iR}$	[Daw85]
+	281	$bq \to \tilde{b}_1\tilde{q}_L$ (q not b)	
+	282	$bq \to \tilde{b}_2\tilde{q}_R$	
+	283	$bq \to \tilde{b}_1\tilde{q}_R + \tilde{b}2\tilde{q}_L$	
+	284	$b\bar{q} \to \tilde{b}_1\tilde{q}^*_L$	
+	285	$b\bar{q} \to \tilde{b}_2\tilde{q}^*_R$	
+	286	$b\bar{q} \to \tilde{b}_1\tilde{q}^*_R + \tilde{b}_2\tilde{q}^*_L$	
+	287	$q\bar{q} \to \tilde{b}_1\tilde{b}^*_1$	
+	288	$q\bar{q} \to \tilde{b}_2\tilde{b}^*_2$	
+	289	$gg \to \tilde{b}_1\tilde{b}^*_1$	
+	290	$gg \to \tilde{b}_2\tilde{b}^*_2$	
+	291	$bb \to \tilde{b}_1\tilde{b}_1$	
+	292	$bb \to \tilde{b}_2\tilde{b}_2$	
+	293	$bb \to \tilde{b}_1\tilde{b}_2$	
+	294	$bg \to \tilde{b}_1\tilde{g}$	
+	295	$bg \to \tilde{b}_2\tilde{g}$	
+	296	$b\bar{b} \to \tilde{b}_1\tilde{b}^*_2 + \tilde{b}^*_1\tilde{b}_2$	
+	297	$f_i\bar{f}_j \to H^\pm h^0$	
+	298	$f_i\bar{f}_j \to H^\pm H^0$	
+	299	$f_i\bar{f}_i \to Ah^0$	
+	300	$f_i\bar{f}_i \to AH^0$	
+	301	$f_i\bar{f}_i \to H^+H^-$	
+	341	$\ell_i\ell_j \to H^{\pm\pm}_L$	[Hui97]
+	342	$\ell_i\ell_j \to H^{\pm\pm}_R$	[Hui97]
+	343	$\ell_i\gamma \to H^{\pm\pm}_L e^\mp$	[Hui97]
+	344	$\ell_i\gamma \to H^{\pm\pm}_R e^\mp$	[Hui97]
+	345	$\ell_i\gamma \to H^{\pm\pm}_L \mu^\mp$	[Hui97]
+	346	$\ell_i\gamma \to H^{\pm\pm}_R \mu^\mp$	[Hui97]
+	347	$\ell_i\gamma \to H^{\pm\pm}_L \tau^\mp$	[Hui97]
+	348	$\ell_i\gamma \to H^{\pm\pm}_R \tau^\mp$	[Hui97]
+	349	$f_i\bar{f}_i \to H^{++}_L H^{--}_L$	[Hui97]
+	350	$f_i\bar{f}_i \to H^{++}_R H^{--}_R$	[Hui97]
+	351	$f_i f_j \to f_k f_l H^{\pm\pm}_L$ (WW) fusion)	[Hui97]
+	352	$f_i f_j \to f_k f_l H^{\pm\pm}_R$ (WW) fusion)	[Hui97]
+	353	$f_i\bar{f}_i \to Z^0_R$	[Eic84]
+	354	$f_i\bar{f}_j \to W^+_R$	[Eic84]
+	361	$f_i\bar{f}_i \to W^+_L W^-_L$	[Lan99]
+	362	$f_i\bar{f}_i \to W^\pm_L \pi^\mp_{tc}$	[Lan99]

Table 24: Subprocess codes, part 8. Comments as before.

In	No.	Subprocess	Reference
+	363	$f_i\bar{f}_i \to \pi^+_{tc}\pi^-_{tc}$	[Lan99]
+	364	$f_i\bar{f}_i \to \gamma\pi^0_{tc}$	[Lan99]
+	365	$f_i\bar{f}_i \to \gamma\pi'^0_{tc}$	[Lan99]
+	366	$f_i\bar{f}_i \to Z^0\pi^0_{tc}$	[Lan99]
+	367	$f_i\bar{f}_i \to Z^0\pi'^0_{tc}$	[Lan99]
+	368	$f_i\bar{f}_i \to W^\pm\pi^\mp_{tc}$	[Lan99]
+	370	$f_i\bar{f}_j \to W^\pm_L Z^0_L$	[Lan99]
+	371	$f_i\bar{f}_j \to W^\pm_L\pi^0_{tc}$	[Lan99]
+	372	$f_i\bar{f}_j \to \pi^\pm_{tc}Z^0_L$	[Lan99]
+	373	$f_i\bar{f}_j \to \pi^\pm_{tc}\pi^0_{tc}$	[Lan99]
+	374	$f_i\bar{f}_j \to \gamma\pi^\pm_{tc}$	[Lan99]
+	375	$f_i\bar{f}_j \to Z^0\pi^\pm_{tc}$	[Lan99]
+	376	$f_i\bar{f}_j \to W^\pm\pi^0_{tc}$	[Lan99]
+	377	$f_i\bar{f}_j \to W^\pm\pi'^0_{tc}$	[Lan99]
+	391	$f\bar{f} \to G^*$	[Ran99]
+	392	$gg \to G^*$	[Ran99]
+	393	$q\bar{q} \to gG^*$	[Ran99, Bij01]
+	394	$qg \to qG^*$	[Ran99, Bij01]
+	395	$gg \to gG^*$	[Ran99, Bij01]

Charge-conjugate channels are always assumed included as well (where separate), and processes involving a W^+ also imply those involving a W^-. Wherever Z^0 is written, it is understood that γ^* and γ^*/Z^0 interference should be included as well (with possibilities to switch off either, if so desired). In some cases this is not fully implemented, see further below. Correspondingly, Z'^0 denotes the complete set $\gamma^*/Z^0/Z'^0$ (or some subset of it). Thus the notation γ is only used for a photon on the mass shell.

In the last column of the tables below, references are given to works from which formulae have been taken. Sometimes these references are to the original works on the subject, sometimes only to the place where the formulae are given in the most convenient or accessible form, or where chance lead us. Apologies to all matrix-element calculators who are not mentioned. However, remember that this is not a review article on physics processes, but only a way for readers to know what is actually found in the program, for better or worse. In several instances, errata have been obtained from the authors. Often the formulae given in the literature have been generalized to include trivial radiative corrections, Breit–Wigner line shapes with $\hat{s}$-dependent widths (see section 7.3), etc.

The following sections contain some useful comments on the processes included in the program, grouped by physics interest rather than sequentially by `ISUB` or `MSEL` code (see 9.2 for further information on the `MSEL` code). The different `ISUB` and `MSEL` codes that can be used to simulate the different groups are given. `ISUB` codes within brackets indicate the kind of processes that indirectly involve the given physics topic, although only as part of a larger whole. Some obvious examples, such as the possibility to produce jets in just about any process, are not spelled out in detail.

The text at times contains information on which special switches or parameters are of particular interest to a given process. All these switches are described in detail in sections 9.3 9.4 and

9.5, but are alluded to here so as to provide a more complete picture of the possibilities available for the different subprocesses. However, the list of possibilities is certainly not exhausted by the text below.

8.2 QCD Processes

Obviously most processes in PYTHIA contain QCD physics one way or another, so the above title should not be overstressed. One example: a process like $e^+e^- \to \gamma^*/Z^0 \to q\overline{q}$ is also traditionally called a QCD event, but is here book-kept as γ^*/Z^0 production. In this section we discuss scatterings between coloured partons, plus a few processes that are close relatives to other processes of this kind.

8.2.1 QCD jets

`MSEL` = 1, 2

`ISUB` =
- 11 $q_i q_j \to q_i q_j$
- 12 $q_i \overline{q}_i \to q_k \overline{q}_k$
- 13 $q_i \overline{q}_i \to gg$
- 28 $q_i g \to q_i g$
- 53 $gg \to q_k \overline{q}_k$
- 68 $gg \to gg$
- 96 semihard QCD $2 \to 2$

No higher-order processes are explicitly included, nor any higher-order loop corrections to the $2 \to 2$ processes. However, by initial- and final-state QCD radiation, multijet events are being generated, starting from the above processes. The shower rate of multijet production is clearly uncertain by some amount, especially for well-separated jets.

A string-based fragmentation scheme such as the Lund model needs cross sections for the different colour flows; these have been calculated in [Ben84] and differ from the usual calculations by interference terms of the order $1/N_C^2$. By default, the standard colour-summed QCD expressions for the differential cross sections are used. In this case, the interference terms are distributed among the various colour flows according to the pole structure of the terms. However, the interference terms can be excluded, by changing `MSTP(34)`

As an example, consider subprocess 28, $qg \to qg$. The total cross section for this process, obtained by summing and squaring the Feynman $\hat{s}$-, $\hat{t}$-, and $\hat{u}$-channel graphs, is [Com77]

$$2\left(1-\frac{\hat{u}\hat{s}}{\hat{t}^2}\right)-\frac{4}{9}\left(\frac{\hat{s}}{\hat{u}}+\frac{\hat{u}}{\hat{s}}\right)-1\,. \tag{131}$$

(An overall factor $\pi\alpha_s^2/\hat{s}^2$ is ignored.) Using the identity of the Mandelstam variables for the massless case, $\hat{s}+\hat{t}+\hat{u}=0$, this can be rewritten as

$$\frac{\hat{s}^2+\hat{u}^2}{\hat{t}^2}-\frac{4}{9}\left(\frac{\hat{s}}{\hat{u}}+\frac{\hat{u}}{\hat{s}}\right)\,. \tag{132}$$

On the other hand, the cross sections for the two possible colour flows of this subprocess are [Ben84]

$$\begin{aligned} A: &\quad \frac{4}{9}\left(2\frac{\hat{u}^2}{\hat{t}^2}-\frac{\hat{u}}{\hat{s}}\right)\,; \\ B: &\quad \frac{4}{9}\left(2\frac{\hat{s}^2}{\hat{t}^2}-\frac{\hat{s}}{\hat{u}}\right)\,. \end{aligned} \tag{133}$$

Colour configuration A is one in which the original colour of the q annihilates with the anticolour of the g, the g colour flows through, and a new colour–anticolour is created between the final q and g. In colour configuration B, the gluon anticolour flows through, but the q and g colours are interchanged. Note that these two colour configurations have different kinematics dependence. For `MSTP(34)=0`, these are the cross sections actually used.

The sum of the A and B contributions is

$$\frac{8}{9}\frac{\hat{s}^2+\hat{u}^2}{\hat{t}^2}-\frac{4}{9}\left(\frac{\hat{s}}{\hat{u}}+\frac{\hat{u}}{\hat{s}}\right) \; . \tag{134}$$

The difference between this expression and that of [Com77], corresponding to the interference between the two colour-flow configurations, is then

$$\frac{1}{9}\frac{\hat{s}^2+\hat{u}^2}{\hat{t}^2} \; , \tag{135}$$

which can be naturally divided between colour flows A and B:

$$\begin{aligned} A: &\quad \frac{1}{9}\frac{\hat{u}^2}{\hat{t}^2} \; ; \\ B: &\quad \frac{1}{9}\frac{\hat{s}^2}{\hat{t}^2} \; . \end{aligned} \tag{136}$$

For `MSTP(34)=1`, the standard QCD matrix element is therefore used, with the same relative importance of the two colour configurations as above. Similar procedures are followed also for the other QCD subprocesses.

All the matrix elements in this group are for massless quarks (although final-state quarks are of course put on the mass shell). As a consequence, cross sections are divergent for $p_\perp \to 0$, and some kind of regularization is required. Normally you are expected to set the desired $p_{\perp\mathrm{min}}$ value in `CKIN(3)`.

The new flavour produced in the annihilation processes (`ISUB` = 12 and 53) is determined by the flavours allowed for gluon splitting into quark–antiquark; see switch `MDME`.

Subprocess 96 is special among all the ones in the program. In terms of the basic cross section, it is equivalent to the sum of the other ones, i.e. 11, 12, 13, 28, 53 and 68. The phase space is mapped differently, however, and allows $p_\perp$ as input variable. This is especially useful in the context of the multiple interactions machinery, see subsection 11.2, where potential scatterings are considered in order of decreasing $p_\perp$, with a form factor related to the probability of not having another scattering with a $p_\perp$ larger than the considered one. You are not expected to access process 96 yourself. Instead it is automatically initialized and used either if process 95 is included or if multiple interactions are switched on. The process will then appear in the maximization information output, but not in the cross section table at the end of a run. Instead, the hardest scattering generated within the context of process 95 is reclassified as an event of the 11, 12, 13, 28, 53 or 68 kinds, based on the relative cross section for these in the point chosen. Further multiple interactions, subsequent to the hardest one, also do not show up in cross section tables.

8.2.2 Heavy flavours

`MSEL` = 4, 5, 6, 7, 8

`ISUB` =	81	$\mathrm{q}_i\overline{\mathrm{q}}_i \to \mathrm{Q}_k\overline{\mathrm{Q}}_k$
	82	$\mathrm{gg} \to \mathrm{Q}_k\overline{\mathrm{Q}}_k$
	(83)	$\mathrm{q}_i\mathrm{f}_j \to Q_k f_l$
	(84)	$\mathrm{g}\gamma \to \mathrm{Q}_k\overline{\mathrm{Q}}_k$
	(85)	$\gamma\gamma \to \mathrm{F}_k\overline{\mathrm{F}}_k$

The matrix elements in this group differ from the corresponding ones in the group above in that they correctly take into account the quark masses. As a consequence, the cross sections are finite for $p_\perp \to 0$. It is therefore not necessary to introduce any special cuts.

The two first processes that appear here are the dominant lowest-order QCD graphs in hadron colliders — a few other graphs will be mentioned later, such as process 83.

The choice of flavour to produce is according to a hierarchy of options:

1. if `MSEL=4-8` then the flavour is set by the `MSEL` value;
2. else if `MSTP(7)=1-8` then the flavour is set by the `MSTP(7)` value;
3. else the flavour is determined by the heaviest flavour allowed for gluon splitting into quark–antiquark; see switch `MDME`.

Note that only one heavy flavour is allowed at a time; if more than one is turned on in `MDME`, only the heaviest will be produced (as opposed to the case for `ISUB` = 12 and 53 above, where more than one flavour is allowed simultaneously).

The lowest-order processes listed above just represent one source of heavy-flavour production. Heavy quarks can also be present in the parton distributions at the Q^2 scale of the hard interaction, leading to processes like $\mathrm{Qg} \to \mathrm{Qg}$, so-called flavour excitation, or they can be created by gluon splittings $\mathrm{g} \to \mathrm{Q\overline{Q}}$ in initial- or final-state shower evolution. The implementation and importance of these various production mechanisms is discussed in detail in [Nor98]. In fact, as the c.m. energy is increased, these other processes gain in importance relative to the lowest-order production graphs above. As as example, only 10%–20% of the b production at LHC energies come from the lowest-order graphs. The figure is even smaller for charm, while it is well above 50% for top. At LHC energies, the specialized treatment described in this section is therefore only of interest for top (and potential fourth-generation quarks) — the higher-order corrections can here be approximated by an effective K factor, except possibly in some rare corners of phase space.

For charm and bottom, on the other hand, it is necessary to simulate the full event sample (within the desired kinematics cuts), and then only keep those events that contain b/c, be that either from lowest-order production, or flavour excitation, or gluon splitting. Obviously this may be a time-consuming enterprise — although the probability for a high-$p_\perp$ event at collider energies to contain (at least) one charm or bottom pair is fairly large, most of these heavy flavours are carrying a small fraction of the total $p_\perp$ flow of the jets, and therefore do not survive normal experimental cuts. We note that the lowest-order production of charm or bottom with processes 12 and 53, as part of the standard QCD mix, is now basically equivalent with that offered by processes 81 and 82. For 12 vs. 81 this is rather trivial, since only s-channel gluon exchange is involved, but for process 53 it requires a separate evaluation of massive matrix elements for c and b in the flavour loop. This is performed by retaining the $\hat{s}$ and $\hat{\theta}$ values already preliminarily selected for the massless kinematics, and recalculating $\hat{t}$ and $\hat{u}$ with mass effects included. Some of the documentation information in `PARI` does not properly reflect this recalculation, but that is purely a documentation issue. Also process 96, used internally for the total QCD jet cross section, includes c and b masses. Only the hardest interaction in a multiple interactions scenario may contain c/b, however, for technical reasons, so that the total rate may be underestimated. (Quite apart from other uncertainties, of course.)

As an aside, it is not only for the lowest-order graphs that events may be generated with a guaranteed heavy-flavour content. One may also generate the flavour excitation process by itself, in the massless approximation, using `ISUB` = 28 and setting the `KFIN` array appropriately. No trick exists to force the gluon splittings without introducing undesirable biases, however. In order to have it all, one therefore has to make a full QCD jets run, as already noted.

Also other processes can generate heavy flavours, all the way up to top, but then without a proper account of masses. By default, top production is switched off in those processes where a new flavour pair is produced at a gluon or photon vertex, i.e. 12, 53, 54, 58, 96 and 135–140, while charm and bottom is allowed. These defaults can be changed by setting the `MDME(IDC,1)` values of the appropriate g or γ 'decay channels', see further below.

The cross section for a heavy quark pair close to threshold can be modified according to the

formulae of [Fad90], see MSTP(35). Here threshold effects due to $Q\overline{Q}$ bound-state formation are taken into account in a smeared-out, average sense. Then the naïve cross section is multiplied by the squared wave function at the origin. In a colour-singlet channel this gives a net enhancement of the form

$$|\Psi^{(s)}(0)|^2 = \frac{X_{(s)}}{1-\exp(-X_{(s)})} \ , \quad \text{where } X_{(s)} = \frac{4}{3}\frac{\pi\alpha_s}{\beta} \ , \tag{137}$$

where β is the quark velocity, while in a colour octet channel there is a net suppression given by

$$|\Psi^{(8)}(0)|^2 = \frac{X_{(8)}}{\exp(-X_{(8)})-1} \ , \quad \text{where } X_{(8)} = \frac{1}{6}\frac{\pi\alpha_s}{\beta} \ . \tag{138}$$

The α_s factor in this expression is related to the energy scale of bound-state formation; it is selected independently from the one of the standard production cross section. The presence of a threshold factor affects the total rate and also kinematical distributions.

Heavy flavours, i.e. top and fourth generation, are assumed to be so short-lived that they decay before they have time to hadronize. This means that the light quark in the decay $Q \to W^{\pm}q$ inherits the colour of the heavy one. The current PYTHIA description represents a change of philosophy compared to older versions, formulated at a time when the top was thought to be much lighter than we now know it to be. For event shapes the difference between the two time orderings normally has only marginal effects [Sjö92a].

It should be noted that cross section calculations are different in the two cases. The top (or a fourth generation fermion) is assumed short-lived, and is treated like a resonance in the sense of section 7.6.2, i.e. the cross-section is reduced so as only to correspond to the channels left open by you. This also includes the restrictions on secondary decays, i.e. on the decays of a W^+ or a H^+ produced in the top decay. For b and c quarks, which are long-lived enough to form hadrons, no such reduction takes place. Branching ratios then have to be folded in by hand to get the correct cross sections. The logic behind this difference is that when hadronization takes place, one would normally decay the D^0 and D^+ meson according to different branching ratios. But which D mesons are to be formed is not known at the bottom quark creation, so one could not weight for that. For a t quark, which decays rapidly, this ambiguity does not exist, and so a reduction factor can be introduced directly coupled to the t quark production process.

This rule about cross-section calculations applies to all the processes explicitly set up to handle heavy flavour creation. In addition to the ones above, this means all the ones in Tables 17–24 where the fermion final state is given as capital letters ('Q' and 'F') and also flavours produced in resonance decays (Z^0, $W^{\pm}$, h^0, etc., including processes 165 and 166). However, heavy flavours could also be produced in a process such as 31, $q_i g \to q_k W^{\pm}$, where q_k could be a top quark. In this case, the thrust of the description is clearly on light flavours — the kinematics of the process is formulated in the massless fermion limit — so any top production is purely incidental. Since here the choice of scattered flavour is only done at a later stage, the top branching ratios are not correctly folded in to the hard scattering cross section. So, for applications like these, it is not recommended to restrict the allowed top decay modes. Often one might like to get rid of the possibility of producing top together with light flavours. This can be done by switching off (i.e. setting MDME(I,1)=0) the 'channels' $d \to W^-t$, $s \to W^-t$, $b \to W^-t$, $g \to t\bar{t}$ and $\gamma \to t\bar{t}$. Also any heavy flavours produced by parton shower evolution would not be correctly weighted into the cross section. However, currently top production is switched off both as a beam remnant (see MSTP(9) and in initial (see KFIN array) and final (see MSTJ(45)) state radiation.

In pair production of heavy flavour (top) in processes 81,82, 84 and 85, matrix elements are only given for one common mass, although Breit-Wigners are used to select two separate masses. As described in subsection 7.3, an average mass value is constructed for the matrix element evaluation so that the β_{34} kinematics factor can be retained.

Because of its large mass, it is possible that the top quark can decay to some not yet discovered particle. Some such alternatives are included in the program, such as $t \to bH^+$ or $t \to \tilde{G}\tilde{t}$.

These decays are not obtained by default, but can be included as discussed for the respective physics scenario.

8.2.3 J/ψ and other Hidden Heavy Flavours

ISUB =	86	$gg \to J/\psi g$
	87	$gg \to \chi_{0c} g$
	88	$gg \to \chi_{1c} g$
	89	$gg \to \chi_{2c} g$
	104	$gg \to \chi_{0c}$
	105	$gg \to \chi_{2c}$
	106	$gg \to J/\psi\gamma$
	107	$g\gamma \to J/\psi g$
	108	$\gamma\gamma \to J/\psi\gamma$

In PYTHIA one may distinguish between three main sources of J/ψ production.

1. Decays of B mesons and baryons.
2. Parton-shower evolution, wherein a c and a $\bar{c}$ quark produced in adjacent branchings (e.g. $g \to gg \to c\bar{c}c\bar{c}$) turn out to have so small an invariant mass that the pair collapses to a single particle.
3. Direct production, where a c quark loop gives a coupling between a set of gluons and a $c\bar{c}$ bound state. Higher-lying states, like the χ_c ones, may subsequently decay to J/ψ.

The first two sources are implicit in the production of b and c quarks, although the forcing specifically of J/ψ production is difficult. In this section are given the main processes for the third source, intended for applications at hadron colliders. Processes 104 and 105 are the equivalents of 87 and 89 in the limit of $p_\perp \to 0$; note that $gg \to J/\psi$ and $gg \to \chi_{1c}$ are forbidden and thus absent. As always one should beware of double-counting between 87 and 104, and between 89 and 105, and thus use either the one or the other depending on the kinematical domain to be studied. The cross sections depend on wave function values at the origin, see `PARP(38)` and `PARP(39)`. A review of the physics issues involved may be found in [Glo88] (note, however, that the choice of Q^2 scale is different in PYTHIA).

It is known that the above sources are not enough to explain the full J/ψ rate, and further production mechanisms have been proposed, extending on the more conventional treatment here [Can97].

While programmed for the charm system, it would be straightforward to apply these processes instead to bottom mesons. One needs to change the codes of states produced, which is achieved by `KFPR(ISUB,1)=KFPR(ISUB,1)+110` for the processes `ISUB` above, and changing the values of the wave functions at the origin, `PARP(38)` and `PARP(39)`.

8.2.4 Minimum bias

`MSEL` = 1, 2

ISUB =	91	elastic scattering
	92	single diffraction ($AB \to XB$)
	93	single diffraction ($AB \to AX$)
	94	double diffraction
	95	low-$p_\perp$ production

These processes are briefly discussed in section 7.7. They are mainly intended for interactions between hadrons, although one may also consider γp and γγ interactions in the options where the incoming photon(s) is (are) assumed resolved.

Uncertainties come from a number of sources, e.g. from the parameterizations of the various cross sections and slope parameters.

In diffractive scattering, the structure of the selected hadronic system may be regulated with `MSTP(101)`. No high-$p_\perp$ jet production in diffractive events is included so far.

The subprocess 95, low-$p_\perp$ events, is somewhat unique in that no meaningful physical border-line to high-$p_\perp$ events can be defined. Even if the QCD $2 \to 2$ high-$p_\perp$ processes are formally switched off, some of the generated events will be classified as belonging to this group, with a $p_\perp$ spectrum of interactions to match the 'minimum-bias' event sample. The generation of such jets is performed with the help of the auxiliary subprocess 96, see subsection 8.2.1. Only with the option `MSTP(82)=0` will subprocess 95 yield strictly low-$p_\perp$ events, events which will then probably not be compatible with any experimental data. A number of options exist for the detailed structure of low-$p_\perp$ events, see in particular `MSTP(81)` and `MSTP(82)`. Further details on the model(s) for minimum-bias events are found in section 11.2.

8.3 Physics with Incoming Photons

With recent additions, the machinery for photon physics has become rather extensive [Fri00]. The border between the physics of real photon interactions and of virtual photon ones is now bridged by a description that continuously interpolates between the two extremes, as summarized in section 7.7.2. Furthermore, the `'gamma/`*lepton*`'` option (where *lepton* is to be replaced by `e-`, `e+`, `mu-`, `mu+`, `tau-` or `tau+` as the case may be) in a `PYINIT` call gives access to an internally generated spectrum of photons of varying virtuality. The `CKIN(61) - CKIN(78)` variables can be used to set experimentally motivated x and Q^2 limits on the photon fluxes. With this option, and the default `MSTP(14)=30`, one automatically obtains a realistic first approximation to 'all' QCD physics of γ^*p and $\gamma^*\gamma^*$ interactions. The word 'all' clearly does not mean that a perfect description is guaranteed, or that all issues are addressed, but rather that the intention is to simulate all processes that give a significant contribution to the total cross section in whatever Q^2 range is being studied: jets, low-$p_\perp$ events, elastic and diffractive scattering, etc.

The material to be covered encompasses many options, several of which have been superseded by further developments but have been retained for backwards compatibility. Therefore it is here split into three sections. The first covers the physics of real photons and the subsequent one that of (very) virtual ones. Thereafter, in the final section, the threads are combined into a machinery applicable at all Q^2.

8.3.1 Photoproduction and $\gamma\gamma$ physics

`MSEL` = 1, 2, 4, 5, 6, 7, 8

`ISUB` =	33	$q_i\gamma \to q_i g$
	34	$f_i\gamma \to f_i\gamma$
	54	$g\gamma \to q_k\overline{q}_k$
	58	$\gamma\gamma \to f_k\overline{f}_k$
	80	$q_i\gamma \to q_k\pi^\pm$
	84	$g\gamma \to Q_k\overline{Q}_k$
	85	$\gamma\gamma \to F_k\overline{F}_k$

An (almost) real photon has both a point-like component and a hadron-like one. This means that several classes of processes may be distinguished, see section 7.7.2.

1. The processes listed above are possible when the photon interacts as a point-like particle, i.e. couples directly to quarks and leptons.
2. When the photon acts like a hadron, i.e. is resolved in a partonic substructure, then high-$p_\perp$ parton–parton interactions are possible, as described in sections 8.2.1 and 8.4.1. These

interactions may be further subdivided into VMD and anomalous (GVMD) ones [Sch93, Sch93a].

3. A hadron-like photon can also produce the equivalent of the minimum bias processes of section 8.2.4. Again, these can be subdivided into VMD and GVMD (anomalous) ones.

For γp events, we believe that the best description can be obtained when three separate event classes are combined, one for direct, one for VMD and one for GVMD/anomalous events, see the detailed description in [Sch93, Sch93a]. These correspond to `MSTP(14)` being 0, 2 and 3, respectively. The direct component is high-$p_\perp$ only, while VMD and GVMD contain both high-$p_\perp$ and low-$p_\perp$ events. The option `MSTP(14)=1` combines the VMD and GVMD/anomalous parts of the photon into one single resolved photon concept, which therefore is less precise than the full subdivision.

When combining three runs to obtain the totality of γp interactions, to the best of our knowledge, it is necessary to choose the $p_\perp$ cut-offs with some care, so as to represent the expected total cross section.

- The direct processes by themselves only depend on the `CKIN(3)` cut-off of the generation. In older program versions the preferred value was 0.5 GeV [Sch93, Sch93a]. In the more recent description in [Fri00], also eikonalization of direct with anomalous interactions into the GVMD event class is considered. That is, given a branching $\gamma \to q\bar{q}$, direct interactions are viewed as the low-$p_\perp$ events and anomalous ones as high-$p_\perp$ events that have to merge smoothly. Then the `CKIN(3)` cut-off is increased to the $p_{\perp\mathrm{min}}$ of multiple interactions processes, see `PARP(81)` (or `PARP(82)`, depending on minijet unitarization scheme). See `MSTP(18)` for a possibility to switch back to the older behaviour. However, full backwards compatibility cannot be assured, so the older scenarios are better simulated by using an older PYTHIA version.
- The VMD processes work as ordinary hadron–hadron ones, i.e. one obtains both low- and high-$p_\perp$ events by default, with dividing line set by $p_{\perp\mathrm{min}}$ above.
- Also the GVMD processes work like the VMD ones. Again this is a change from previous versions, where the anomalous processes only contained high-$p_\perp$ physics and the low-$p_\perp$ part was covered in the direct event class. See `MSTP(15)=5` for a possibility to switch back to the older behaviour, with comments as above for the direct class. A GVMD state is book-kept as a diffractive state in the event listing, even when it scatters 'elastically', since the subsequent hadronization descriptions are very similar.

The processes in points 1 and 2 can be simulated with a photon beam, i.e. when `'gamma'` appears as argument in the `PYINIT` call. It is then necessary to use option `MSTP(14)` to switch between a point-like and a resolved photon — it is not possible to simulate the two sets of processes in a single run. This would be the normal mode of operation for beamstrahlung photons, which have $Q^2 = 0$ but with a nontrivial energy spectrum that would be provided by some external routine.

For bremsstrahlung photons, the x and Q^2 spectrum can be simulated internally, with the `'gamma/`*lepton*`'` argument in the `PYINIT` call. This is the recommended procedure, wherein direct and resolved processes can be mixed. An older — now not recommended — alternative is to use a parton-inside-electron structure function concept, obtainable with a simple `'e-'` (or other lepton) argument in `PYINIT`. To access these quark and gluon distributions inside the photon (itself inside the electron), `MSTP(12)=1` must then be used. Also the default value `MSTP(11)=1` is required for the preceding step, that of finding photons inside the electron. Also here the direct and resolved processes may be generated together. However, this option only works for high-$p_\perp$ physics. It is not possible to have also the low-$p_\perp$ physics (including multiple interactions in high-$p_\perp$ events) for an electron beam. Kindly note that subprocess 34 contains both the scattering of an electron off a photon and the scattering of a quark (inside a photon inside an electron) off a photon; the former can be switched off with the help of the `KFIN` array.

If you are only concerned with standard QCD physics, the option `MSTP(14)=10` or the default `MSTP(14)=30` gives an automatic mixture of the VMD, direct and GVMD/anomalous event classes. The mixture is properly given according to the relative cross sections. Whenever

possible, this option is therefore preferable in terms of user-friendliness. However, it can only work because of a completely new layer of administration, not found anywhere else in PYTHIA. For instance, a subprocess like qg $\to$ qg is allowed in several of the classes, but appears with different sets of parton distributions and different $p_\perp$ cut-offs in each of these, so that it is necessary to switch gears between each event in the generation. It is therefore not possible to avoid a number of restrictions on what you can do in this case:

- The `MSTP(14)=10` and `=30` options can only be used for incoming photon beams, with or without convolution with the bremsstrahlung spectrum, i.e. when `'gamma'` or `'gamma/`*lepton*`'` is the argument in the `PYINIT` call.
- The machinery has only been set up to generate standard QCD physics, specifically either 'minimum-bias' one or high-$p_\perp$ jets. There is thus no automatic mixing of processes only for heavy-flavour production, say, or of some exotic particle. For minimum bias, you are not allowed to use the `CKIN` variables at all. This is not a major limitation, since it is in the spirit of minimum-bias physics not to impose any constraints on allowed jet production. (If you still do, these cuts will be ineffective for the VMD processes but take effect for the other ones, giving inconsistencies.) The minimum-bias physics option is obtained by default; by switching from `MSEL=1` to `MSEL=2` also the elastic and diffractive components of the VMD and GVMD parts are included. High-$p_\perp$ jet production is obtained by setting the `CKIN(3)` cut-off larger than the $p_{\perp\mathrm{min}}(W^2)$ of the multiple interactions scenario. For lower input `CKIN(3)` values the program will automatically switch back to minimum-bias physics.
- Multiple interactions become possible in both the VMD and GVMD sector, with the average number of interactions given by the ratio of the jet to the total cross section. Currently only the simpler default scenario `MSTP(82)=1` is implemented, however, i.e. the more sophisticated variable-impact-parameter ones need further physics studies and model development.
- Some variables are internally recalculated and reset, notably `CKIN(3)`. This is because it must have values that depend on the component studied. It can therefore not be modified without changing `PYINPR` and recompiling the program, which obviously is a major exercise.
- Pileup events are not at all allowed.

Also, a warning about the usage of PDFLIB for photons. So long as `MSTP(14)=1`, i.e. the photon is not split up, PDFLIB is accessed by `MSTP(56)=2` and `MSTP(55)` as the parton distribution set. However, when the VMD and anomalous pieces are split, the VMD part is based on a rescaling of pion distributions by VMD factors (except for the SaS sets, that already come with a separate VMD piece). Therefore, to access PDFLIB for `MSTP(14)=10`, it is not correct to set `MSTP(56)=2` and a photon distribution in `MSTP(55)`. Instead, one should put `MSTP(56)=2`, `MSTP(54)=2` and a pion distribution code in `MSTP(53)`, while `MSTP(55)` has no function. The anomalous part is still based on the SaS parameterization, with `PARP(15)` as main free parameter.

Currently, hadrons are not defined with any photonic content. None of the processes are therefore relevant in hadron–hadron collisions. In ep collisions, the electron can emit an almost real photon, which may interact directly or be resolved. In e^+e^- collisions, one may have direct, singly-resolved or doubly-resolved processes.

The $\gamma\gamma$ equivalent to the γp description involves six different event classes, see section 7.7.2. These classes can be obtained by setting `MSTP(14)` to 0, 2, 3, 5, 6 and 7, respectively. If one combines the VMD and anomalous parts of the parton distributions of the photon, in a more coarse description, it is enough to use the `MSTP(14)` options 0, 1 and 4. The cut-off procedures follows from the ones used for the γp ones above.

As with γp events, the options `MSTP(14)=10` or `MSTP(14)=30` give a mixture of the six possible $\gamma\gamma$ event classes. The same complications and restrictions exist here as already listed above.

Process 54 generates a mixture of quark flavours; allowed flavours are set by the gluon `MDME`

values. Process 58 can generate both quark and lepton pairs, according to the MDME values of the photon. Processes 84 and 85 are variants of these matrix elements, with fermion masses included in the matrix elements, but where only one flavour can be generated at a time. This flavour is selected as described for processes 81 and 82 in section 8.2.2, with the exception that for process 85 the 'heaviest' flavour allowed for photon splitting takes to place of the heaviest flavour allowed for gluon splitting. Since lepton KF codes come after quark ones, they are counted as being 'heavier', and thus take precedence if they have been allowed.

Process 80 is a higher twist one. The theory for such processes is rather shaky, so results should not be taken too literally. The messy formulae given in [Bag82] have not been programmed in full, instead the pion form factor has been parameterized as $Q^2 F_\pi(Q^2) \approx 0.55/\ln Q^2$, with Q in GeV.

8.3.2 Deeply Inelastic Scattering and $\gamma^*\gamma^*$ physics

MSEL = 1, 2, 35, 36, 37, 38

ISUB =	10	$\mathrm{f}_i\mathrm{f}_j \to \mathrm{f}_k\mathrm{f}_l$
	83	$\mathrm{q}_i\mathrm{f}_j \to \mathrm{Q}_k\mathrm{f}_l$
	99	$\gamma^*\mathrm{q} \to \mathrm{q}$
	131	$\mathrm{f}_i\gamma^*_\mathrm{T} \to \mathrm{f}_i\mathrm{g}$
	132	$\mathrm{f}_i\gamma^*_\mathrm{L} \to \mathrm{f}_i\mathrm{g}$
	133	$\mathrm{f}_i\gamma^*_\mathrm{T} \to \mathrm{f}_i\gamma$
	134	$\mathrm{f}_i\gamma^*_\mathrm{L} \to \mathrm{f}_i\gamma$
	135	$\mathrm{g}\gamma^*_\mathrm{T} \to \mathrm{f}_i\bar{\mathrm{f}}_i$
	136	$\mathrm{g}\gamma^*_\mathrm{L} \to \mathrm{f}_i\bar{\mathrm{f}}_i$
	137	$\gamma^*_\mathrm{T}\gamma^*_\mathrm{T} \to \mathrm{f}_i\bar{\mathrm{f}}_i$
	138	$\gamma^*_\mathrm{T}\gamma^*_\mathrm{L} \to \mathrm{f}_i\bar{\mathrm{f}}_i$
	139	$\gamma^*_\mathrm{L}\gamma^*_\mathrm{T} \to \mathrm{f}_i\bar{\mathrm{f}}_i$
	140	$\gamma^*_\mathrm{L}\gamma^*_\mathrm{L} \to \mathrm{f}_i\bar{\mathrm{f}}_i$

Among the processes in this section, 10 and 83 are intended to stand on their own, while the rest are part of the newer machinery for γ^*p and $\gamma^*\gamma^*$ physics. We therefore separate the description in this section into these two main parts.

The Deeply Inelastic Scattering (DIS) processes, i.e. t-channel electroweak gauge boson exchange, are traditionally associated with interactions between a lepton or neutrino and a hadron, but processes 10 and 83 can equally well be applied for qq scattering in hadron colliders (with a cross section much smaller than corresponding QCD processes, however). If applied to incoming $\mathrm{e}^+\mathrm{e}^-$ beams, process 10 corresponds to Bhabha scattering.

For process 10 both γ, Z^0 and $\mathrm{W}^\pm$ exchange contribute, including interference between γ and Z^0. The switch MSTP(21) may be used to restrict to only some of these, e.g. neutral or charged current only.

The option MSTP(14)=10 (see previous section) has now been extended so that it also works for DIS of an electron off a (real) photon, i.e. process 10. What is obtained is a mixture of the photon acting as a vector meson and it acting as an anomalous state. This should therefore be the sum of what can be obtained with MSTP(14)=2 and =3. It is distinct from MSTP(14)=1 in that different sets are used for the parton distributions — in MSTP(14)=1 all the contributions to the photon distributions are lumped together, while they are split in VMD and anomalous parts for MSTP(14)=10. Also the beam remnant treatment is different, with a simple Gaussian distribution (at least by default) for MSTP(14)=1 and the VMD part of MSTP(14)=10, but a powerlike distribution $\mathrm{d}k_\perp^2/k_\perp^2$ between PARP(15) and Q for the anomalous part of MSTP(14)=10.

To access this option for e and γ as incoming beams, it is only necessary to set MSTP(14)=10 and keep MSEL at its default value. Unlike the corresponding option for γp and $\gamma\gamma$, no cuts are overwritten, i.e. it is still your responsibility to set these appropriately.

Cuts especially appropriate for DIS usage include either `CKIN(21)-CKIN(22)` or `CKIN(23)-CKIN(24)` for the x range (former or latter depending on which side is the incoming real photon), `CKIN(35)-CKIN(36)` for the Q^2 range, and `CKIN(39)-CKIN(40)` for the W^2 range.

In principle, the DIS x variable of an event corresponds to the x value stored in `PARI(33)` or `PARI(34)`, depending on which side the incoming hadron is on, while the DIS $Q^2 = -\hat{t} =$-`PARI(15)`. However, just like initial- and final-state radiation can shift jet momenta, they can modify the momentum of the scattered lepton. Therefore the DIS x and Q^2 variables are not automatically conserved. An option, on by default, exists in `MSTP(23)`, where the event can be 'modified back' so as to conserve x and Q^2, but this option is rather primitive and should not be taken too literally.

Process 83 is the equivalent of process 10 for $\mathrm{W}^{\pm}$ exchange only, but with the heavy-quark mass included in the matrix element. In hadron colliders it is mainly of interest for the production of very heavy flavours, where the possibility of producing just one heavy quark is kinematically favoured over pair production. The selection of the heavy flavour is already discussed in section 8.2.2.

Turning to the other processes, part of the $\gamma^*\mathrm{p}$ and $\gamma^*\gamma^*$ process-mixing machineries, 99 has close similarities with the above discussed 10 one. Whereas 10 would simulate the full process $\mathrm{eq} \to \mathrm{eq}$, 99 assumes a separate machinery for the flux of virtual photons, $\mathrm{e} \to \mathrm{e}\gamma^*$ and only covers the second half of the process, $\gamma^*\mathrm{q} \to \mathrm{q}$. One limitation of this factorization is that only virtual photons are considered in process 99, not contributions from the Z^0 neutral current or the $\mathrm{W}^{\pm}$ charged current.

Note that 99 has no correspondence in the real-photon case, but has to vanish in this limit by gauge invariance, or indeed by simple kinematics considerations. This, plus the desire to avoid double-counting with real-photon physics processes, is why the cross section for this process is explicitly made to vanish for photon virtuality $Q^2 \to 0$, eq. (127), also when parton distributions have not been constructed to fulfil this, see `MSTP(19)`. (No such safety measures are present in 10, again illustrating how the two are intended mainly to be used at large or at small Q^2, respectively.)

For a virtual photon, processes 131–136 may be viewed as first-order corrections to 99. The three with a transversely polarized photon, 131, 133 and 135, smoothly reduce to the real-photon direct (single-resolved for $\gamma\gamma$) processes 33, 34 and 54. The other three, corresponding to the exchange of a longitudinal photon, vanish like Q^2 for $Q^2 \to 0$. The double-counting issue with process 99 is solved by requiring the latter process not to contain any shower branchings with a $p_\perp$ above the lower $p_\perp$ cut-off of processes 131-136. The cross section is then to be reduced accordingly, see eq. (128) and the discussion there, and again `MSTP(19)`.

We thus see that process 99 by default is a low-$p_\perp$ process in about the same sense as process 95, giving 'what is left' of the total cross section when jet events have been removed. Therefore, it will be switched off in event class mixes such as `MSTP(14)=30` if `CKIN(3)` is above $p_{\perp\mathrm{min}}(W^2)$ and `MSEL` is not 2. There is a difference, however, in that process 99 events still are allowed to contain shower evolution (although currently only the final-state kind has been implemented), since the border to the other processes is at $p_\perp = Q$ for large Q and thus need not be so small. The $p_\perp$ scale of the 'hard process', stored e.g. in `PARI(17)` always remains 0, however. (Other `PARI` variables defined for normal $2 \to 2$ and $2 \to 1$ processes are not set at all, and may well contain irrelevant junk left over from previous events.)

Processes 137–140, finally, are extensions of process 58 from the real-photon limit to the virtual-photon case, and correspond to the direct process of $\gamma^*\gamma^*$ physics. The four cases correspond to either of the two photons being either transversely or longitudinally polarized. As above, the cross section of a longitudinal photon vanishes when its virtuality approaches 0.

8.3.3 Photon physics at all virtualities

ISUB =	direct×direct:	137, 138, 139, 140
	direct×resolved:	131, 132, 135, 136
	DIS×resolved:	99
	resolved×resolved, high-$p_\perp$:	11, 12, 13, 28, 53, 68
	resolved×resolved, low-$p_\perp$:	91, 92, 93, 94, 95

where 'resolved' is a hadron or a VMD or GVMD photon.

At intermediate photon virtualities, processes described in both of the sections above are allowed, and have to be mixed appropriately. The sets are of about equal importance at around $Q^2 \sim m_\rho^2 \sim 1$ GeV^2, but the transition is gradual over a larger Q^2 range. The ansatz for this mixing is given by eq. (129) for γ^*p events and eq. (130) for $\gamma^*\gamma^*$ ones. In short, for direct and DIS processes the photon virtuality explicitly enters in the matrix element expressions, and thus is easily taken into account. For resolved photons, perturbation theory does not provide a unique answer, so instead cross sections are suppressed by dipole factors, $(m^2/(m^2+Q^2))^2$, where $m = m_V$ for a VMD state and $m = 2k_\perp$ for a GVMD state characterized by a $k_\perp$ scale of the $\gamma^* \to q\bar{q}$ branching. These factors appear explicitly for total, elastic and diffractive cross sections, and are also implicitly used e.g. in deriving the SaS parton distributions for virtual photons. Finally, some double-counting need to be removed, between direct and DIS processes as mentioned in the previous section, and between resolved and DIS at large x.

Since the mixing is not trivial, it is recommended to use the default `MSTP(14)=30` to obtain it in one go and hopefully consistently, rather than building it up by combining separate runs. The main issues still under your control include, among others

- The `CKIN(61) - CKIN(78)` should be used to set the range of x and Q^2 values emitted from the lepton beams. That way one may decide between almost real or very virtual photons, say. Also some other quantities, like W^2, can be constrained to desirable ranges.
- Whether or not minimum bias events are simulated depends on the `CKIN(3)` value, just like in hadron physics. The only difference is that the initialization energy scale $W_{\rm init}$ is selected in the allowed W range rather than to be the full c.m. energy.
  For a high `CKIN(3)`, `CKIN(3)`$> p_{\perp\rm min}(W^2_{\rm init})$, only jet production is included. Then further `CKIN` values can be set to constrain e.g. the rapidity of the jets produced.
  For a low `CKIN(3)`, `CKIN(3)`$< p_{\perp\rm min}(W^2_{\rm init})$, like the default value `CKIN(3) = 0`, low-$p_\perp$ physics is switched on together with jet production, with the latter properly eikonalized to be lower than the total one. The ordinary `CKIN` cuts, not related to the photon flux, cannot be used here.
  For a low `CKIN(3)`, when `MSEL=2` instead of the default `=1`, also elastic and diffractive events are simulated.
- The impact of resolved longitudinal photons is not unambiguous, e.g. only recently the first parameterization of parton distributions appeared [Chý00]. Different simple alternatives can be probed by changing `MSTP(17)` and associated parameters.
- The choice of scales to use in parton distributions for jet rates is always ambiguous, but depends on even more scales for virtual photons than in hadronic collisions. `MSTP(32)` allows a choice between several alternatives.
- The matching of $p_\perp$ generation by shower evolution to that by primordial $k_\perp$ is a general problem, for photons with an additional potential source in the $\gamma^* \to q\bar{q}$ vertex. `MSTP(66)` offer some alternatives.
- `PARP(15)` is the k_0 parameter separating VMD from GVMD.
- `PARP(18)` is the k_ρ parameter in GVMD total cross sections.
- `MSTP(16)` selects the momentum variable for an $e \to e\gamma^*$ branching.
- `MSTP(18)` regulates the choice of $p_{\perp\rm min}$ for direct processes.
- `MSTP(19)` regulates the choice of partonic cross section in process 99, $\gamma^* q \to q$.

- `MSTP(20)` regulates the suppression of the resolved cross section at large x.

The above list is not complete, but gives some impression what can be done.

8.4 Electroweak Gauge Bosons

This section covers the production and/or exchange of γ, Z^0 and $W^\pm$ gauge bosons, singly and in pairs. The topic of longitudinal gauge-boson scattering at high energies is deferred to the Higgs section, since the presence or absence of a Higgs here makes a big difference.

8.4.1 Prompt photon production

`MSEL` = 10

`ISUB` =	14	$q_i\bar{q}_i \to g\gamma$
	18	$f_i\bar{f}_i \to \gamma\gamma$
	29	$q_i g \to q_i\gamma$
	114	$gg \to \gamma\gamma$
	115	$gg \to g\gamma$

In hadron colliders, processes `ISUB` = 14 and 29 give the main source of single-γ production, with `ISUB` = 115 giving an additional contribution which, in some kinematics regions, may become important. For γ-pair production, the process `ISUB` = 18 is often overshadowed in importance by `ISUB` = 114.

Another source of photons is bremsstrahlung off incoming or outgoing quarks. This has to be treated on an equal footing with QCD parton showering. For time-like parton-shower evolution, i.e. in the final-state showering and in the side branches of the initial-state showering, photon emission may be switched on or off with `MSTJ(41)`. Photon radiation off the space-like incoming quark legs is not yet included, but should be of lesser importance for production at reasonably large $p_\perp$ values. Radiation off an incoming electron is included in a leading-log approximation.

Warning: the cross sections for the box graphs 114 and 115 become very complicated, numerically unstable and slow when the full quark mass dependence is included. For quark masses much below the $\hat{s}$ scale, the simplified massless expressions are therefore used — a fairly accurate approximation. However, there is another set of subtle numerical cancellations between different terms in the massive matrix elements in the region of small-angle scattering. The associated problems have not been sorted out yet. There are therefore two possible solutions. One is to use the massless formulae throughout. The program then becomes faster and numerically stable, but does not give, for example, the characteristic dip (due to destructive interference) at top threshold. This is the current default procedure, with five flavours assumed, but this number can be changed in `MSTP(38)`. The other possibility is to impose cuts on the scattering angle of the hard process, see `CKIN(27)` and `CKIN(28)`, since the numerically unstable regions are when $|\cos\hat{\theta}|$ is close to unity. It is then also necessary to change `MSTP(38)` to 0.

8.4.2 Single W/Z production

MSEL = 11, 12, 13, 14, 15, (21)

ISUB =	1	$f_i\bar{f}_i \to \gamma^*/Z^0$
	2	$f_i\bar{f}_j \to W^+$
	15	$f_i\bar{f}_i \to g(\gamma^*/Z^0)$
	16	$f_i\bar{f}_j \to gW^+$
	19	$f_i\bar{f}_i \to \gamma(\gamma^*/Z^0)$
	20	$f_i\bar{f}_j \to \gamma W^+$
	30	$f_i g \to f_i(\gamma^*/Z^0)$
	31	$f_i g \to f_k W^+$
	35	$f_i\gamma \to f_i(\gamma^*/Z^0)$
	36	$f_i\gamma \to f_k W^+$
	(141)	$f_i\bar{f}_i \to \gamma/Z^0/Z'^0$

This group consists of $2 \to 1$ processes, i.e. production of a single resonance, and $2 \to 2$ processes, where the resonance is recoiling against a jet or a photon. The process 141, which also is listed here, is described further elsewhere.

With initial-state showers turned on, the $2 \to 1$ processes also generate additional jets; in order to avoid double-counting, the corresponding $2 \to 2$ processes should therefore not be turned on simultaneously. The basic rule is to use the $2 \to 1$ processes for inclusive generation of W/Z, i.e. where the bulk of the events studied have $p_\perp \ll m_{W/Z}$. With the introduction of explicit matrix-element-inspired corrections to the parton shower [Miu99], also the high-$p_\perp$ tail is well described in this approach, thus offering an overall good decription of the full $p_\perp$ spectrum of gauge bosons [Bál01].

If one is interested in the high-$p_\perp$ tail only, however, the generation efficiency will be low. It is here better to start from the $2 \to 2$ matrix elements and add showers to these. However, the $2 \to 2$ matrix elements are divergent for $p_\perp \to 0$, and should not be used down to the low-$p_\perp$ region, or one may get unphysical cross sections. As soon as the generated $2 \to 2$ cross section corresponds to a non-negligible fraction of the total $2 \to 1$ one, say 10%–20%, Sudakov effects are likely to be affecting the shape of the $p_\perp$ spectrum to a corresponding extent, and results should not be trusted.

The problems of double-counting and Sudakov effects apply not only to W/Z production in hadron colliders, but also to a process like $e^+e^- \to Z^0\gamma$, which clearly is part of the initial-state radiation corrections to $e^+e^- \to Z^0$ obtained for `MSTP(11)=1`. As is the case for Z production in association with jets, the $2 \to 2$ process should therefore only be used for the high-$p_\perp$ region.

The Z^0 of subprocess 1 includes the full interference structure γ^*/Z^0; via `MSTP(43)` you can select to produce only γ^*, only Z^0, or the full γ^*/Z^0. The same holds true for the Z'^0 of subprocess 141; via `MSTP(44)` any combination of γ^*, Z^0 and Z'^0 can be selected. Thus, subprocess 141 with `MSTP(44)=4` is essentially equivalent to subprocess 1 with `MSTP(43)=3`; however, process 141 also includes the possibility of a decay into Higgses. Also processes 15, 19, 30 and 35 contain the full mixture of γ^*/Z^0, with `MSTP(43)` available to change this.

Note that process 1, with only $q\bar{q} \to \gamma^* \to \ell^+\ell^-$ allowed, and studied in the region well below the Z^0 mass, is what is conventionally called Drell–Yan. This latter process therefore does not appear under a separate heading, but can be obtained by a suitable setting of switches and parameters.

A process like $f_i\bar{f}_j \to \gamma W^+$ is only included in the limit that the γ is emitted in the 'initial state', while the possibility of a final-state radiation off the W^+ decay products is not explicitly included (but can be obtained implicitly by the parton-shower machinery) and various interference terms are not at all present. Some caution must therefore be exercised; see also section 8.4.3 for related comments.

For the $2 \to 1$ processes, the Breit–Wigner includes an $\hat{s}$-dependent width, which should

provide an improved description of line shapes. In fact, from a line-shape point of view, process 1 should provide a more accurate simulation of e^+e^- annihilation events than the dedicated e^+e^- generation scheme of PYEEVT (see section 6.1). Another difference is that PYEEVT only allows the generation of $\gamma^*/Z^0 \to q\bar{q}$, while process 1 additionally contains $\gamma^*/Z^0 \to \ell^+\ell^-$ and $\gamma^*/Z^0 \to \nu\bar{\nu}$. The parton-shower and fragmentation descriptions are the same, but the process 1 implementation only contains a partial interface to the first- and second-order matrix-element options available in PYEEVT, see MSTP(48).

All processes in this group have been included with the correct angular distribution in the subsequent $W/Z \to f\bar{f}$ decays. In process 1 also fermion mass effects have been included in the angular distributions, while this is not the case for the other ones. Normally mass effects are not large anyway.

The process $e^+e^- \to e^+e^-Z^0$ can be simulated in two different ways. One is to make use of the e 'sea' distribution inside e, i.e. have splittings $e \to \gamma \to e$. This can be obtained, together with ordinary Z^0 production, by using subprocess 1, with MSTP(11)=1 and MSTP(12)=1. Then the contribution of the type above is 5.0 pb for a 500 GeV e^+e^- collider, compared with the correct 6.2 pb [Hag91]. Alternatively one may use process 35, with MSTP(11)=1 and MSTP(12)=0. This process has a singularity in the forward direction, regularized by the electron mass and also sensitive to the virtuality of the photon. It is therefore among the few where the incoming masses have been included in the matrix element expression. Nevertheless, it may be advisable to set small lower cut-offs, e.g. CKIN(3)=CKIN(5)=0.01, if one should experience problems (e.g. at higher energies).

Process 36, $f\gamma \to f'W^\pm$ may have corresponding problems; except that in e^+e^- the forward scattering amplitude for $e\gamma \to \nu W$ is killed (radiation zero), which means that the differential cross section is vanishing for $p_\perp \to 0$. It is therefore feasible to use the default CKIN(3) and CKIN(5) values in e^+e^-, and one also comes closer to the correct cross section.

The process $gg \to Z^0b\bar{b}$, formerly available as process 131, has been removed from the current version, since the implementation turned out to be slow and unstable. However, process 1 with incoming flavours set to be $b\bar{b}$ (by KFIN(1,5)= KFIN(1,-5)=KFIN(2,5)=KFIN(2,-5)=1 and everything else =0) provides an alternative description, where the additional $b\bar{b}$ are generated by $g \to b\bar{b}$ branchings in the initial-state showers. (Away from the low-$p_\perp$ region, process 30 with KFIN values as above except that also incoming gluons are allowed, offers yet another description. Here it is in terms of $gb \to Z^0b$, with only one further $g \to b\bar{b}$ branching constructed by the shower.) At first glance, the shower approach would seem less reliable than the full $2 \to 3$ matrix element. The relative lightness of the b quark will generate large logs of the type $\ln(m_Z^2/m_b^2)$, however, that ought to be resummed [Car00]. This is implicit in the parton-density approach of incoming b quarks but absent from the lowest-order $gg \to Z^0b\bar{b}$ matrix elements. Therefore actually the shower approach may be the more accurate of the two. Within the general range of uncertainty of any leading-order description, at least it is not any worse.

8.4.3 W/Z pair production

MSEL = 15

ISUB =	22	$f_i\bar{f}_i \to (\gamma^*/Z^0)(\gamma^*/Z^0)$
	23	$f_i\bar{f}_j \to Z^0W^+$
	25	$f_i\bar{f}_i \to W^+W^-$
	69	$\gamma\gamma \to W^+W^-$
	70	$\gamma W^+ \to Z^0W^+$

In this section we mainly consider the production of W/Z pairs by fermion–antifermion annihilation, but also include two processes which involve γ/W beams. Scatterings between gauge-boson pairs, i.e. processes like $W^+W^- \to Z^0Z^0$, depend so crucially on the assumed Higgs scenario that they are considered separately in section 8.5.2.

The cross sections used for the above processes are those derived in the narrow-width limit, but have been extended to include Breit–Wigner shapes with mass-dependent widths for the final-state particles. In process 25, the contribution from Z^0 exchange to the cross section is now evaluated with the fixed nominal Z^0 mass and width in the propagator. If instead the actual mass and the running width were to be used, it gives a diverging cross section at large energies, by imperfect gauge cancellation.

However, one should realize that other graphs, not included here, can contribute in regions away from the W/Z mass. This problem is especially important if several flavours coincide in the four-fermion final state. Consider, as an example, $e^+e^- \to \mu^+\mu^-\nu_\mu\overline{\nu}_\mu$. Not only would such a final state receive contributions from intermediate Z^0Z^0 and W^+W^- states, but also from processes $e^+e^- \to Z^0 \to \mu^+\mu^-$, followed either by $\mu^+ \to \mu^+Z^0 \to \mu^+\nu_\mu\overline{\nu}_\mu$, or by $\mu^+ \to \overline{\nu}_\mu W^+ \to \overline{\nu}_\mu\mu^+\nu_\mu$. In addition, all possible interferences should be considered. Since this is not done, the processes have to be used with some sound judgement. Very often, one may wish to constrain a lepton pair mass to be close to m_Z, in which case a number of the possible 'other' processes are negligible.

For the W pair production graph, one experimental objective is to do precision measurements of the cross section near threshold. Then also other effects enter. One such is Coulomb corrections, induced by photon exchange between the two W's and their decay products. The gauge invariance issues induced by the finite W lifetime are not yet fully resolved, and therefore somewhat different approximate formulae may be derived [Kho96]. The options in `MSTP(40)` provide a reasonable range of uncertainty.

Of the above processes, the first contains the full $f_i\overline{f}_i \to (\gamma^*/Z^0)(\gamma^*/Z^0)$ structure, obtained by a straightforward generalization of the formulae in ref. [Gun86] (done by one of the PYTHIA authors). Of course, the possibility of there being significant contributions from graphs that are not included is increased, in particular if one γ^* is very light and therefore could be a bremsstrahlung-type photon. It is possible to use `MSTP(43)` to recover the pure Z^0 case, i.e. $f_i\overline{f}_i \to Z^0Z^0$ exclusively. In processes 23 and 70, only the pure Z^0 contribution is included.

Full angular correlations are included for the first three processes, i.e. the full $2 \to 2 \to 4$ matrix elements are included in the resonance decays, including the appropriate γ^*/Z^0 interference in process 22. In the latter two processes no spin information is currently preserved, i.e. the W/Z bosons are allowed to decay isotropically.

We remind you that the mass ranges of the two resonances may be set with the `CKIN(41)` - `CKIN(44)` parameters; this is particularly convenient, for instance, to pick one resonance almost on the mass shell and the other not.

8.5 Higgs Production

A fair fraction of all the processes in PYTHIA deal with Higgs production in one form or another. This multiplication is caused by the need to consider production by several different processes, depending on Higgs mass and machine type. Further, the program contains a full two-Higgs-multiplet scenario, as predicted for example in the Minimal Supersymmetric extension of the Standard Model (MSSM). Therefore the continued discussion is, somewhat arbitrarily, subdivided into a few different scenarios. Doubly-charged Higgs particles appear in left–right symmetric models, and are covered in section 8.6.3.

8.5.1 Light Standard Model Higgs

`MSEL` = 16, 17, 18

`ISUB` =	3	$f_i\bar{f}_i \to h^0$
	24	$f_i\bar{f}_i \to Z^0h^0$
	26	$f_i\bar{f}_j \to W^+h^0$
	102	$gg \to h^0$
	103	$\gamma\gamma \to h^0$
	110	$f_i\bar{f}_i \to \gamma h^0$
	111	$f_i\bar{f}_i \to gh^0$
	112	$f_ig \to f_ih^0$
	113	$gg \to gh^0$
	121	$gg \to Q_k\overline{Q}_kh^0$
	122	$q_i\overline{q}_i \to Q_k\overline{Q}_kh^0$
	123	$f_if_j \to f_if_jh^0$ (Z^0Z^0 fusion)
	124	$f_if_j \to f_kf_lh^0$ (W^+W^- fusion)

In this section we discuss the production of a reasonably light Standard Model Higgs, below 700 GeV, say, so that the narrow width approximation can be used with some confidence. Below 400 GeV there would certainly be no trouble, while above that the narrow width approximation is gradually starting to break down.

In a hadron collider, the main production processes are 102, 123 and 124, i.e. gg, Z^0Z^0 and W^+W^- fusion. In the latter two processes, it is also necessary to take into account the emission of the space-like W/Z bosons off quarks, which in total gives the $2 \to 3$ processes above.

Further processes of lower cross sections may be of interest because of easier signals. For instance, processes 24 and 26 give associated production of a Z or a W together with the h^0. There is also the processes 3 (see below), 121 and 122, which involve production of heavy flavours.

Process 3 contains contributions from all flavours, but is completely dominated by the subprocess $t\bar{t} \to h^0$, i.e. by the contribution from the top sea distributions. Assuming, of course, that parton densities for top quarks are available, which is no longer the case in current parameterizations. This process is by now known to overestimate the cross section for Higgs production as compared with a more careful calculation based on the subprocess $gg \to t\bar{t}h^0$, process 121. The difference between the two is that in process 3 the t and $\bar{t}$ are added by the initial-state shower, while in 121 the full matrix element is used. The price to be paid is that the complicated multibody phase space in process 121 makes the program run slower than with most other processes. As usual, it would be double-counting to include the same flavour both with 3 and 121. Process 122 is similar in structure to 121, but is less important. In both process 121 and 122 the produced quark is assumed to be a t; this can be changed in `KFPR(121,2)` and `KFPR(122,2)` before initialization, however. For b quarks it could well be that process 3 with $b\bar{b} \to h^0$ is more reliable than process 121 with $gg \to b\bar{b}h^0$ [Car00]; see the discussion on $Z^0b\bar{b}$ final states in section 8.4.2. Thus it would make sense to run with all quarks up to and including b simulated in process 3 and then consider t quarks separately in process 121. Assuming no t parton densities, this would actually be the default behaviour, meaning that the two could be combined in the same run without doublecounting.

The two subprocess 112 and 113, with a Higgs recoiling against a quark or gluon jet, are also effectively generated by initial-state corrections to subprocess 102. Thus, in order to avoid double-counting, just as for the case of Z^0/W^+ production, section 8.4.2, these subprocesses should not be switched on simultaneously. Process 111, $q\overline{q} \to gh^0$ is different, in the sense that it proceeds through an *s*-channel gluon coupling to a heavy-quark loop, and that therefore the emitted gluon is necessary in the final state in order to conserve colours. It is not to be confused with a gluon-radiation correction to the Born-level process 3, $q\overline{q} \to h^0$, since process 3 vanishes

for massless quarks while process 111 is mainly intended for such. The lack of a matching Born-level process shows up by process 111 being vanishing in the $p_\perp \to 0$ limit. Numerically it is of negligible importance, except at very large $p_\perp$ values. Process 102, possibly augmented by 111, should thus be used for inclusive production of Higgs, and 111–113 for the study of the Higgs subsample with high transverse momentum.

A warning is that the matrix-element expressions for processes 111–113 are very lengthy and the coding therefore more likely to contain some errors and numerical instabilities than for most other processes. Therefore the full expressions are only available by setting the non-default value `MSTP(38)=0`. Instead the default is based on the simplified expressions obtainable if only the top quark contribution is considered, in the $m_t \to \infty$ limit [Ell88]. As a slight improvement, this expression is rescaled by the ratio of the $gg \to h^0$ cross sections (or, equivalently, the $h \to gg$ partial widths) of the full calculation and that in the $m_t \to \infty$ limit. Simple checks show that this approach normally agrees with the full expressions to within $\sim 20\%$, which is small compared with other uncertainties. The agreement is worse for process 111 alone, about a factor of 2, but this process is small anyway. We also note that the matrix element correction factors, used in the initial-state parton shower for process 102, subsection 10.3.5, are based on the same $m_t \to \infty$ limit expressions, so that the high-$p_\perp$ tail of process 102 is well matched to the simple description of process 112 and 113.

In e^+e^- annihilation, associated production of an h^0 with a Z^0, process 24, is usually the dominant one close to threshold, while the Z^0Z^0 and W^+W^- fusion processes 123 and 124 win out at high energies. Process 103, $\gamma\gamma$ fusion, may also be of interest, in particular when the possibilities of beamstrahlung photons and backscattered photons are included (see subsection 7.1.3). Process 110, which gives an h^0 in association with a γ, is a loop process and is therefore suppressed in rate. It would have been of interest for a h^0 mass above 60 GeV at LEP 1, since its phase space suppression there is less severe than for the associated production with a Z^0. Now it is not likely to be of any further interest.

The branching ratios of the Higgs are very strongly dependent on the mass. In principle, the program is set up to calculate these correctly, as a function of the actual Higgs mass, i.e. not just at the nominal mass. However, higher-order corrections may at times be important and not fully unambiguous; see for instance `MSTP(37)`.

Since the Higgs is a spin-0 particle it decays isotropically. In decay processes such as $h^0 \to W^+W^-/Z^0Z^0 \to 4$ fermions angular correlations are included [Lin97]. Also in processes 24 and 26, Z^0 and $W^\pm$ decay angular distributions are correctly taken into account.

8.5.2 Heavy Standard Model Higgs

`ISUB =`	5	$Z^0Z^0 \to h^0$
	8	$W^+W^- \to h^0$
	71	$Z^0Z^0 \to Z^0Z^0$ (longitudinal)
	72	$Z^0Z^0 \to W^+W^-$ (longitudinal)
	73	$Z^0W^+ \to Z^0W^+$ (longitudinal)
	76	$W^+W^- \to Z^0Z^0$ (longitudinal)
	77	$W^+W^\pm \to W^+W^\pm$ (longitudinal)

Processes 5 and 8 are the simple $2 \to 1$ versions of what is now available in 123 and 124 with the full $2 \to 3$ kinematics. For low Higgs masses processes 5 and 8 overestimate the correct cross sections and should not be used, whereas good agreement between the $2 \to 1$ and $2 \to 3$ descriptions is observed when heavy Higgs production is studied.

The subprocesses 5 and 8, $VV \to h^0$, which contribute to the processes $VV \to V'V'$, show a bad high-energy behaviour. Here V denotes a longitudinal intermediate gauge boson, Z^0 or $W^\pm$. This can be cured only by the inclusion of all $VV \to V'V'$ graphs, as is done in subprocesses 71, 72, 73, 76 and 77. In particular, subprocesses 5 and 8 give rise to a fictitious high-mass tail of the Higgs. If this tail is thrown away, however, the agreement between the s-channel

graphs only (subprocesses 5 and 8) and the full set of graphs (subprocesses 71 etc.) is very good: for a Higgs of nominal mass 300 (800) GeV, a cut at 600 (1200) GeV retains 95% (84%) of the total cross section, and differs from the exact calculation, cut at the same values, by only 2% (11%) (numbers for SSC energies). With this prescription there is therefore no need to use subprocesses 71 etc. rather than subprocesses 5 and 8.

For subprocess 77, there is an option, see `MSTP(45)`, to select the charge combination of the scattering W's: like-sign, opposite-sign (relevant for Higgs), or both.

Process 77 contains a divergence for $p_\perp \to 0$ due to γ-exchange contributions. This leads to an infinite total cross section, which is entirely fictitious, since the simple parton-distribution function approach to the longitudinal W flux is not appropriate in this limit. For this process, it is therefore necessary to make use of a cut, e.g. $p_\perp > m_{\mathrm{W}}$.

For subprocesses 71, 72, 76 and 77, an option is included (see `MSTP(46)`) whereby you can select only the *s*-channel Higgs graph; this will then be essentially equivalent to running subprocess 5 or 8 with the proper decay channels (i.e. $\mathrm{Z}^0\mathrm{Z}^0$ or $\mathrm{W}^+\mathrm{W}^-$) set via `MDME`. The difference is that the Breit–Wigners in subprocesses 5 and 8 contain a mass-dependent width, whereas the width in subprocesses 71–77 is calculated at the nominal Higgs mass; also, higher-order corrections to the widths are treated more accurately in subprocesses 5 and 8. Further, processes 71–77 assume the incoming W/Z to be on the mass shell, with associated kinematics factors, while processes 5 and 8 have W/Z correctly space-like. All this leads to differences in the cross sections by up to a factor of 1.5.

In the absence of a Higgs, the sector of longitudinal Z and W scattering will become strongly interacting at energies above 1 TeV. The models proposed by Dobado, Herrero and Terron [Dob91] to describe this kind of physics have been included as alternative matrix elements for subprocesses 71, 72, 73, 76 and 77, selectable by `MSTP(46)`. From the point of view of the general classification scheme for subprocesses, this kind of models should appropriately be included as separate subprocesses with numbers above 100, but the current solution allows a more efficient reuse of existing code. By a proper choice of parameters, it is also here possible to simulate the production of a techni-ρ (see subsection 8.6.7).

Currently, the scattering of transverse gauge bosons has not been included, neither that of mixed transverse–longitudinal scatterings. These are expected to be less important at high energies, and do not contain an h^0 resonance peak, but need not be entirely negligible in magnitude. As a rule of thumb, processes 71–77 should not be used for VV invariant masses below 500 GeV.

The decay products of the longitudinal gauge bosons are correctly distributed in angle.

8.5.3 Extended neutral Higgs sector

`MSEL` = 19

`ISUB` =	h^0	H^0	A^0	
	3	151	156	$\mathrm{f}_i\bar{\mathrm{f}}_i \to X$
	102	152	157	$\mathrm{gg} \to X$
	103	153	158	$\gamma\gamma \to X$
	111	183	188	$\mathrm{q}\bar{\mathrm{q}} \to \mathrm{g}X$
	112	184	189	$\mathrm{qg} \to \mathrm{q}X$
	113	185	190	$\mathrm{gg} \to \mathrm{g}X$
	24	171	176	$\mathrm{f}_i\bar{\mathrm{f}}_i \to \mathrm{Z}^0X$
	26	172	177	$\mathrm{f}_i\bar{\mathrm{f}}_j \to \mathrm{W}^+X$
	123	173	178	$\mathrm{f}_i\mathrm{f}_j \to \mathrm{f}_i\mathrm{f}_jX$ (ZZ fusion)
	124	174	179	$\mathrm{f}_i\mathrm{f}_j \to \mathrm{f}_k\mathrm{f}_lX$ ($\mathrm{W}^+\mathrm{W}^-$ fusion)
	121	181	186	$\mathrm{gg} \to \mathrm{Q}_k\overline{\mathrm{Q}}_kX$
	122	182	187	$\mathrm{q}_i\bar{\mathrm{q}}_i \to \mathrm{Q}_k\overline{\mathrm{Q}}_kX$

In PYTHIA, the particle content of a two-Higgs-doublet scenario is included: two neutral scalar particles, 25 and 35, one pseudoscalar one, 36, and a charged doublet, ± 37. (Of course, these particles may also be associated with corresponding Higgs states in larger multiplets.) By convention, we choose to call the lighter scalar Higgs h^0 and the heavier H^0. The pseudoscalar is called A^0 and the charged $\mathrm{H}^\pm$. Charged-Higgs production is covered in section 8.5.4.

A number of h^0 processes have been duplicated for H^0 and A^0. The correspondence between `ISUB` numbers is shown in the table above: the first column of `ISUB` numbers corresponds to $X = \mathrm{h}^0$, the second to $X = \mathrm{H}^0$, and the third to $X = \mathrm{A}^0$. Note that several of these processes are not expected to take place at all, owing to vanishing Born term couplings. We have still included them for flexibility in simulating arbitrary couplings at the Born or loop level, or for the case of mixing between the scalar and pseudoscalar sectors.

A few Standard Model Higgs processes have no correspondence in the scheme above. These include

- 5 and 8, which anyway have been superseded by 123 and 124;
- 71, 72, 73, 76 and 77, which deal with what happens if there is no light Higgs, and so is a scenario complementary to the one above, where several light Higgses are assumed; and
- 110, which is mainly of interest in Standard Model Higgs searches.

The processes 111–113, 183–185 and 188–190 have only been worked out in full detail for the Standard Model Higgs case, and not when e.g. squark loop contributions need be considered. The approximate procedure outlined in subsection 8.5.1, based on combining the kinematics shape from simple expressions in the $m_\mathrm{t} \to \infty$ limit with a normalization derived from the $\mathrm{gg} \to X$ cross section, should therefore be viewed as a first ansatz only. In particular, it is not recommended to try the non-default `MSTP(38)=0` option, which is incorrect beyond the Standard Model.

In processes 121, 122, 181, 182, 186 and 187 the recoiling heavy flavour is assumed to be top, which is the only one of interest in the Standard Model, and the one where the parton-distribution-function approach invoked in processes 3, 151 and 156 is least reliable. However, it is possible to change the quark flavour in 121 etc.; for each process `ISUB` this flavour is given by `KFPR(ISUB,2)`. This may become relevant if couplings to $\mathrm{b\bar{b}}$ states are enhanced, e.g. if $\tan\beta \gg 1$ in the MSSM. The matrix elements in this group are based on scalar Higgs couplings; differences for a pseudoscalar Higgs remains to be worked out.

By default, the h^0 has the couplings of the Standard Model Higgs, while the H^0 and A^0 have couplings set in `PARU(171) - PARU(178)` and `PARU(181) - PARU(190)`, respectively. The default values for the H^0 and A^0 have no deep physics motivation, but are set just so that the program will not crash due to the absence of any couplings whatsoever. You should therefore set the above couplings to your desired values if you want to simulate either H^0 or A^0. Also the couplings of the h^0 particle can be modified, in `PARU(161) - PARU(165)`, provided that `MSTP(4)` is set to 1.

For `MSTP(4)=2`, the mass of the h^0 (in `PMAS(25,1)`) and the $\tan\beta$ value (in `PARU(141)`) are used to derive the masses of the other Higgses, as well as all Higgs couplings. `PMAS(35,1) - PMAS(37,1)` and `PARU(161) - PARU(195)` are overwritten accordingly. The relations used are the ones of the Born-level MSSM [Gun90]. Loop corrections to those expressions have been calculated within specific supersymmetric scenarios, and are known to have a non-negligible effects on the resulting phenomenology. By switching on supersymmetry simulation and setting parameters appropriately, one will gain access to these mass formulae, see section 9.5.

Note that not all combinations of m_h and $\tan\beta$ are allowed; for `MSTP(4)=2` the requirement of a finite A^0 mass imposes the constraint

$$m_\mathrm{h} < m_\mathrm{Z} \frac{\tan^2\beta - 1}{\tan^2\beta + 1}, \tag{139}$$

or, equivalently,

$$\tan^2\beta > \frac{m_\mathrm{Z} + m_\mathrm{h}}{m_\mathrm{Z} - m_\mathrm{h}}. \tag{140}$$

If this condition is not fulfilled, the program will crash.

A more realistic approach to the Higgs mass spectrum is to include radiative corrections to the Higgs potential. Such a machinery has never been implemented as such in PYTHIA, but appears as part of the Supersymmetry framework described in subsection 8.7. At tree level, the minimal set of inputs would be `IMSS(1)=1` to switch on SUSY, `RMSS(5)` to set the $\tan\beta$ value (this overwrites the `PARU(141)` value when SUSY is switched on) and `RMSS(19)` to set A^0 mass. However, the significant radiative corrections depend on the properties of all particles that couple to the Higgs boson, and the user may want to change the default values of the relevant `RMSS` inputs. In practice, the most important are those related indirectly to the physical masses of the third generation supersymmetric quarks and the Higgsino: `RMSS(10)` to set the left–handed doublet SUSY mass parameter, `RMSS(11)` to set the right stop mass parameter, `RMSS(12)` to set the right sbottom mass parameter, `RMSS(4)` to set the Higgsino mass and a portion of the squark mixing, and `RMSS(16)` and `RMSS(17)` to set the stop and bottom trilinear couplings, respectively, which specifies the remainder of the squark mixing. From these inputs, the Higgs masses and couplings would be derived. Note that switching on SUSY also implies that Supersymmetric decays of the Higgs particles become possible if kinematically allowed. If you do not want this to happen, you may want to increase the SUSY mass parameters. (Use `CALL PYSTAT(2)` after initialization to see the list of branching ratios.)

Pair production of Higgs states may be a relevant source, see section 8.5.5 below.

Finally, heavier Higgses may decay into lighter ones, if kinematically allowed, in processes like $A^0 \to Z^0 h^0$ or $H^+ \to W^+ h^0$. Such modes are included as part of the general mixture of decay channels, but they can be enhanced if the uninteresting channels are switched off.

8.5.4 Charged Higgs sector

`MSEL` = 23

`ISUB` =	143	$f_i \bar{f}_j \to H^+$
	161	$f_i g \to f_k H^+$

A charged Higgs doublet, $H^\pm$, is included in the program. This doublet may be the one predicted in the MSSM scenario, see section 8.5.3, or in any other scenario. The $\tan\beta$ parameter, which is relevant also for charged Higgs couplings, is set via `PARU(141)` or, if `Susy` is switched on, via `RMSS(5)`.

The basic subprocess for charged Higgs production in hadron colliders is `ISUB` = 143. However, this process is dominated by $t\bar{b} \to H^+$, and so depends on the choice of t parton distribution, if at all present. A better representation is provided by subprocess 161, $fg \to f' H^+$; i.e. actually $\bar{b}g \to \bar{t}H^+$. It is therefore recommended to use 161 and not 143; to use both would be double-counting.

Pair production of Higgs states may be a relevant source, see section 8.5.5 below.

A major potential source of charged Higgs production is top decay. It is possible to switch on the decay channel $t \to bH^+$. Top will then decay to H^+ a fraction of the time, whichever way it is produced. The branching ratio is automatically calculated, based on the $\tan\beta$ value and masses. It is possible to only have the H^+ decay mode switched on, in which case the cross section is reduced accordingly.

8.5.5 Higgs pairs

`ISUB` =	(141)	$f_i \bar{f}_i \to \gamma/Z^0/Z'^0$
	297	$f_i \bar{f}_j \to H^\pm h^0$
	298	$f_i \bar{f}_j \to H^\pm H^0$
	299	$f_i \bar{f}_i \to A h^0$
	300	$f_i \bar{f}_i \to A H^0$
	301	$f_i \bar{f}_i \to H^+ H^-$

The subprocesses 297–301 give the production of a pair of Higgses via the *s*-channel exchange of a γ^*/Z^0 or a $W^\pm$ state.

Note that Higgs pair production is still possible through subprocess 141, as part of the decay of a generic combination of $\gamma^*/Z^0/Z'^0$. Thus it can be used to simulate $Z^0 \to h^0A^0$ and $Z^0 \to H^0A^0$ for associated neutral Higgs production. The fact that we here make use of the Z'^0 can easily be discounted, either by letting the relevant couplings vanish, or by the option `MSTP(44)=4`.

Similarly the decay $\gamma^*/Z^0/Z'^0 \to H^+H^-$ allows the production of a pair of charged Higgs particles. This process is especially important in e^+e^- colliders. The coupling of the γ^* to H^+H^- is determined by the charge alone (neglecting loop effects), while the Z^0 coupling is regulated by `PARU(142)`, and that of the Z'^0 by `PARU(143)`. The Z'^0 piece can be switched off, e.g. by `MSTP(44)=4`. An ordinary Z^0, i.e. particle code 23, cannot be made to decay into a Higgs pair, however.

The advantage of the explicit pair production processes is the correct implementation of the pair threshold.

8.6 Non-Standard Physics

The number of possible non-Standard Model scenarios is essentially infinite, but many of the studied scenarios still share a lot of aspects. For instance, new W' and Z' gauge bosons can arise in a number of different ways. Therefore it still makes sense to try to cover a few basic classes of particles, with enough freedom in couplings that many kinds of detailed scenarios can be accommodated by suitable parameter choices. We have already seen one example of this, in the extended Higgs sector above. In this section a few other kinds of non-standard generic physics are discussed. Supersymmetry is covered separately in the following section, since it is such a large sector by itself.

8.6.1 Fourth-generation fermions

`MSEL` = 7, 8, 37, 38

`ISUB` =	1	$f_i\bar{f}_i \to \gamma^*/Z^0$
	2	$f_i\bar{f}_j \to W^+$
	81	$q_i\bar{q}_i \to Q_k\overline{Q}_k$
	82	$gg \to Q_k\overline{Q}_k$
	83	$q_if_j \to Q_kf_l$
	84	$g\gamma \to Q_k\overline{Q}_k$
	85	$\gamma\gamma \to F_k\overline{F}_k$
	141	$f_i\bar{f}_i \to \gamma/Z^0/Z'^0$
	142	$f_i\bar{f}_j \to W'^+$

The prospects of a fourth generation currently seem rather dim, but the appropriate flavour content is still found in the program. In fact, the fourth generation is included on an equal basis with the first three, provided `MSTP(1)=4`. Also processes other than the ones above can therefore be used, e.g. all other processes with gauge bosons, including non-standard ones such as the Z'^0. We therefore do not repeat the descriptions found elsewhere, e.g. how to set only the desired flavour in processes 81–85. Note that it may be convenient to set `CKIN(1)` and other cuts such that the mass of produced gauge bosons is enough for the wanted particle production — in principle the program will cope even without that, but possibly at the expense of very slow execution.

8.6.2 New gauge bosons

`MSEL` = 21, 22, 24

`ISUB` = 141 $f_i\bar{f}_i \to \gamma/Z^0/Z'^0$

142 $f_i\bar{f}_j \to W'^+$

144 $f_i\bar{f}_j \to R$

The Z'^0 of subprocess 141 contains the full $\gamma^*/Z^0/Z'^0$ interference structure for couplings to fermion pairs. With `MSTP(44)` it is possible to pick only a subset, e.g. only the pure Z'^0 piece. The couplings of the Z'^0 to quarks and leptons in the first generation can be set via `PARU(121) - PARU(128)`, in the second via `PARJ(180) - PARJ(187)` and in the third via `PARJ(188) - PARJ(195)`. The eight numbers correspond to the vector and axial couplings of down-type quarks, up-type quarks, leptons and neutrinos, respectively. The default corresponds to the same couplings as that of the Standard Model Z^0, with axial couplings $a_f = \pm 1$ and vector couplings $v_f = a_f - 4e_f \sin^2\theta_W$. This implies a resonance width that increases linearly with the mass. By a suitable choice of the parameters, it is possible to simulate just about any imaginable Z'^0 scenario, with full interference effects in cross sections and decay angular distributions. Note that also the possibility of a generation dependence has been included for the Z'^0, which is normally not the case.

The coupling to the decay channel $Z'^0 \to W^+W^-$ is regulated by `PARU(129) - PARU(130)`. The former gives the strength of the coupling, which determines the rate. The default, `PARU(129)=1.`, corresponds to the 'extended gauge model' of [Alt89], wherein the $Z^0 \to W^+W^-$ coupling is used, scaled down by a factor $m_W^2/m_{Z'}^2$, to give a Z'^0 partial width into this channel that again increases linearly. If this factor is cancelled, by having `PARU(129)` proportional to $m_{Z'}^2/m_W^2$, one obtains a partial width that goes like the fifth power of the Z'^0 mass, the 'reference model' of [Alt89]. In the decay angular distribution one could imagine a much richer structure than is given by the one parameter `PARU(130)`.

Other decay modes include $Z'^0 \to Z^0h^0$, predicted in left–right symmetric models (see `PARU(145)` and ref. [Coc91]), and a number of other Higgs decay channels, see sections 8.5.3 and 8.5.4.

The $W'^\pm$ of subprocess 142 so far does not contain interference with the Standard Model $W^\pm$ — in practice this should not be a major limitation. The couplings of the W' to quarks and leptons are set via `PARU(131) - PARU(134)`. Again one may set vector and axial couplings freely, separately for the $q\bar{q}'$ and the $\ell\nu_\ell$ decay channels. The defaults correspond to the $V - A$ structure of the Standard Model W, but can be changed to simulate a wide selection of models. One possible limitation is that the same Cabibbo–Kobayashi–Maskawa quark mixing matrix is assumed as for the standard W.

The coupling $W' \to Z^0W$ can be set via `PARU(135) - PARU(136)`. Further comments on this channel as for Z'; in particular, default couplings again agree with the 'extended gauge model' of [Alt89]. A $W' \to Wh^0$ channel is also included, in analogy with the $Z'^0 \to Z^0h^0$ one, see `PARU(146)`.

The R boson (particle code 41) of subprocess 144 represents one possible scenario for a horizontal gauge boson, i.e. a gauge boson that couples between the generations, inducing processes like $s\bar{d} \to R^0 \to \mu^-e^+$. Experimental limits on flavour-changing neutral currents forces such a boson to be fairly heavy. The model implemented is the one described in [Ben85a].

A further example of new gauge groups follows right after this.

8.6.3 Left–Right Symmetry and Doubly Charged Higgses

ISUB = 341 $\ell_i\ell_j \to \mathrm{H}_L^{\pm\pm}$
342 $\ell_i\ell_j \to \mathrm{H}_R^{\pm\pm}$
343 $\ell_i\gamma \to \mathrm{H}_L^{\pm\pm}\mathrm{e}^{\mp}$
344 $\ell_i\gamma \to \mathrm{H}_R^{\pm\pm}\mathrm{e}^{\mp}$
345 $\ell_i\gamma \to \mathrm{H}_L^{\pm\pm}\mu^{\mp}$
346 $\ell_i\gamma \to \mathrm{H}_R^{\pm\pm}\mu^{\mp}$
347 $\ell_i\gamma \to \mathrm{H}_L^{\pm\pm}\tau^{\mp}$
348 $\ell_i\gamma \to \mathrm{H}_R^{\pm\pm}\tau^{\mp}$
349 $\mathrm{f}_i\bar{\mathrm{f}}_i \to \mathrm{H}_L^{++}\mathrm{H}_L^{--}$
350 $\mathrm{f}_i\bar{\mathrm{f}}_i \to \mathrm{H}_R^{++}\mathrm{H}_R^{--}$
351 $\mathrm{f}_i\mathrm{f}_j \to \mathrm{f}_k f_l\mathrm{H}_L^{\pm\pm}$ (WW fusion)
352 $\mathrm{f}_i\mathrm{f}_j \to \mathrm{f}_k f_l\mathrm{H}_R^{\pm\pm}$ (WW fusion)
353 $\mathrm{f}_i\bar{\mathrm{f}}_i \to \mathrm{Z}_R^0$
354 $\mathrm{f}_i\bar{\mathrm{f}}_i \to \mathrm{W}_R^{\pm}$

At current energies, the world is lefthanded, i.e. the Standard Model contains an $\mathbf{SU(2)}_L$ group. Left–right symmetry at some larger scale implies the need for an $\mathbf{SU(2)}_R$ group. Thus the particle content is expanded by righthanded Z_R^0 and $\mathrm{W}_R^{\pm}$ and righthanded neutrinos. The Higgs fields have to be in a triplet representation, leading to doubly-charged Higgs particles, one set for each of the two **SU(2)** groups. Also the number of neutral and singly-charged Higgs states is increased relative to the Standard Model, but a search for the lowest-lying states of this kind is no different from e.g. the freedom already accorded by the MSSM Higgs scenarios.

PYTHIA implements the scenario of [Hui97]. The expanded particle content with default masses is:

KF	name	m (GeV)
9900012	ν_{Re}	500
9900014	$\nu_{R\mu}$	500
9900016	$\nu_{R\tau}$	500
9900023	Z_R^0	1200
9900024	W_R^+	750
9900041	H_L^{++}	200
9900042	H_R^{++}	200

The main decay modes implemented are
$\mathrm{H}_L^{++} \to \mathrm{W}_L^+\mathrm{W}_L^+, \ell_i^+\ell_j^+$ (i, j generation indices); and
$\mathrm{H}_R^{++} \to \mathrm{W}_R^+\mathrm{W}_R^+, \ell_i^+\ell_j^+$.
The physics parameters of the scenario are found in `PARP(181) - PARP(192)`.

The $W_R^{\pm}$ has been implemented as a simple copy of the ordinary $\mathrm{W}^{\pm}$, with the exception that it couple to righthanded neutrinos instead of the ordinary lefthanded ones. Thus the standard CKM matrix is used in the quark sector, and the same vector and axial coupling strengths, leaving only the mass as free parameter. The Z_R^0 implementation (without interference with γ or the ordinary Z^0) allows decays both to left- and righthanded neutrinos, as well as other fermions, according to one specific model ansatz [Fer00]. Obviously both the $W_R^{\pm}$ and the Z_R^0 descriptions are likely to be simplifications, but provide a starting point.

The righthanded neutrinos can be allowed to decay further [Riz81, Fer00]. Assuming them to have a mass below that of W_R^+, they decay to three-body states via a virtual W_R^+, $\nu_{R\ell} \to \ell^+\mathrm{f}\bar{\mathrm{f}}'$ and $\nu_{R\ell} \to \ell^-\bar{\mathrm{f}}\mathrm{f}'$, where both choices are allowed owing to the Majorana character of the neutrinos. If there is a significant mass splitting, also sequential decays $\nu_{R\ell} \to \ell^{\pm}\ell'^{\mp}\nu'_{R\ell}$ are allowed. Currently the decays are isotropic in phase space. If the neutrino masses are close to or above the W_R ones, this description has to be substituted by a sequential decay via a real W_R

(not implemented, but actually simpler to do than the one here).

8.6.4 Leptoquarks

MSEL = 25

ISUB = 145 $q_i \ell_j \to L_Q$

162 $qg \to \ell L_Q$

163 $gg \to L_Q \overline{L}_Q$

164 $q_i \overline{q}_i \to L_Q \overline{L}_Q$

Several processes that can generate a leptoquark have been included. Currently only one leptoquark has been implemented, as particle 42, denoted L_Q. The leptoquark is assumed to carry specific quark and lepton quantum numbers, by default u quark plus electron. These flavour numbers are conserved, i.e. a process such as $ue^- \to L_Q \to d\nu_e$ is not allowed. This may be a bit restrictive, but it represents one of many leptoquark possibilities. The spin of the leptoquark is assumed to be zero, i.e. its decay is isotropical.

Although only one leptoquark is implemented, its flavours may be changed arbitrarily to study the different possibilities. The flavours of the leptoquark are defined by the quark and lepton flavours in the decay mode list. Since only one decay channel is allowed, this means that the quark flavour is stored in KFDP(MDCY(42,2),1) and the lepton one in KFDP(MDCY(42,2),2). The former must always be a quark, while the latter could be a lepton or an antilepton; a charge-conjugate partner is automatically defined by the program. At initialization, the charge is recalculated as a function of the flavours defined; also the leptoquark name is redefined to be of the type 'LQ_(q)(l)', where actual quark (q) and lepton (l) flavours are displayed.

The $L_Q \to q\ell$ vertex contains an undetermined Yukawa coupling strength, which affects both the width of the leptoquark and the cross section for many of the production graphs. This strength may be changed in PARU(151). The definition of PARU(151) corresponds to the k factor of [Hew88], i.e. to $\lambda^2/(4\pi\alpha_{em})$, where λ is the Yukawa coupling strength of [Wud86]. Note that PARU(151) is thus quadratic in the coupling.

The leptoquark is likely to be fairly long-lived, in which case it has time to fragment into a mesonic- or baryonic-type state, which would decay later on. This is a bit tedious to handle; therefore the leptoquark is always assumed to decay before fragmentation. This may give some imperfections in the event generation, but should not be off by much in the final analysis [Fri97].

Inside the program, the leptoquark is treated as a resonance. Since it carries colour, some extra care is required. In particular, it is not allowed to put the leptoquark stable, by modifying either MDCY(42,1) or MSTP(41): then the leptoquark would be handed undecayed to PYTHIA, which would try to fragment it (as it does with any other coloured object), and most likely crash.

8.6.5 Compositeness and anomalous couplings

ISUB = 11 $f_i f_j \to f_i f_j$ (QCD)

12 $f_i \overline{f}_i \to f_k \overline{f}_k$

20 $f_i \overline{f}_j \to \gamma W^+$

165 $f_i \overline{f}_i \to f_k \overline{f}_k$ (via γ^*/Z^0)

166 $f_i \overline{f}_j \to f_k \overline{f}_l$ (via $W^\pm$)

Some processes have been set up to allow anomalous coupling to be introduced, in addition to the Standard Model ones. These can be switched on by MSTP(5)≥ 1; the default MSTP(5)=0 corresponds to the Standard Model behaviour.

In processes 11 and 12, the quark substructure is included in the left–left isoscalar model [Eic84, Chi90] for MSTP(5)=1, with compositeness scale Λ given in PARU(155) (default 1000 GeV) and sign η of interference term in PARU(156) (default $+1$; only other alternative -1). The above model assumes that only u and d quarks are composite (at least at the

scale studied); with MSTP(5)=2 compositeness terms are included in the interactions between all quarks.

The processes 165 and 166 are basically equivalent to 1 and 2, i.e. γ^*/Z^0 and $W^\pm$ exchange, respectively, but a bit less fancy (no mass-dependent width etc.). The reason for this duplication is that the resonance treatment formalism of processes 1 and 2 could not easily be extended to include other than *s*-channel graphs. In processes 165 and 166, only one final-state flavour is generated at the time; this flavour should be set in KFPR(165,1) and KFPR(166,1), respectively. For process 166 one gives the down-type flavour, and the program will associate the up-type flavour of the same generation. Defaults are 11 in both cases, i.e. e^+e^- and $e^+\nu_e$ ($e^-\overline{\nu}_e$) final states. While MSTP(5)=0 gives the Standard Model results, MSTP(5)=1 contains the left–left isoscalar model (which does not affect process 166), and MSTP(5)=3 the helicity-non-conserving model (which affects both) [Eic84, Lan91]. Both models above assume that only u and d quarks are composite; with MSTP(5)= 2 or 4, respectively, contact terms are included for all quarks in the initial state. Parameters are PARU(155) and PARU(156), as above.

Note that processes 165 and 166 are book-kept as $2 \to 2$ processes, while 1 and 2 are $2 \to 1$ ones. This means that the default Q^2 scale in parton distributions is $p_\perp^2$ for the former and $\hat{s}$ for the latter. To make contact between the two, it is recommended to set MSTP(32)=4, so as to use $\hat{s}$ as scale also for processes 165 and 166.

In process 20, for $W\gamma$ pair production, it is possible to set an anomalous magnetic moment for the W in PARU(153) ($= \eta = \kappa - 1$; where $\kappa = 1$ is the Standard Model value). The production process is affected according to the formulae of [Sam91], while W decay currently remains unaffected. It is necessary to set MSTP(5)=1 to enable this extension.

8.6.6 Excited fermions

ISUB =	146	$e\gamma \to e^*$
	147	$dg \to d^*$
	148	$ug \to u^*$
	167	$q_i q_j \to q_k d^*$
	168	$q_i q_j \to q_k u^*$
	169	$q_i \overline{q}_i \to e^\pm e^{*\mp}$

Compositeness scenarios may also give rise to sharp resonances of excited quarks and leptons. An excited copy of the first generation is implemented, consisting of spin 1/2 particles d^* (code 4000001), u^* (4000002), e^* (4000011) and ν_e^* (4000012).

The current implementation contains gauge interaction production by quark–gluon fusion (processes 147 and 148) or lepton–photon fusion (process 146) and contact interaction production by quark–quark or quark–antiquark scattering (processes 167–169) . The couplings f, f' and f_s to the **SU(2)**, **U(1)** and **SU(3)** groups are stored in PARU(157) - PARU(159), the scale parameter Λ in PARU(155); you are also expected to change the f^* masses in accordance with what is desired — see [Bau90] for details on conventions. Decay processes are of the types $q^* \to qg$, $q^* \to q\gamma$, $q^* \to qZ^0$ or $q^* \to q'W^\pm$, with the latter three (two) available also for e^* (ν_e^*). A non-trivial angular dependence is included in the q^* decay for processes 146–148, but has not been included for processes 167–169.

8.6.7 Technicolor

MSEL = 50

ISUB =	149	$gg \to \eta_{tc}$
	191	$f_i\bar{f}_i \to \rho^0_{tc}$
	192	$f_i\bar{f}_j \to \rho^+_{tc}$
	193	$f_i\bar{f}_i \to \omega^0_{tc}$
	194	$f_i\bar{f}_i \to f_k\bar{f}_k$
	195	$f_i\bar{f}_j \to f_k\bar{f}_l$
	361	$f_i\bar{f}_i \to W^+_L W^-_L$
	362	$f_i\bar{f}_i \to W^\pm_L \pi^\mp_{tc}$
	363	$f_i\bar{f}_i \to \pi^+_{tc}\pi^-_{tc}$
	364	$f_i\bar{f}_i \to \gamma\pi^0_{tc}$
	365	$f_i\bar{f}_i \to \gamma\pi'^0_{tc}$
	366	$f_i\bar{f}_i \to Z^0\pi^0_{tc}$
	367	$f_i\bar{f}_i \to Z^0\pi'^0_{tc}$
	368	$f_i\bar{f}_i \to W^\pm\pi^\mp_{tc}$
	370	$f_i\bar{f}_j \to W^\pm_L Z^0_L$
	371	$f_i\bar{f}_j \to W^\pm_L \pi^0_{tc}$
	372	$f_i\bar{f}_j \to \pi^\pm_{tc} Z^0_L$
	373	$f_i\bar{f}_j \to \pi^\pm_{tc}\pi^0_{tc}$
	374	$f_i\bar{f}_j \to \gamma\pi^\pm_{tc}$
	375	$f_i\bar{f}_j \to Z^0\pi^\pm_{tc}$
	376	$f_i\bar{f}_j \to W^\pm\pi^0_{tc}$
	377	$f_i\bar{f}_j \to W^\pm\pi'^0_{tc}$

The technicolor (TC) scenario offers an alternative to the ordinary Higgs mechanism for giving masses to the W and Z, by using strong dynamics for the electroweak symmetry breaking. The technicolor gauge group is an analogue of QCD, with a rich spectrum of technimesons made out of techniquarks. An effective Lagrangian can be derived for the lightest resonances. In TC, the breaking of a chiral symmetry in the new, strongly interacting gauge theory generates the Goldstone bosons necessary for electroweak symmetry breaking. Thus three of the technipions assume the rôle of the longitudinal components of the W and Z bosons, but many other states remain as separate particles: technipions (π_{tc}), technirhos (ρ_{tc}), techniomegas (ω_{tc}), etc. The mass hierarchies, however, are unlike QCD because of the behaviour of the gauge couplings in realistic models of extended TC (ETC). The difficulties of ETC in explaining the top quark mass while suppressing FCNC's is circumvented by the addition of topcolor interactions, which provide the bulk of m_t.

No fully realistic model has been found so far, however, so any phenomenology has to be taken as indicative only. As experimental constraints have accumulated, also the sophistication of model building has had to follow suit. The processes represented here correspond to several generations of development. Thus processes 149, 191, 192 and 193 should nowadays be considered as obsolete and superseded by the other processes 194, 195 and 361–377.

In section 8.5.2 it is discussed how processes 71–77, in some of its options, can be used to simulate a scenario with techni-ρ resonances in longitudinal gauge boson scattering.

Process 149 is that of the production of a techni-η, particle code KF = 3000331. This particle has zero spin, is a singlet under electroweak **SU(2)**×**U(1)**, but carries octet colour charge. It is one of the possible techni-π particles; the name techni-η is part of a sub-classification not used by all authors.

The techni-η couples to ordinary fermions according to the fermion squared mass. The

dominant decay mode is therefore $t\bar{t}$, if allowed. The coupling to a gg state is roughly comparable with that to $b\bar{b}$. Production at hadron colliders is therefore predominantly through gg fusion, as implemented in process 149.

The two main free parameters are the techni-η mass and the decay constant F_π. The latter appears inversely quadratically in all the partial widths. Also the total cross section is affected, since the cross section is proportional to the gg partial width. F_π is stored in `PARP(46)` and has the default value 123 GeV, which is the number predicted in some models.

All the other processes refer to ETC models, where hard mass contributions to technipion masses make decays like $\rho_{tc} \to \pi_{tc}\pi_{tc}$ kinematically inaccessible. Instead, decays like $\rho_{tc}^{ew} \to \pi_{tc}^{ew} W_L$, for example, dominate, where ew denotes constituent technifermions with only electroweak quantum numbers and W_L is a longitudinal W bosons. As a result, the ew technirho and techniomega tend to have small total widths.

Effective couplings are derived in the valence technifermion approximation, and the techniparticle decays can be calculated directly [Eic96, Lan99]. Technirhos and techniomegas are produced through kinematic mixing with gauge bosons, leading to final states containing Standard Model particles and/or pseudo-Goldstone bosons (technipions).

As an additional possibility, $SU_c(3)$ non-singlet states are included along with the coloron of topcolor assisted technicolor. In this case, coloured technirhos (and the coloron) can have substantial total widths and enhanced couplings to bottom and top quarks.

The expanded particle content with default masses is:

KF	name	m (GeV)
3000111	π_{tc}^0	110
3000211	π_{tc}^+	110
3000221	π'^0_{tc}	110
3000113	ρ_{tc}^0	210
3000213	ρ_{tc}^+	210
3000223	ω_{tc}^0	210

The ρ_{tc} and ω_{tc}^0 masses are not pole masses. Parameters regulating production and decay rates are stored in `PARP(137) - PARP(150)`.

Processes 191, 192 and 193 are based on *s*-channel production of the respective resonance [Eic96]. All decay modes implemented can be simulated separately or in combination, in the standard fashion. These include pairs of fermions, of gauge bosons, of technipions, and of mixtures of gauge bosons and technipions

Process 194 is intended to more accurately represent the mixing between the γ^*, Z^0, ρ_{tc}^0 and ω_{tc}^0 particles in the Drell-Yan process [Lan99]. Process 195 is the analogous charged channel process including $W^\pm$ and $\rho_{tc}^\pm$ mixing. By default, the final state fermions are e^+e^- and $e^\pm\nu_e$, respectively. These can be changed through the parameters `KFPR(194,1)` and `KFPR(195,1)`, respectively (where the `KFPR` value should represent a charged fermion).

Processes 361–368 give the pair production of technipions and gauge bosons by $\rho_{tc}^0/\omega_{tc}^0$ exchange, with processes 370–377 corresponding pair production by $\rho_{tc}^\pm$ exchange. It is worth mentioning that the decay products of the W and Z bosons are distributed according to phase space, regardless of their designation as longitudinal W_L/Z_L or ordinary transverse gauge bosons. The exact meaning of longitudinal or transverse polarizations in this case requires more thought from the side of the model authors.

The possibility of coloured technihadrons has been included, specifically for the case of Topcolor assisted Technicolor. The main effects of these particles are indirect, in that they modify the underlying two-parton QCD processes much like compositeness terms, except that resonances are visible. More details will be forthcoming. The parameter dependence of the 'model' is included in the tangent of the mixing angle `PARP(155)` and a mass parameter `PARP(156)`. Similar to compositeness, the effects of these colored technihadrons are simulated by setting `MSTP(5)=5`.

8.6.8 Extra Dimensions

ISUB = 391 $f\bar{f} \to G^*$
392 $gg \to G^*$
393 $q\bar{q} \to gG^*$
394 $qg \to qG^*$
395 $gg \to gG^*$

In recent years, the area of observable consequences of extra dimensions has attracted a strong interest. The field is still in rapid development, so there still does not exist a 'standard phenomenology'. The topic is also fairly new in PYTHIA, and currently only a first scenario is available.

The G^*, temporarily introduced as new particle code 41, is intended to represent the lowest excited graviton state in a Randall-Sundrum scenario [Ran99] of extra dimensions. The lowest-order production processes by fermion or gluon fusion are found in 391 and 392. The further processes 393–395 are intended for the high-$p_\perp$ tail in hadron colliders. As usual, it would be double-counting to have both sets of processes switched on at the same time. Processes 391 and 392, with initial-state showers switched on, are appropriate for the full cross section at all $p_\perp$ values, and gives a reasonable description also of the high-$p_\perp$ tail. Processes 393–395 could be useful e.g. for the study of invisible decays of the G^*, where a large $p_\perp$ imbalance would be required. It also serves to test/confirm the shower expectations of different $p_\perp$ spectra for different production processes [Bij01].

Decay channels of the G^* to $f\bar{f}$, gg, $\gamma\gamma$, Z^0Z^0 and W^+W^- contribute to the total width. The correct angular distributions are included for decays to a fermion pair in the lowest-order processes, whereas other decays currently are taken to be isotropic.

The G^* mass is to be considered a free parameter. The other degree of freedom in this scenario is a dimensionless coupling, see PARP(50).

8.7 Supersymmetry

MSEL = 39–45
ISUB = 201–296 (see tables at the beginning of the chapter)

PYTHIA simulates the Minimal Supersymmetric Standard Model (MSSM), based on an effective Lagrangian of softly-broken SUSY with parameters defined at the weak scale, which is typically between m_Z and 1 TeV. The MSSM particle spectrum is minimal in the sense that it includes only the partners of all Standard Model particles (presently without massive neutrinos), two Higgs doublets with partners, and the gravitino.

Each SM fermion has a scalar partner with the same quantum numbers. For counting purposes, the fermion states ψ_L and ψ_R are separate particles with scalar partners ϕ_L and ϕ_R. Each SM gauge boson has a fermion partner with the same quantum numbers. The Higgs sector is extended to two complex scalar doublets — one Higgs H_u which couples only to up-type fermions and one H_d which couples only to down-type fermions — leading to 3 additional physical Higgs bosons H^0, A^0 and $H^\pm$ to complement the SM Higgs h^0. Fermion higgsino partners are added for the two scalar doublets. Additionally, a light gravitino $\tilde{G}$ is included to allow studies of gauge mediated dynamical SUSY breaking [Din96].

Because of the large Yukawa couplings and the running masses for coloured particles, the interaction and mass eigenstates for the third generation sfermions can be significantly mixed. We denote the top and bottom squark mass eigenstates $\tilde{t}_1$,$\tilde{t}_2$,$\tilde{b}_1$, and $\tilde{b}_2$ to distinguish them from the nearly degenerate eigenstates for the lighter squarks. In addition, the tau slepton mass eigenstates are $\tilde{\tau}_1$ and $\tilde{\tau}_2$. The subscript 1 or 2 refers to the lightest and heaviest state respectively. In SUGRA inspired models, in the absence of mixing so that interaction eigenstates are the same as mass eigenstates, the right eigenstate is lighter than the left. In this case, for example, $\tilde{b}_1 = \tilde{b}_R$ and $\tilde{b}_2 = \tilde{b}_L$. For completeness, we include the sterile $\tilde{\nu}_R$ particles. The mixing of the partners

to the electroweak gauge bosons (gauginos) and the two Higgs doublets (higgsinos) lead to the mass eigenstates for neutralino χ_i^0 and chargino $\tilde{\chi}_i^{\pm}$ particles.

The particle partners and KF codes are listed in Table 13. Note that, at times, antiparticles of scalar particles are denoted by *, i.e. $\tilde{t}^*$ rather than the more correct but cumbersome $\bar{\tilde{t}}$ or $\tilde{t}$.

Once the parameters of the softly-broken SUSY Lagrangian are specified, the interactions are fixed, and the sparticle masses can be calculated [Hab85]. The masses of the scalar partners to fermions, sfermions, depend on soft scalar masses, trilinear couplings, the Higgsino mass μ, and $\tan\beta$, the ratio of Higgs vacuum expectation values $\langle H_u \rangle / \langle H_d \rangle$. The masses of the fermion partners to the gauge and Higgs bosons, the neutralinos and charginos, depend on soft gaugino masses, μ, and $\tan\beta$. Finally, the properties of the Higgs scalar sector is calculated from the input pseudoscalar Higgs boson mass m_A, $\tan\beta$, μ, trilinear couplings and the sparticle properties in an effective potential approach [Car95]. Of course, these calculations also depend on SM parameters (m_t, m_Z, α_s, etc.). Any modifications to these quantities from virtual MSSM effects are not taken into account. In principle, the sparticle masses also acquire loop corrections that depend on all MSSM masses.

Several models exist for deriving the rich set of directly measurable mass and mixing parameters from the assumed soft SUSY breaking scenario with a much smaller set of free parameters. One example is Supergravity (SUGRA) inspired models, where the number of free parameters is reduced by imposing universality and exploiting the apparent unification of gauge couplings. Five parameters fixed at the gauge coupling unification scale, $\tan\beta, M_0, m_{1/2}, A_0$, and $sign(\mu)$, are then related to the mass parameters at the scale of EWSB by renormalization group equations (see e.g. [Pie97]). ISASUSY/ISAJET [Bae93] and SUSYGEN [Kat98] numerically solve these equations to determine the mass parameters. Alternatively, they can input a general set of parameters. PYTHIA operates in the second manner, with a slightly more general set of input parameters. There are three reasons for this: (1) programs already exist to calculate the SUGRA inspired mass parameters, (2) approximate analytic formulae [Dre95] also exist which reproduce the output of ISASUSY within $\simeq 10\%$, and (3) we desire to study a much richer phenomenology than that possible in SUGRA inspired models. The PYTHIA input parameters are described in detail later, in section 9.5.

R-parity conservation is assumed (at least on the time and distance scale of a typical collider experiment), and only lowest order, sparticle pair production processes are included. Only those processes with $e^+e^-, \mu^+\mu^-$, or quark and gluon initial states are simulated. Tables 21, 22 and 23 list available SUSY processes. In processes 210 and 213, $\tilde{\ell}$ refers to both $\tilde{e}$ and $\tilde{\mu}$. For ease of readability, we have removed the subscript L on $\tilde{\nu}$. $\tilde{t}_i\tilde{t}_i^*, \tilde{\tau}_i\tilde{\tau}_j^*$ and $\tilde{\tau}_i\tilde{\nu}_\tau^*$ production correctly account for sfermion mixing. Several processes are conspicuously absent from the table. For example, processes 255 and 257 would simulate the associated production of right handed squarks with charginos. Since the right handed squark only couples to the higgsino component of the chargino, the interaction strength is proportional to the quark mass, so these processes can be ignored.

Likewise, only *R*-parity conserving decays are allowed, so that one sparticle is stable, either the lightest neutralino, the gravitino, or a sneutrino. SUSY decays of the top quark are included, but all other SM particle decays are unaltered.

The decays of superpartners are calculated using the formulae of refs. [Gun88, Bar86a, Bar86b, Bar95]. All decays are spin averaged. For simplicity, the kinematics of three body decays of neutralinos, charginos, and the gluino are sampled using only the phase space weight. Decays involving $\tilde{b}$ and $\tilde{t}$ use the formulae of [Bar95], so they are valid for large values of $\tan\beta$. The one loop decays $\tilde{\chi}_j \to \tilde{\chi}_i\gamma$ and $\tilde{t} \to c\tilde{\chi}_1$ are also included.

One difference between the SUSY simulation and the other parts of the program is that it is not beforehand known which sparticles may be stable. Normally this would mean either the $\tilde{\chi}_1^0$ or the gravitino $\tilde{G}$, but in principle also other sparticles could be stable. The ones found to be stable have their MWID(KC) and MDCY(KC,1) values set zero at initialization. If several PYINIT calls are made in the same run, with different SUSY parameters, the ones set zero above are not necessarily set back to nonzero values, since most original values are not saved anywhere. As

an exception to this rule, the PYMSIN SUSY initialization routine, called by PYINIT, does save and restore the MWID(KC) and MDCY(KC,1) values of the lightest SUSY particle. It is therefore possible to combine several PYINIT calls in a single run, provided that only the lightest SUSY particle is stable. If this is not the case, MWID(KC) and MDCY(KC,1) values may have to be reset by hand, or else some particles that ought to decay will not do that.

Various improvements to the simulation are being implemented in stages. Some of these can have a significant impact on the collider phenomenology. Among these are: the generalization to complex-valued soft SUSY-breaking parameters in the neutralino and chargino sector; the same in the Higgs sector, which removes the possibility of CP-even or CP-odd labels; the calculation of neutralino and chargino decay rates which are accurate for large $\tan\beta$; and matrix element weighting of particle distributions in three-body decays.

Recently, several improvements have been introduced to the supersymmetric machinery:

- Complex phases in the neutralino, chargino, and gluino sector. The parameters M_1, M_2, M_3, and μ (specifically RMSS(1) - RMSS(4)) now refer only to the modulus of these parameters. Phases are specified by the parameters RMSS(30) - RMSS(33), respectively. Chargino and neutralino masses and mixing angles now differ from the case with vanishing phases, and the production cross sections and decay rates including pairs of neutralinos and charginos have been modified accordingly. Gluino decays to neutralinos and charginos are also affected (but the effect on gluino + neutralino/chargino production has not yet been simulated).
  The PYSSMT commonblock has been expanded to
  ```
  COMMON/PYSSMT/ZMIX(4,4),UMIX(2,2),VMIX(2,2),SMZ(4),SMW(2),
 &SFMIX(16,4),ZMIXI(4,4),UMIXI(2,2),VMIXI(2,2)
  ```
  where ZMIX, UMIX, VMIX give the real and ZMIXI, UMIXI, VMIXI the imaginary parts of the complex mixing matrices.
- Matrix Element weighting in 3-body sparticle decays. This replaces the older approach based purely on phase space. The matrix element includes the dependence on complex phases.
- Large $\tan\beta$ corrections to Higgs boson properties have been included. The array values RMSS(40) and RMSS(41) are used for temporary storage of the corrections Δm_t and Δm_b. The updated version of SubHpole, written by Carena et al, also includes some bug fixes, so that it is generally better behaved.
- More flexibility in the treatment of stops, sbottoms and staus. With the flag IMSS(5)=1, the properties of the third generation sparticles can be specified by their mixing angle and mass eigenvalues (instead of being derived from the soft SUSY-breaking parameters). The parameters RMSS(26) - RMSS(28) specify the mixing angle (in radians) for the sbottom, stop, and stau. The parameters RMSS(10) - RMSS(14) specify the two stop masses, the one sbottom mass (the other being fixed by the other parameters) and the two stau masses. Note that the masses RMSS(10), RMSS(11) and RMSS(13) correspond to the left-left entries of the diagonalized matrices, while RMSS(12) and RMSS(14) correspond to the right-right entries. Note that these entries need not be ordered in mass. Also, one technical change has arisen with the stau sector. Previously, it was assumed that the soft SUSY-breaking parameters associated with the stau included D-terms. This is no longer the case, and is more consistent with the treatment of the stop and sbottom.

8.7.1 SUSY examples

The SUSY routines and commonblock variables are described in section 9.5. To illustrate the usage of the switches and parameters, we give three simple examples.

Example 1: Light Stop

The first example is an MSSM model with a light neutralino $\tilde{\chi}_1$ and a light stop $\tilde{t}_1$, so that $t \to \tilde{t}_1\tilde{\chi}_1$ can occur. The input parameters are

IMSS(1)=1, RMSS(1)=70., RMSS(2)=70., RMSS(3)=225., RMSS(4)=-40., RMSS(5)=1.5,

`RMSS(6)=100.`, `RMSS(7)=125.`, `RMSS(8)=250.`, `RMSS(9)=250.`, `RMSS(10)=1500.`, `RMSS(11)=1500.`, `RMSS(12)=-128.`, `RMSS(13)=100.`, `RMSS(14)=125.`, `RMSS(15)=800.`, `RMSS(16)=800.`, `RMSS(17)=0.`, and `RMSS(19)=400.0`.
The top mass is fixed at 175 GeV, `PMAS(6,1)=175.0`. The resulting model has $M_{\tilde{t}_1} = 55$ GeV and $M_{\tilde{\chi}_1} = 38$ GeV. `IMSS(1)=1` turns on the MSSM simulation. By default, there are no intrinsic relations between the gaugino masses, so $M_1 = 70$ GeV, $M_2 = 70$ GeV, and $M_3 = 225$ GeV. The pole mass of the gluino is slightly higher than the parameter M_3, and the decay $\tilde{g} \to \tilde{t}_1^* t + \tilde{t}_1 \bar{t}$ occurs almost 100% of the time.

Example 2: Approximate SUGRA
The second example is an approximate SUGRA model. The input parameters are `IMSS(1)=2`, `RMSS(1)=200.`, `RMSS(4)=1.`, `RMSS(5)=10.`, `RMSS(8)=800.`, and `RMSS(16)=0.0`.
The resulting model has $M_{\tilde{d}_L} = 901$ GeV, $M_{\tilde{u}_R} = 890$ GeV, $M_{\tilde{t}_1} = 538$ GeV, $M_{\tilde{e}_L} = 814$ GeV, $M_{\tilde{g}} = 560$ GeV, $M_{\tilde{\chi}_1} = 80$ GeV, $M_{\tilde{\chi}_1^\pm} = 151$ GeV, $M_h = 110$ GeV, and $M_A = 883$ GeV. It corresponds to the choice M_0=800 GeV, $M_{1/2}$ =200 GeV, $\tan\beta = 10$, $A_0 = 0$, and $sign(\mu) > 0$. The output is similar to an ISASUSY run, but there is not exact agreement.

Example 3: ISASUSY *Model*
The final example demonstrates how to convert the output of an ISASUSY run using the same SUGRA inputs into the PYTHIA format. This assumes that you already made an ISASUSY run, e.g. with the equivalents of the input parameters above. From the output of this run you can now extract those physical parameters that need to be handed to PYTHIA, in the above example `IMSS(1)=1`, `IMSS(3)=1`, `RMSS(1)=83.81`, `RMSS(2)=168.90`, `RMSS(3)=581.83`, `RMSS(4)=283.37`, `RMSS(5)=10.`, `RMSS(6)=813.63`, `RMSS(7)=804.87`, `RMSS(8)=917.73`, `RMSS(9)=909.89`, `RMSS(10)=772.87`, `RMSS(11)=901.52`, `RMSS(12)=588.33`, `RMSS(13)=813.63`, `RMSS(14)=804.87`, `RMSS(15)=610.54`, `RMSS(16)=422.35`, `RMSS(17)=600.`, and `RMSS(19)=858.412`.

8.7.2 Lepton number violation

The most general SUSY Lagrangian contains both lepton and baryon number violating couplings, customarily forbidden by the introduction of R parity to avoid problems with proton decay. This is not the only possible choice, but certainly the simplest. On the other hand, the violation of baryon *or* lepton number is allowed without resulting in proton decay, and both give rise to very distinct phenomenologies, most notably the consequences of a decaying LSP and the removal of the cosmological constraint that the LSP should be neutral.

In the current version of PYTHIA, decays of supersymmetric particles to SM particles via two different lepton number violating couplings can be invoked (Details about the implementation and tests can be found in [Ska01], to which references concering L-violation in PYTHIA can be made). These couplings, λ and λ', correspond to the two lepton number violating terms possible in the most general supersymmetric Lagrangian, usually denoted LLE and LQD respectively. They are Yuakawa-type trilinear couplings carrying three generation indices, i, j, and k. The LLE coupling (λ_{ijk}) is antisymmetric in i and j whereas no similar constraint applies to the LQD coupling (λ'_{ijk}).

Complete matrix elements for all two-body sfermion and three-body neutralino and chargino decays are included (as given in [Dre00]). Lepton number violating decays of the gluino are not yet implemented. This should not lead to a large source of error since the gluino, being heavy in most scenarios, often has a number of other, unsuppressed decay modes available, keeping the probability that it branches into an L-violating mode small as long as the L-violating couplings are small compared with the gauge couplings.

The existence of R-odd couplings also allows for single sparticle production, i.e. there is no requirement that SUSY particles should be produced in pairs. These production cross sections are likewise not yet included in the program. For low-mass sparticles, the associated error is estimated to be negligible, as long as the L-violating couplings are smaller than the gauge

couplings. For higher mass sparticles, the reduction of the phase space for pair production becomes an important factor, and single sparticle production could dominate even for very small values of the L-violating couplings. The total SUSY production cross sections, as calculated by PYTHIA in its current form are thus underestimated, possibly quite severely for heavy-mass sparticles.

Three possibilities exist for the initializations of the couplings. The first, selected by setting `IMSS(51)=1` for LLE and `IMSS(52)=1` for LQD type couplings, sets all the couplings, independent of generation, to a common value of $10^{-\mathtt{RMSS(51)}}$ or $10^{-\mathtt{RMSS(52)}}$, as the case may be.

Setting `IMSS(51)=2` and/or `IMSS(52)=2` causes the couplings to be initialized (in `PYINIT`) to so-called 'natural' generation-hierarchical values, as proposed in [Hin93]. These values, inspired by the structure of the Yukawa couplings in the SM, are defined by:

$$\begin{aligned} |\lambda_{ijk}|^2 &= (\mathtt{RMSS(51)})^2 \hat{m}_{e_i} \hat{m}_{e_j} \hat{m}_{e_k} \\ |\lambda'_{ijk}|^2 &= (\mathtt{RMSS(52)})^2 \hat{m}_{e_i} \hat{m}_{q_j} \hat{m}_{d_k} \end{aligned} \quad ; \quad \hat{m} \equiv \frac{m}{v} = \frac{m}{126\mathrm{GeV}} \tag{141}$$

where m_{q_j} is the arithmetic mean of m_{u_j} and m_{d_j}.

The third option available is to set `IMSS(51)=3` and/or `IMSS(52)=3`, in which case all the relevant couplings are zero by default (but the lepton number violating processes are turned on) and the user is expected to enter all non-zero coupling values by hand. `RVLAM(`i,j,k`)` contains the λ_{ijk} and `RVLAMP(`i,j,k`)` contains the λ'_{ijk} couplings.

8.8 Polarization

In most processes, incoming beams are assumed unpolarized. However, especially for e^+e^- linear collider studies, polarized beams would provide further important information on many new physics phenomena, and at times help to suppress backgrounds. Therefore a few process cross sections are now available also for polarized incoming beams. The average polarization of the two beams is then set by `PARJ(131)` and `PARJ(132)`, respectively. In some cases, noted below, `MSTP(50)` need also be switched on to access the formulae for polarized beams.

Process 25, W^+W^- pair production, allows polarized incoming lepton beam particles. The polarization effects are included both in the production matrix elements and in the angular distribution of the final four fermions. Note that the matrix element used [Mah98] is for on-shell W production, with a suppression factor added for finite width effects. This polarized cross section expression, evaluated at vanishing polarization, disagrees with the standard unpolarized one, which presumably is the more accurate of the two. The difference can be quite significant below threshold and at very high energies. This can be traced to the simplified description of off-shell W's in the polarized formulae. Good agreement is obtained either by switching off the W width with `MSTP(42)=0` or by restricting the W mass ranges (with `CKIN(41) - CKIN(44)`) to be close to on-shell. It is therefore necessary to set `MSTP(50)=1` to switch from the default standard unpolarized formulae to the polarized ones.

Also many SUSY production processes now include the effects from polarization of the incoming fermion beams. This applies for scalar pair production, with the exception of sneutrino pair production and h^0A^0 and H^0A^0 production, this omission being an oversight at the time of this release, but easily remedied in the future.

The effect of polarized photons is included in the process $\gamma\gamma \to F_k\overline{F}_k$, process 85. Here the array values PARJ(131) and PARJ(132) are used to define the average longitudinal polarization of the two photons.

8.9 Main Processes by Machine

In the previous section we have already commented on which processes have limited validity, or have different meanings (according to conventional terminology) in different contexts. Let us just repeat a few of the main points to be remembered for different machines.

8.9.1 e^+e^- collisions

The main annihilation process is number 1, $e^+e^- \to Z^0$, where in fact the full γ^*/Z^0 interference structure is included. This process can be used, with some confidence, for c.m. energies from about 4 GeV upwards, i.e. at DORIS/CESR, PETRA/PEP, TRISTAN, LEP, and any future linear colliders. (To get below 10 GeV, you have to change `PARP(2)`, however.) This is the default process obtained when `MSEL=1`, i.e. when you do not change anything yourself.

Process 141 contains a Z'^0, including full interference with the standard γ^*/Z^0. With the value `MSTP(44)=4` in fact one is back at the standard γ^*/Z^0 structure, i.e. the Z'^0 piece has been switched off. Even so, this process may be useful, since it can simulate e.g. $e^+e^- \to h^0A^0$. Since the h^0 may in its turn decay to Z^0Z^0, a decay channel of the ordinary Z^0 to h^0A^0, although physically correct, would be technically confusing. In particular, it would be messy to set the original Z^0 to decay one way and the subsequent ones another. So, in this sense, the Z'^0 could be used as a copy of the ordinary Z^0, but with a distinguishable label.

The process $e^+e^- \to \Upsilon$ does not exist as a separate process in PYTHIA, but can be simulated by using `PYONIA`, see section 6.2.

At LEP 2 and even higher energy machines, the simple *s*-channel process 1 loses out to other processes, such as $e^+e^- \to Z^0Z^0$ and $e^+e^- \to W^+W^-$, i.e. processes 22 and 25. The former process in fact includes the structure $e^+e^- \to (\gamma^*/Z^0)(\gamma^*/Z^0)$, which means that the cross section is singular if either of the two γ^*/Z^0 masses is allowed to vanish. A mass cut therefore needs to be introduced, and is actually also used in other processes, such as $e^+e^- \to W^+W^-$.

For practical applications, both with respect to cross sections and to event shapes, it is imperative to include initial-state radiation effects. Therefore `MSTP(11)=1` is the default, wherein exponentiated electron-inside-electron distributions are used to give the momentum of the actually interacting electron. By radiative corrections to process 1, such processes as $e^+e^- \to \gamma Z^0$ are therefore automatically generated. If process 19 were to be used at the same time, this would mean that radiation were to be double-counted. In the alternative `MSTP(11)=0`, electrons are assumed to deposit their full energy in the hard process, i.e. initial-state QED radiation is not included. This option is very useful, since it often corresponds to the 'ideal' events that one wants to correct back to.

Resolved electrons also means that one may have interactions between photons. This opens up the whole field of $\gamma\gamma$ processes, which is described in section 8.3. In particular, with `'gamma/e+','gamma/e-'` as beam and target particles in a `PYINIT` call, a flux of photons of different virtualities is convoluted with a description of direct and resolved photon interaction processes, including both low-$p_\perp$ and high-$p_\perp$ processes. This machinery is directed to the description of the QCD processes, and does e.g. not address the production of gauge bosons or other such particles by the interactions of resolved photons. For the latter kind of applications, a simpler description of partons inside photons inside electrons may be obtained with the `MSTP(12)=1` options and $e^\pm$ as beam and target particles.

The thrust of the PYTHIA programs is towards processes that involve hadron production, one way or another. Because of generalizations from other areas, also a few completely non-hadronic processes are available. These include Bhabha scattering, $e^+e^- \to e^+e^-$ in process 10, and photon pair production, $e^+e^- \to \gamma\gamma$ in process 18. However, note that the precision that could be expected in a PYTHIA simulation of those processes is certainly far less than that of dedicated programs. For one thing, electroweak loop effects are not included. For another, nowhere is the electron mass taken into account, which means that explicit cut-offs at some minimum $p_\perp$ are always necessary.

8.9.2 Lepton–hadron collisions

The main option for photoproduction and Deeply Inelastic Scattering (DIS) physics is provided by the `'gamma/`*lepton*`'` option as beam or target in a `PYINIT` call, see section 8.3. The Q^2 range

to be covered, and other kinematics constraints, can be set by CKIN values. By default, when the whole Q^2 range is populated, obviously photoproduction dominates.

The older DIS process 10, $\ell q \to \ell' q'$, includes $\gamma^0/Z^0/W^\pm$ exchange, with full interference, as described in section 8.3.2. The $Z^0/W^\pm$ contributions are not implemented in the 'gamma/*lepton*' machinery. Therefore process 10 is still the main option for physics at very high Q^2, but has been superseded for lower Q^2. Radiation off the incoming lepton leg is included by MSTP(11)=1 and off the outgoing one by MSTJ(41)=2 (both are default). Note that both QED and QCD radiation (off the e and the q legs, respectively) are allowed to modify the x and Q^2 values of the process, while the conventional approach in the literature is to allow only the former. Therefore an option (on by default) has been added to preserve these values by a post-facto rescaling, MSTP(23)=1. Further comments on HERA applications are found in [Sjö92b].

8.9.3 Hadron–hadron collisions

The default is to include QCD jet production by $2 \to 2$ processes, see section 8.2.1. Since the differential cross section is divergent for $p_\perp \to 0$, a lower cut-off has to be introduced. Normally that cut-off is given by the user-set $p_{\perp\mathrm{min}}$ value in CKIN(3). If CKIN(3) is chosen smaller than a given value of the order of 2 GeV (see PARP(81) and PARP(82)), then low-$p_\perp$ events are also switched on. The jet cross section is regularized at low $p_\perp$, so as to obtain a smooth joining between the high-$p_\perp$ and the low-$p_\perp$ descriptions, see further section 11.2. As CKIN(3) is varied, the jump from one scenario to another is abrupt, in terms of cross section: in a high-energy hadron collider, the cross section for jets down to a $p_{\perp\mathrm{min}}$ scale of a few GeV can well reach values much larger than the total inelastic, non-diffractive cross section. Clearly this is nonsense; therefore either $p_{\perp\mathrm{min}}$ should be picked so large that the jet cross section be only a fraction of the total one, or else one should select $p_{\perp\mathrm{min}} = 0$ and make use of the full description.

If one switches to MSEL=2, also elastic and diffractive processes are switched on, see section 8.2.4. However, the simulation of these processes is fairly primitive, and should not be used for dedicated studies, but only to estimate how much they may contaminate the class of non-diffractive minimum bias events.

Most processes can be simulated in hadron colliders, since the bulk of PYTHIA processes can be initiated by quarks or gluons. However, there are limits. Currently we include no photon or lepton parton distributions, which means that a process like $\gamma q \to \gamma q$ is not accessible. Further, the possibility of having Z^0 and $W^\pm$ interacting in processes such as 71–77 has been hardwired process by process, and does not mean that there is a generic treatment of Z^0 and $W^\pm$ distributions.

The emphasis in the hadron–hadron process description is on high energy hadron colliders. The program can be used also at fixed-target energies, but the multiple interaction model for underlying events then breaks down and should not be used. The limit of applicability is somewhere at around 100 GeV. Only with the simpler model obtained for MSTP(82)=1 can one go arbitrarily low.

9 The Process Generation Program Elements

In the previous two sections, the physics processes and the event-generation schemes of PYTHIA have been presented. Here, finally, the event-generation routines and the common block variables are described. However, routines and variables related to initial- and final-state showers, beam remnants and underlying events, and fragmentation and decay are relegated to subsequent sections on these topics.

In the presentation in this section, information less important for an efficient use of PYTHIA has been put closer to the end. We therefore begin with the main event generation routines, and follow this by the main common block variables.

It is useful to distinguish three phases in a normal run with PYTHIA. In the first phase, the initialization, the general character of the run is determined. At a minimum, this requires the specification of the incoming hadrons and the energies involved. At the discretion of the user, it is also possible to select specific final states, and to make a number of decisions about details in the subsequent generation. This step is finished by a `PYINIT` call, at which time several variables are initialized in accordance with the values set. The second phase consists of the main loop over the number of events, with each new event being generated by a `PYEVNT` call. This event may then be analysed, using information stored in some common blocks, and the statistics accumulated. In the final phase, results are presented. This may often be done without the invocation of any PYTHIA routines. From `PYSTAT`, however, it is possible to obtain a useful list of cross sections for the different subprocesses.

9.1 The Main Subroutines

There are two routines that you must know: `PYINIT` for initialization and `PYEVNT` for the subsequent generation of each new event. In addition, the cross section and other kinds of information available with `PYSTAT` are frequently useful. The other two routines described here, `PYFRAM` and `PYKCUT`, are of more specialized interest.

```
CALL PYINIT(FRAME,BEAM,TARGET,WIN)
```

Purpose: to initialize the generation procedure.

`FRAME` : a character variable used to specify the frame of the experiment. Upper-case and lower-case letters may be freely mixed.

- `= 'CMS'` : colliding beam experiment in c.m. frame, with beam momentum in $+z$ direction and target momentum in $-z$ direction.
- `= 'FIXT'` : fixed-target experiment, with beam particle momentum pointing in $+z$ direction.
- `= '3MOM'` : full freedom to specify frame by giving beam momentum in `P(1,1)`, `P(1,2)` and `P(1,3)` and target momentum in `P(2,1)`, `P(2,2)` and `P(2,3)` in common block `PYJETS`. Particles are assumed on the mass shell, and energies are calculated accordingly.
- `= '4MOM'` : as `'3MOM'`, except also energies should be specified, in `P(1,4)` and `P(2,4)`, respectively. The particles need not be on the mass shell; effective masses are calculated from energy and momentum. (But note that numerical precision may suffer; if you know the masses the option `'5MOM'` below is preferable.)
- `= '5MOM'` : as `'3MOM'`, except also energies and masses should be specified, i.e the full momentum information in `P(1,1) - P(1,5)` and `P(2,1) - P(2,5)` should be given for beam and target, respectively. Particles need not be on the mass shell. Space-like virtualities should be stored as $-\sqrt{-m^2}$. Especially useful for physics with virtual photons. (The virtuality could be varied from one event to the next, but then it is convenient to initialize for the lowest virtuality

likely to be encountered.) Four-momentum and mass information must match.

= 'USER' : a run primarily intended to involve external, user-defined processes, see subsection 9.9. Information on incoming beam particles and energies is read from the HEPRUP common block. In this option, the BEAM, TARGET and WIN arguments are dummy.

= 'NONE' : there will be no initialization of any processes, but only of resonance widths and a few other process-independent variables. Subsequent to such a call, PYEVNT cannot be used to generate events, so this option is mainly intended for those who will want to construct their own events afterwards, but still want to have access to some of the PYTHIA facilities. In this option, the BEAM, TARGET and WIN arguments are dummy.

BEAM, TARGET : character variables to specify beam and target particles. Upper-case and lower-case letters may be freely mixed. An antiparticle can be denoted by 'bar' at the end of the name ('∼' is a valid alternative for reasons of backwards compatibility). It is also possible to leave out the underscore ('_') directly after 'nu' in neutrino names, and the charge for proton and neutron. The arguments are dummy when the FRAME argument above is either 'USER' or 'NONE'.

= 'e-' : electron.

= 'e+' : positron.

= 'nu_e' : ν_e.

= 'nu_ebar' : $\overline{\nu}_e$.

= 'mu-' : μ^-.

= 'mu+' : μ^+.

= 'nu_mu' : ν_μ.

= 'nu_mubar' : $\overline{\nu}_\mu$.

= 'tau-' : τ^-.

= 'tau+' : τ^+.

= 'nu_tau' : ν_τ.

= 'nu_taubar' : $\overline{\nu}_\tau$.

= 'gamma' : photon (real, i.e. on the mass shell).

= 'gamma/e-' : photon generated by the virtual-photon flux in an electron beam; WIN below refers to electron, while photon energy and virtuality varies between events according to what is allowed by CKIN(61) - CKIN(78).

= 'gamma/e+' : as above for a positron beam.

= 'gamma/mu-' : as above for a μ^- beam.

= 'gamma/mu+' : as above for a μ^+ beam.

= 'gamma/tau-' : as above for a τ^- beam.

= 'gamma/tau+' : as above for a τ^+ beam.

= 'pi0' : π^0.

= 'pi+' : π^+.

= 'pi-' : π^-.

= 'n0' : neutron.

= 'nbar0' : antineutron.

= 'p+' : proton.

= 'pbar-' : antiproton.

= 'K+' : K^+ meson; since parton distributions for strange hadrons are not available, very simple and untrustworthy recipes are used for this and subsequent hadrons, see subsection 7.1.

= 'K-' : K^- meson.

= 'KS0' : K^0_S meson.

= 'KL0' : K^0_L meson.

= 'Lambda0' : Λ baryon.

= 'Sigma-' : Σ^- baryon.

= 'Sigma0' : Σ^0 baryon.
= 'Sigma+' : Σ^+ baryon.
= 'Xi-' : Ξ^- baryon.
= 'Xi0' : Ξ^0 baryon.
= 'Omega-' : Ω^- baryon.
= 'pomeron' : the pomeron $\mathbb{P}$; since pomeron parton distribution functions have not been defined this option can not be used currently.
= 'reggeon' : the reggeon $\mathbb{R}$, with comments as for the pomeron above.

WIN : related to energy of system, exact meaning depends on FRAME.
FRAME='CMS' : total energy of system (in GeV).
FRAME='FIXT' : momentum of beam particle (in GeV/c).
FRAME='3MOM', '4MOM', '5MOM' : dummy (information is taken from the P vectors, see above).
FRAME='USER' : dummy (information is taken from the HEPRUP common block, see above).
FRAME='NONE' : dummy (no information required).

```
CALL PYEVNT
```

Purpose: to generate one event of the type specified by the PYINIT call. (This is the main routine, which calls a number of other routines for specific tasks.)

```
CALL PYSTAT(MSTAT)
```

Purpose: to print out cross-sections statistics, decay widths, branching ratios, status codes and parameter values. PYSTAT may be called at any time, after the PYINIT call, e.g. at the end of the run, or not at all.

MSTAT : specification of desired information.

= 1 : prints a table of how many events of the different kinds that have been generated and the corresponding cross sections. All numbers already include the effects of cuts required by you in PYKCUT.

Note that no errors are given on the cross sections. In most cases a cross section is obtained by Monte Carlo integration during the course of the run. (Exceptions include e.g. total and elastic hadron–hadron cross sections, which are parameterized and thus known from the very onset.) A rule of thumb would then be that the statistical error of a given subprocess scales like $\delta\sigma/\sigma \approx 1/\sqrt{n}$, where n is the number of events generated of this kind. In principle, the numerator of this relation could be decreased by making use of the full information accumulated during the run, i.e. also on the cross section in those phase space points that are eventually rejected. This is actually the way the cross section itself is calculated. However, once you introduce further cuts so that only some fraction of the generated events survive to the final analysis, you would be back to the simple $1/\sqrt{n}$ scaling rule for that number of surviving events. Statistical errors are therefore usually better evaluated within the context of a specific analysis. Furthermore, systematic errors often dominate over the statistical ones.

Also note that runs with very few events, in addition to having large errors, tend to have a bias towards overestimating the cross sections. In a typical case, the average cross section obtained with many runs of only one event each may be twice that of the correct answer of a single run with many events. The reason is a 'quit while you are ahead' phenomenon, that an upwards fluctuation in the differential cross section in an early try gives an acceptable event and

thus terminates the run, while a downwards one leads to rejection and a continuation of the run.

`= 2 :` prints a table of the resonances defined in the program, with their particle codes (`KF`), and all allowed decay channels. (If the number of generations in `MSTP(1)` is 3, however, channels involving fourth-generation particles are not displayed.) For each decay channel is shown the sequential channel number (IDC) of the PYTHIA decay tables, the decay products (usually two but sometimes three), the partial decay width, branching ratio and effective branching ratio (in the event some channels have been excluded by you).

`= 3 :` prints a table with the allowed hard interaction flavours `KFIN(I,J)` for beam and target particles.

`= 4 :` prints a table of the kinematical cuts `CKIN(I)` set by you in the current run.

`= 5 :` prints a table with all the values of the status codes `MSTP(I)` and the parameters `PARP(I)` used in the current run.

`= 6 :` prints a table of all subprocesses implemented in the program.

`= 7 :` prints two tables related to *R*-violating supersymmetry. The first is a collection of semi-inclusive branching ratios where the entries have the form `~chi_10 --> nu + q + q` where a sum has been performed over all lepton and quark flavours. In the rightmost coloumn of the table, the number of modes that went into the sum is given. The purpose of this table is to give a quick overview of the branching fractions, since there are currently more than 1200 individual *R*-violating processes included in the generator. Note that only the pure $1 \to 3$ parts of the 3-body modes are included in this sum. If a process can also proceed via two successive $1 \to 2$ branchings (i.e. the intermediate resonance is on shell) the product of these branchings should be added to the number given in this table. A small list at the bottom of the table shows the total number of *R*-violating processes in the generator, the number with non-zero branching ratios in the current run, and the number with branching ratios larger than 10^{-3}. The second table which is printed by this call merely lists the *R*-violating λ, λ', and λ'' couplings.

```
CALL PYFRAM(IFRAME)
```

Purpose: to transform an event listing between different reference frames, if so desired. The use of this routine assumes you do not do any boosts yourself.

`IFRAME :` specification of frame the event is to be boosted to.

`= 1 :` frame specified by you in the `PYINIT` call.

`= 2 :` c.m. frame of incoming particles.

`= 3 :` hadronic c.m. frame of lepton–hadron interaction events. Mainly intended for Deeply Inelastic Scattering, but can also be used in photoproduction. Is not guaranteed to work with the `'gamma/`*lepton*`'` options, however, and so of limited use. Note that both the lepton and any photons radiated off the lepton remain in the event listing, and have to be removed separately if you only want to study the hadronic subsystem.

```
CALL PYKCUT(MCUT)
```

Purpose: to enable you to reject a given set of kinematic variables at an early stage of the generation procedure (before evaluation of cross sections), so as not to spend unnecessary time on the generation of events that are not wanted. The routine will not be called unless you require is by setting `MSTP(141)=1`, and never if 'minimum-bias'-type events (including elastic and diffractive scattering) are to be generated as

well. Furthermore it is never called for user-defined external processes. A dummy routine PYKCUT is included in the program file, so as to avoid unresolved external references when the routine is not used.

MCUT : flag to signal effect of user-defined cuts.

= 0 : event is to be retained and generated in full.

= 1 : event is to be rejected and a new one generated.

Remark : at the time of selection, several variables in the MINT and VINT arrays in the PYINT1 common block contain information that can be used to make the decision. The routine provided in the program file explicitly reads the variables that have been defined at the time PYKCUT is called, and also calculates some derived quantities. The information available includes subprocess type ISUB, E_{cm}, $\hat{s}$, $\hat{t}$, $\hat{u}$, $\hat{p}_{\perp}$, x_1, x_2, x_{F}, τ, y, τ', $\cos\hat{\theta}$, and a few more. Some of these may not be relevant for the process under study, and are then set to zero.

9.2 Switches for Event Type and Kinematics Selection

By default, if PYTHIA is run for a hadron collider, only QCD $2 \to 2$ processes are generated, composed of hard interactions above $p_{\perp\mathrm{min}}$ =PARP(81), with low-$p_{\perp}$ processes added on so as to give the full (parameterized) inelastic, non-diffractive cross section. In an $\mathrm{e}^+\mathrm{e}^-$ collider, γ^*/Z^0 production is the default, and in an ep one it is Deeply Inelastic Scattering. With the help of the common block PYSUBS, it is possible to select the generation of another process, or combination of processes. It is also allowed to restrict the generation to specific incoming partons/particles at the hard interaction. This often automatically also restricts final-state flavours but, in processes such as resonance production or QCD/QED production of new flavours, switches in the PYTHIA program may be used to this end; see section 14.4.

The CKIN array may be used to impose specific kinematics cuts. You should here be warned that, if kinematical variables are too strongly restricted, the generation time per event may become very long. In extreme cases, where the cuts effectively close the full phase space, the event generation may run into an infinite loop. The generation of $2 \to 1$ resonance production is performed in terms of the $\hat{s}$ and y variables, and so the ranges CKIN(1) - CKIN(2) and CKIN(7) - CKIN(8) may be arbitrarily restricted without a significant loss of speed. For $2 \to 2$ processes, $\cos\hat{\theta}$ is added as a third generation variable, and so additionally the range CKIN(27) - CKIN(28) may be restricted without any loss of efficiency..

Effects from initial- and final-state radiation are not included, since they are not known at the time the kinematics at the hard interaction is selected. The sharp kinematical cut-offs that can be imposed on the generation process are therefore smeared, both by QCD radiation and by fragmentation. A few examples of such effects follow.

- Initial-state radiation implies that each of the two incoming partons has a non-vanishing $p_{\perp}$ when they interact. The hard scattering subsystem thus receives a net transverse boost, and is rotated with respect to the beam directions. In a $2 \to 2$ process, what typically happens is that one of the scattered partons receives an increased $p_{\perp}$, while the $p_{\perp}$ of the other parton is reduced.
- Since the initial-state radiation machinery assigns space-like virtualities to the incoming partons, the definitions of x in terms of energy fractions and in terms of momentum fractions no longer coincide, and so the interacting subsystem may receive a net longitudinal boost compared with naïve expectations, as part of the parton-shower machinery.
- Initial-state radiation gives rise to additional jets, which in extreme cases may be mistaken for either of the jets of the hard interaction.
- Final-state radiation gives rise to additional jets, which smears the meaning of the basic $2 \to 2$ scattering. The assignment of soft jets is not unique. The energy of a jet becomes dependent on the way it is identified, e.g. what jet cone size is used.
- The beam remnant description assigns primordial $k_{\perp}$ values, which also gives a net $p_{\perp}$

shift of the hard-interaction subsystem; except at low energies this effect is overshadowed by initial-state radiation, however. Beam remnants may also add further activity under the 'perturbative' event.

- Fragmentation will further broaden jet profiles, and make jet assignments and energy determinations even more uncertain.

In a study of events within a given window of experimentally defined variables, it is up to you to leave such liberal margins that no events are missed. In other words, cuts have to be chosen such that a negligible fraction of events migrate from outside the simulated region to inside the interesting region. Often this may lead to low efficiency in terms of what fraction of the generated events are actually of interest to you. See also section 3.6.

In addition to the variables found in PYSUBS, also those in the PYPARS common block may be used to select exactly what one wants to have simulated. These possibilities will be described in the following section.

The notation used above and in the following is that '^' denotes internal variables in the hard scattering subsystem, while '*' is for variables in the c.m. frame of the event as a whole.

```
COMMON/PYSUBS/MSEL,MSELPD,MSUB(500),KFIN(2,-40:40),CKIN(200)
```

Purpose: to allow you to run the program with any desired subset of processes, or restrict flavours or kinematics. If the default values, denoted below by (D=...), are not satisfactory, they must be changed before the PYINIT call.

MSEL : (D=1) a switch to select between full user control and some preprogrammed alternatives.

= 0 : desired subprocesses have to be switched on in MSUB, i.e. full user control.

= 1 : depending on incoming particles, different alternatives are used.

Lepton–lepton: Z or W production (ISUB = 1 or 2).

Lepton–hadron: Deeply Inelastic Scattering (ISUB = 10; this option is now out of date for most applications, superseded by the 'gamma/*lepton*' machinery).

Hadron–hadron: QCD high-$p_\perp$ processes (ISUB = 11, 12, 13, 28, 53, 68); additionally low-$p_\perp$ production if CKIN(3)<PARP(81) or PARP(82), depending on MSTP(82) (ISUB = 95). If low-$p_\perp$ is switched on, the other CKIN cuts are not used.

A resolved photon counts as hadron. When the photon is not resolved, the following cases are possible.

Photon–lepton: Compton scattering (ISUB = 34).

Photon–hadron: photon-parton scattering (ISUB = 33, 34, 54).

Photon–photon: fermion pair production (ISUB = 58).

When photons are given by the 'gamma/*lepton*' argument in the PYINIT call, the outcome depends on the MSTP(14) value. Default is a mixture of many kinds of processes, as described in section 8.3.

= 2 : as MSEL = 1 for lepton–lepton, lepton–hadron and unresolved photons. For hadron–hadron (including resolved photons) all QCD processes, including low-$p_\perp$, single and double diffractive and elastic scattering, are included (ISUB = 11, 12, 13, 28, 53, 68, 91, 92, 93, 94, 95). The CKIN cuts are here not used.

For photons given with the 'gamma/*lepton*' argument in the PYINIT call, the above processes are replaced by other ones that also include the photon virtuality in the cross sections. The principle remains to include both high- and low-$p_\perp$ processes, however.

= 4 : charm ($c\bar{c}$) production with massive matrix elements (ISUB = 81, 82, 84, 85).

= 5 : bottom ($b\bar{b}$) production with massive matrix elements (ISUB = 81, 82, 84, 85).

= 6 : top ($t\bar{t}$) production with massive matrix elements (ISUB = 81, 82, 84, 85).

= 7 : fourth generation b' ($b'\bar{b}'$) production with massive matrix elements (ISUB = 81, 82, 84, 85).

= 8 : fourth generation t' ($t'\bar{t}'$) production with massive matrix elements (ISUB = 81, 82, 84, 85).

= 10 : prompt photons (ISUB = 14, 18, 29).

= 11 : Z^0 production (ISUB = 1).

= 12 : $W^\pm$ production (ISUB = 2).

= 13 : Z^0 + jet production (ISUB = 15, 30).

= 14 : $W^\pm$ + jet production (ISUB = 16, 31).

= 15 : pair production of different combinations of γ, Z^0 and $W^\pm$ (except $\gamma\gamma$; see MSEL = 10) (ISUB = 19, 20, 22, 23, 25).

= 16 : h^0 production (ISUB = 3, 102, 103, 123, 124).

= 17 : h^0Z^0 or $h^0W^\pm$ (ISUB = 24, 26).

= 18 : h^0 production, combination relevant for e^+e^- annihilation (ISUB = 24, 103, 123, 124).

= 19 : h^0, H^0 and A^0 production, excepting pair production (ISUB = 24, 103, 123, 124, 153, 158, 171, 173, 174, 176, 178, 179).

= 21 : Z'^0 production (ISUB = 141).

= 22 : $W'^\pm$ production (ISUB = 142).

= 23 : $H^\pm$ production (ISUB = 143).

= 24 : R^0 production (ISUB = 144).

= 25 : L_Q (leptoquark) production (ISUB = 145, 162, 163, 164).

= 35: single bottom production by W exchange (ISUB = 83).

= 36: single top production by W exchange (ISUB = 83).

= 37: single b' production by W exchange (ISUB = 83).

= 38: single t' production by W exchange (ISUB = 83).

= 39: all MSSM processes except Higgs production.

= 40: squark and gluino production (ISUB = 243, 244, 258, 259, 271–280).

= 41: stop pair production (ISUB = 261–265).

= 42: slepton pair production (ISUB = 201–214).

= 43: squark or gluino with chargino or neutralino, (ISUB = 237–242, 246–256).

= 44: chargino–neutralino pair production (ISUB = 216–236).

= 45: sbottom production (ISUB = 281–296).

= 50: pair production of technipions and gauge bosons by $\pi_{tc}^{0,\pm}/\omega_{tc}^0$ exchange (ISUB = 361–377).

MSUB : (D=500*0) array to be set when MSEL=0 (for MSEL$\geq$ 1 relevant entries are set in PYINIT) to choose which subset of subprocesses to include in the generation. The ordering follows the ISUB code given in section 8.1 (with comments as given there).

MSUB(ISUB) = 0 : the subprocess is excluded.

MSUB(ISUB) = 1 : the subprocess is included.

Note: when MSEL=0, the MSUB values set by you are never changed by PYTHIA. If you want to combine several different 'subruns', each with its own PYINIT call, into one single run, it is up to you to remember not only to switch on the new processes before each new PYINIT call, but also to switch off the old ones that are no longer desired.

KFIN(I,J) : provides an option to selectively switch on and off contributions to the cross sections from the different incoming partons/particles at the hard interaction. In combination with the PYTHIA resonance decay switches, this also allows you to set restrictions on flavours appearing in the final state.

I : is 1 for beam side of event and 2 for target side.

`J :` enumerates flavours according to the `KF` code; see section 5.1.
`KFIN(I,J) = 0 :` the parton/particle is forbidden.
`KFIN(I,J) = 1 :` the parton/particle is allowed.
Note: By default, the following are switched on: d, u, s, c, b, e^-, ν_e, μ^-, ν_μ, τ^-, ν_τ, g, γ, Z^0, W^+ and their antiparticles. In particular, top is off, and has to be switched on explicitly if needed.

`CKIN :` kinematics cuts that can be set by you before the `PYINIT` call, and that affect the region of phase space within which events are generated. Some cuts are 'hard-wired' while most are 'softwired'. The hardwired ones are directly related to the kinematical variables used in the event selection procedure, and therefore have negligible effects on program efficiency. The most important of these are `CKIN(1) - CKIN(8)`, `CKIN(27) - CKIN(28)`, and `CKIN(31) - CKIN(32)`. The softwired ones are most of the remaining ones, that cannot be fully taken into account in the kinematical variable selection, so that generation in constrained regions of phase space may be slow. In extreme cases the phase space may be so small that the maximization procedure fails to find any allowed points at all (although some small region might still exist somewhere), and therefore switches off some subprocesses, or aborts altogether.

`CKIN(1), CKIN(2) :` (D=2.,-1. GeV) range of allowed $\hat{m} = \sqrt{\hat{s}}$ values. If `CKIN(2)`$< 0.$, the upper limit is inactive.

`CKIN(3), CKIN(4) :` (D=0.,-1. GeV) range of allowed $\hat{p}_\perp$ values for hard $2 \to 2$ processes, with transverse momentum $\hat{p}_\perp$ defined in the rest frame of the hard interaction. If `CKIN(4)`$< 0.$, the upper limit is inactive. For processes that are singular in the limit $\hat{p}_\perp \to 0$ (see `CKIN(6)`), `CKIN(5)` provides an additional constraint. The `CKIN(3)` and `CKIN(4)` limits can also be used in $2 \to 1 \to 2$ processes. Here, however, the product masses are not known and hence are assumed to be vanishing in the event selection. The actual $p_\perp$ range for massive products is thus shifted downwards with respect to the nominal one.

`CKIN(5) :` (D=1. GeV) lower cut-off on $\hat{p}_\perp$ values, in addition to the `CKIN(3)` cut above, for processes that are singular in the limit $\hat{p}_\perp \to 0$ (see `CKIN(6)`).

`CKIN(6) :` (D=1. GeV) hard $2 \to 2$ processes, which do not proceed only via an intermediate resonance (i.e. are $2 \to 1 \to 2$ processes), are classified as singular in the limit $\hat{p}_\perp \to 0$ if either or both of the two final-state products has a mass $m <$`CKIN(6)`.

`CKIN(7), CKIN(8) :` (D=-10.,10.) range of allowed scattering subsystem rapidities $y = y^*$ in the c.m. frame of the event, where $y = (1/2)\ln(x_1/x_2)$. (Following the notation of this section, the variable should be given as y^*, but because of its frequent use, it was called y in section 7.2.)

`CKIN(9), CKIN(10) :` (D=-40.,40.) range of allowed (true) rapidities for the product with largest rapidity in a $2 \to 2$ or a $2 \to 1 \to 2$ process, defined in the c.m. frame of the event, i.e. $\max(y_3^*, y_4^*)$. Note that rapidities are counted with sign, i.e. if $y_3^* = 1$ and $y_4^* = -2$ then $\max(y_3^*, y_4^*) = 1$.

`CKIN(11), CKIN(12) :` (D=-40.,40.) range of allowed (true) rapidities for the product with smallest rapidity in a $2 \to 2$ or a $2 \to 1 \to 2$ process, defined in the c.m. frame of the event, i.e. $\min(y_3^*, y_4^*)$. Consistency thus requires `CKIN(11)`$\leq$`CKIN(9)` and `CKIN(12)`$\leq$`CKIN(10)`.

`CKIN(13), CKIN(14) :` (D=-40.,40.) range of allowed pseudorapidities for the product with largest pseudorapidity in a $2 \to 2$ or a $2 \to 1 \to 2$ process, defined in the c.m. frame of the event, i.e. $\max(\eta_3^*, \eta_4^*)$. Note that pseudorapidities are counted with sign, i.e. if $\eta_3^* = 1$ and $\eta_4^* = -2$ then $\max(\eta_3^*, \eta_4^*) = 1$.

`CKIN(15), CKIN(16) :` (D=-40.,40.) range of allowed pseudorapidities for the product with smallest pseudorapidity in a $2 \to 2$ or a $2 \to 1 \to 2$ process, defined in the c.m. frame of the event, i.e. $\min(\eta_3^*, \eta_4^*)$. Consistency thus requires `CKIN(15)`$\leq$`CKIN(13)` and

CKIN(16)≤CKIN(14).

CKIN(17), CKIN(18) : (D=-1.,1.) range of allowed $\cos\theta^*$ values for the product with largest $\cos\theta^*$ value in a $2 \to 2$ or a $2 \to 1 \to 2$ process, defined in the c.m. frame of the event, i.e. $\max(\cos\theta_3^*, \cos\theta_4^*)$.

CKIN(19), CKIN(20) : (D=-1.,1.) range of allowed $\cos\theta^*$ values for the product with smallest $\cos\theta^*$ value in a $2 \to 2$ or a $2 \to 1 \to 2$ process, defined in the c.m. frame of the event, i.e. $\min(\cos\theta_3^*, \cos\theta_4^*)$. Consistency thus requires CKIN(19)≤CKIN(17) and CKIN(20)≤CKIN(18).

CKIN(21), CKIN(22) : (D=0.,1.) range of allowed x_1 values for the parton on side 1 that enters the hard interaction.

CKIN(23), CKIN(24) : (D=0.,1.) range of allowed x_2 values for the parton on side 2 that enters the hard interaction.

CKIN(25), CKIN(26) : (D=-1.,1.) range of allowed Feynman-x values, where $x_F = x_1 - x_2$.

CKIN(27), CKIN(28) : (D=-1.,1.) range of allowed $\cos\hat{\theta}$ values in a hard $2 \to 2$ scattering, where $\hat{\theta}$ is the scattering angle in the rest frame of the hard interaction.

CKIN(31), CKIN(32) : (D=2.,-1. GeV) range of allowed $\hat{m}' = \sqrt{\hat{s}'}$ values, where $\hat{m}'$ is the mass of the complete three- or four-body final state in $2 \to 3$ or $2 \to 4$ processes (while $\hat{m}$, constrained in CKIN(1) and CKIN(2), here corresponds to the one- or two-body central system). If CKIN(32)< 0., the upper limit is inactive.

CKIN(35), CKIN(36) : (D=0.,-1. GeV^2) range of allowed $|\hat{t}| = -\hat{t}$ values in $2 \to 2$ processes. Note that for Deeply Inelastic Scattering this is nothing but the Q^2 scale, in the limit that initial- and final-state radiation is neglected. If CKIN(36)< 0., the upper limit is inactive.

CKIN(37), CKIN(38) : (D=0.,-1. GeV^2) range of allowed $|\hat{u}| = -\hat{u}$ values in $2 \to 2$ processes. If CKIN(38)< 0., the upper limit is inactive.

CKIN(39), CKIN(40) : (D=4., -1. GeV^2) the W^2 range allowed in DIS processes, i.e. subprocess number 10. If CKIN(40)< 0., the upper limit is inactive. Here W^2 is defined in terms of $W^2 = Q^2(1-x)/x$. This formula is not quite correct, in that *(i)* it neglects the target mass (for a proton), and *(ii)* it neglects initial-state photon radiation off the incoming electron. It should be good enough for loose cuts, however. These cuts are not checked if process 10 is called for two lepton beams.

CKIN(41) - CKIN(44) : (D=12.,-1.,12.,-1. GeV) range of allowed mass values of the two (or one) resonances produced in a 'true' $2 \to 2$ process, i.e. one not (only) proceeding through a single s-channel resonance ($2 \to 1 \to 2$). (These are the ones listed as $2 \to 2$ in the tables in section 8.1.) Only particles with a width above PARP(41) are considered as bona fide resonances and tested against the CKIN limits; particles with a smaller width are put on the mass shell without applying any cuts. The exact interpretation of the CKIN variables depends on the flavours of the two produced resonances.

For two resonances like Z^0W^+ (produced from $f_i\bar{f}_j \to Z^0W^+$), which are not identical and which are not each other's antiparticles, one has

CKIN(41)< m_1 <CKIN(42), and

CKIN(43)< m_2 <CKIN(44),

where m_1 and m_2 are the actually generated masses of the two resonances, and 1 and 2 are defined by the order in which they are given in the production process specification.

For two resonances like Z^0Z^0, which are identical, or W^+W^-, which are each other's antiparticles, one instead has

CKIN(41)< $\min(m_1, m_2)$ <CKIN(42), and

CKIN(43)< $\max(m_1, m_2)$ <CKIN(44).

In addition, whatever limits are set on CKIN(1) and, in particular, on CKIN(2) obviously affect the masses actually selected.

Note 1: If MSTP(42)=0, so that no mass smearing is allowed, the CKIN values have no

effect (the same as for particles with too narrow a width).

Note 2: If `CKIN(42)`<`CKIN(41)` it means that the `CKIN(42)` limit is inactive; correspondingly, if `CKIN(44)`<`CKIN(43)` then `CKIN(44)` is inactive.

Note 3: If limits are active and the resonances are identical, it is up to you to ensure that `CKIN(41)`≤`CKIN(43)` and `CKIN(42)`≤`CKIN(44)`.

Note 4: For identical resonances, it is not possible to preselect which of the resonances is the lighter one; if, for instance, one Z^0 is to decay to leptons and the other to quarks, there is no mechanism to guarantee that the lepton pair has a mass smaller than the quark one.

Note 5: The `CKIN` values are applied to all relevant $2 \to 2$ processes equally, which may not be what one desires if several processes are generated simultaneously. Some caution is therefore urged in the use of the `CKIN(41) - CKIN(44)` values. Also in other respects, you are recommended to take proper care: if a Z^0 is only allowed to decay into $b\overline{b}$, for example, setting its mass range to be 2–8 GeV is obviously not a good idea.

`CKIN(45) - CKIN(48) :` (D=12.,-1.,12.,-1. GeV) range of allowed mass values of the two (or one) secondary resonances produced in a $2 \to 1 \to 2$ process (like $gg \to h^0 \to Z^0Z^0$) or even a $2 \to 2 \to 4$ (or 3) process (like $q\overline{q} \to Z^0h^0 \to Z^0W^+W^-$). Note that these `CKIN` values only affect the secondary resonances; the primary ones are constrained by `CKIN(1)`, `CKIN(2)` and `CKIN(41) - CKIN(44)` (indirectly, of course, the choice of primary resonance masses affects the allowed mass range for the secondary ones). What is considered to be a resonance is defined by `PARP(41)`; particles with a width smaller than this are automatically put on the mass shell. The description closely parallels the one given for `CKIN(41) - CKIN(44)`. Thus, for two resonances that are not identical or each other's antiparticles, one has
`CKIN(45)`$< m_1 <$`CKIN(46)`, and
`CKIN(47)`$< m_2 <$`CKIN(48)`,
where m_1 and m_2 are the actually generated masses of the two resonances, and 1 and 2 are defined by the order in which they are given in the decay channel specification in the program (see e.g. output from `PYSTAT(2)` or `PYLIST(12)`). For two resonances that are identical or each other's antiparticles, one instead has
`CKIN(45)`$< \min(m_1, m_2) <$`CKIN(46)`, and
`CKIN(47)`$< \max(m_1, m_2) <$`CKIN(48)`.

Notes 1 - 5: as for `CKIN(41) - CKIN(44)`, with trivial modifications.

Note 6: Setting limits on secondary resonance masses is possible in any of the channels of the allowed types (see above). However, so far only $h^0 \to Z^0Z^0$ and $h^0 \to W^+W^-$ have been fully implemented, such that an arbitrary mass range below the naïve mass threshold may be picked. For other possible resonances, any restrictions made on the allowed mass range are not reflected in the cross section; and further it is not recommendable to pick mass windows that make a decay on the mass shell impossible.

`CKIN(49) - CKIN(50) :` allow minimum mass limits to be passed from `PYRESD` to `PYOFSH`. They are used for tertiary and higher resonances, i.e. those not controlled by `CKIN(41)-CKIN(48)`. They should not be touched by the user.

`CKIN(51) - CKIN(56) :` (D=0.,-1.,0.,-1.,0.,-1. GeV) range of allowed transverse momenta in a true $2 \to 3$ process. This means subprocesses such as 121–124 for h^0 production, and their H^0, A^0 and $H^{\pm\pm}$ equivalents. `CKIN(51) - CKIN(54)` corresponds to $p_\perp$ ranges for scattered partons, in order of appearance, i.e. `CKIN(51) - CKIN(52)` for the parton scattered off the beam and `CKIN(53) - CKIN(54)` for the one scattered off the target. `CKIN(55)` and `CKIN(56)` here sets $p_\perp$ limits for the third product, the h^0, i.e. the `CKIN(3)` and `CKIN(4)` values have no effect for this process. Since the $p_\perp$ of the Higgs is not one of the primary variables selected, any constraints here may mean reduced Monte Carlo efficiency, while for these processes `CKIN(51) -`

CKIN(54) are 'hardwired' and therefore do not cost anything. As usual, a negative value implies that the upper limit is inactive.

CKIN(61) - CKIN(78) : allows to restrict the range of kinematics for the photons generated off the lepton beams with the 'gamma/*lepton*' option of PYINIT. In each quartet of numbers, the first two corresponds to the range allowed on incoming side 1 (beam) and the last two to side 2 (target). The cuts are only applicable for a lepton beam. Note that the x and Q^2 (P^2) variables are the basis for the generation, and so can be restricted with no loss of efficiency. For leptoproduction (i.e. lepton on hadron) the W is uniquely given by the one x value of the problem, so here also W cuts are fully efficient. The other cuts may imply a slowdown of the program, but not as much as if equivalent cuts only are introduced after events are fully generated. See [Fri00] for details.

CKIN(61) - CKIN(64) : (D=0.0001,0.99,0.0001,0.99) allowed range for the energy fractions x that the photon take of the respective incoming lepton energy. These fractions are defined in the c.m. frame of the collision, and differ from energy fractions as defined in another frame. (Watch out at HERA!) In order to avoid some technical problems, absolute lower and upper limits are set internally at 0.0001 and 0.9999.

CKIN(65) - CKIN(68) : (D=0.,-1.,0.,-1. GeV^2) allowed range for the spacelike virtuality of the photon, conventionally called either Q^2 or P^2, depending on process. A negative number means that the upper limit is inactive, i.e. purely given by kinematics. A nonzero lower limit is implicitly given by kinematics constraints.

CKIN(69) - CKIN(72) : (D=0.,-1.,0.,-1.) allowed range of the scattering angle θ of the lepton, defined in the c.m. frame of the event. (Watch out at HERA!) A negative number means that the upper limit is inactive, i.e. equal to π.

CKIN(73) - CKIN(76) : (D=0.0001,0.99,0.0001,0.99) allowed range for the light-cone fraction y that the photon take of the respective incoming lepton energy. The light-cone is defined by the four-momentum of the lepton or hadron on the other side of the event (and thus deviates from true light-cone fraction by mass effects that normally are negligible). The y value is related to the x and Q^2 (P^2) values by $y = x + Q^2/s$ if mass terms are neglected.

CKIN(77), CKIN(78) : (D=2.,-1. GeV) allowed range for W, i.e. either the photon–hadron or photon–photon invariant mass. A negative number means that the upper limit is inactive.

9.3 The General Switches and Parameters

The PYPARS common block contains the status code and parameters that regulate the performance of the program. All of them are provided with sensible default values, so that a novice user can neglect them, and only gradually explore the full range of possibilities. Some of the switches and parameters in PYPARS will be described later, in the shower and beam remnants sections.

```
COMMON/PYPARS/MSTP(200),PARP(200),MSTI(200),PARI(200)
```

Purpose: to give access to status code and parameters that regulate the performance of the program. If the default values, denoted below by (D=...), are not satisfactory, they must in general be changed before the PYINIT call. Exceptions, i.e. variables that can be changed for each new event, are denoted by (C).

MSTP(1) : (D=3) maximum number of generations. Automatically set ≤ 4.

MSTP(2) : (D=1) calculation of α_s at hard interaction, in the routine PYALPS.

= 0 : α_s is fixed at value PARU(111).

= 1 : first-order running α_s.

= 2 : second-order running α_s.

MSTP(3) : (D=2) selection of Λ value in α_s for MSTP(2)≥ 1.

= 1 : Λ is given by PARP(1) for hard interactions, by PARP(61) for space-like showers, by PARP(72) for time-like showers not from a resonance decay, and by PARJ(81) for time-like ones from a resonance decay (including e.g. $\gamma/Z^0 \to q\overline{q}$ decays, i.e. conventional e^+e^- physics). This Λ is assumed to be valid for 5 flavours; for the hard interaction the number of flavours assumed can be changed by MSTU(112).

= 2 : Λ value is chosen according to the parton-distribution-function parameterizations. The choice is always based on the proton parton-distribution set selected, i.e. is unaffected by pion and photon parton-distribution selection. All the Λ values are assumed to refer to 4 flavours, and MSTU(112) is set accordingly. This Λ value is used both for the hard scattering and the initial- and final-state radiation. The ambiguity in the choice of the Q^2 argument still remains (see MSTP(32), MSTP(64) and MSTJ(44)). This Λ value is used also for MSTP(57)=0, but the sensible choice here would be to use MSTP(2)=0 and have no initial- or final-state radiation. This option does *not* change the PARJ(81) value of timelike parton showers in resonance decays, so that LEP experience on this specific parameter is not overwritten unwittingly. Therefore PARJ(81) can be updated completely independently.

= 3 : as =2, except that here also PARJ(81) is overwritten in accordance with the Λ value of the proton parton-distribution-function set.

MSTP(4) : (D=0) treatment of the Higgs sector, predominantly the neutral one.

= 0 : the h^0 is given the Standard Model Higgs couplings, while H^0 and A^0 couplings should be set by you in PARU(171) - PARU(175) and PARU(181) - PARU(185), respectively.

= 1 : you should set couplings for all three Higgses, for the h^0 in PARU(161) - PARU(165), and for the H^0 and A^0 as above.

= 2 : the mass of h^0 in PMAS(25,1) and the $\tan\beta$ value in PARU(141) are used to derive H^0, A^0 and $H^\pm$ masses, and h^0, H^0, A^0 and $H^\pm$ couplings, using the relations of the Minimal Supersymmetric extension of the Standard Model at Born level [Gun90]. Existing masses and couplings are overwritten by the derived values. See section 8.5.3 for discussion on parameter constraints.

= 3: as =2, but using relations at the one-loop level. This option is not yet implemented as such. However, if you initialize the SUSY machinery with IMSS(1)=1, then the SUSY parameters will be used to calculate also Higgs masses and couplings. These are stored in the appropriate slots, and the value of MSTP(4) is overwritten to 1.

MSTP(5) : (D=0) presence of anomalous couplings in processes. See section 8.6.5 for further details.

= 0 : absent.

= 1 : left–left isoscalar model, with only u and d quarks composite (at the probed scale).

= 2 : left–left isoscalar model, with all quarks composite.

= 3 : helicity-non-conserving model, with only u and d quarks composite (at the probed scale).

= 4 : helicity-non-conserving model, with all quarks composite.

= 5 : coloured technihadrons, affecting the standard QCD $2 \to 2$ cross sections, see subsection 8.6.7 and parameters PARP(155) and PARP(156).

MSTP(7) : (D=0) choice of heavy flavour in subprocesses 81–85. Does not apply for MSEL=4-8, where the MSEL value always takes precedence.

= 0 : for processes 81–84 (85) the 'heaviest' flavour allowed for gluon (photon) splitting into a quark–antiquark (fermion–antifermion) pair, as set in the MDME

array. Note that 'heavy' is defined as the one with largest `KF` code, so that leptons take precedence if they are allowed.

`= 1 - 8 :` pick this particular quark flavour; e.g., `MSTP(7)=6` means that top will be produced.

`= 11 - 18 :` pick this particular lepton flavour. Note that neutrinos are not possible, i.e. only 11, 13, 15 and 17 are meaningful alternatives. Lepton pair production can only occur in process 85, so if any of the other processes have been switched on they are generated with the same flavour as would be obtained in the option `MSTP(7)=0`.

`MSTP(8) :` (D=0) choice of electroweak parameters to use in the decay widths of resonances (W, Z, h, ...) and cross sections (production of W's, Z's, h's, ...).

`= 0 :` everything is expressed in terms of a running $\alpha_{\rm em}(Q^2)$ and a fixed $\sin^2\theta_W$, i.e. $G_{\rm F}$ is nowhere used.

`= 1 :` a replacement is made according to $\alpha_{\rm em}(Q^2) \to \sqrt{2}G_{\rm F}m_{\rm W}^2\sin^2\theta_W/\pi$ in all widths and cross sections. If $G_{\rm F}$ and $m_{\rm Z}$ are considered as given, this means that $\sin^2\theta_W$ and $m_{\rm W}$ are the only free electroweak parameter.

`= 2 :` a replacement is made as for `=1`, but additionally $\sin^2\theta_W$ is constrained by the relation $\sin^2\theta_W = 1 - m_{\rm W}^2/m_{\rm Z}^2$. This means that $m_{\rm W}$ remains as a free parameter, but that the $\sin^2\theta_W$ value in `PARU(102)` is never used, *except* in the vector couplings in the combination $v = a - 4\sin^2\theta_W e$. This latter degree of freedom enters e.g. for forward-backward asymmetries in Z^0 decays.

Note: This option does not affect the emission of real photons in the initial and final state, where $\alpha_{\rm em}$ is always used. However, it does affect also purely electromagnetic hard processes, such as $q\overline{q} \to \gamma\gamma$.

`MSTP(9) :` (D=0) inclusion of top (and fourth generation) as allowed remnant flavour q' in processes that involve $q \to q' + W$ branchings as part of the overall process, and where the matrix elements have been calculated under the assumption that q' is massless.

`= 0 :` no.

`= 1 :` yes, but it is possible, as before, to switch off individual channels by the setting of `MDME` switches. Mass effects are taken into account, in a crude fashion, by rejecting events where kinematics becomes inconsistent when the q' mass is included.

`MSTP(11) :` (D=1) use of electron parton distribution in e^+e^- and ep interactions.

`= 0 :` no, i.e. electron carries the whole beam energy.

`= 1 :` yes, i.e. electron carries only a fraction of beam energy in agreement with next-to-leading electron parton-distribution function, thereby including the effects of initial-state bremsstrahlung.

`MSTP(12) :` (D=0) use of e^- ('sea', i.e. from $e \to \gamma \to e$), e^+, quark and gluon distribution functions inside an electron.

`= 0 :` off.

`= 1 :` on, provided that `MSTP(11)`≥ 1. Quark and gluon distributions are obtained by numerical convolution of the photon content inside an electron (as given by the bremsstrahlung spectrum of `MSTP(11)=1`) with the quark and gluon content inside a photon. The required numerical precision is set by `PARP(14)`. Since the need for numerical integration makes this option somewhat more time-consuming than ordinary parton-distribution evaluation, one should only use it when studying processes where it is needed.

Note: for all traditional photoproduction/DIS physics this option is superseded by the `'gamma/`*lepton*`'` option for `PYINIT` calls, but can still be of use for some less standard processes.

`MSTP(13) :` (D=1) choice of Q^2 range over which electrons are assumed to radiate photons;

affects normalization of e^- (sea), e^+, γ, quark and gluon distributions inside an electron for `MSTP(12)=1`.

`= 1 :` range set by Q^2 argument of parton-distribution-function call, i.e. by Q^2 scale of the hard interaction. Therefore parton distributions are proportional to $\ln(Q^2/m_e^2)$.

`= 2 :` range set by the user-determined $Q^2_{\rm max}$, given in `PARP(13)`. Parton distributions are assumed to be proportional to $\ln((Q^2_{\rm max}/m_e^2)(1-x)/x^2)$. This is normally most appropriate for photoproduction, where the electron is supposed to go undetected, i.e. scatter less than $Q^2_{\rm max}$.

Note: the choice of effective range is especially touchy for the quark and gluon distributions. An (almost) on-the-mass-shell photon has a VMD piece that dies away for a virtual photon. A simple convolution of distribution functions does not take this into account properly. Therefore the contribution from Q values above the ρ mass should be suppressed. A choice of $Q_{\rm max} \approx 1$ GeV is then appropriate for a photoproduction limit description of physics. See also note for `MSTP(12)`.

`MSTP(14)` : (D=30) structure of incoming photon beam or target. Historically, numbers up to 10 were set up for real photons, and subsequent ones have been added also to allow for virtual photon beams. The reason is that the existing options specify e.g. direct$\times$VMD, summing over the possibilities of which photon is direct and which VMD. This is allowed when the situation is symmetric, i.e. for two incoming real photons, but not if one is virtual. Some of the new options agree with previous ones, but are included to allow a more consistent pattern. Further options above 25 have been added also to include DIS processes.

`= 0 :` a photon is assumed to be point-like (a direct photon), i.e. can only interact in processes which explicitly contain the incoming photon, such as $f_i\gamma \to f_i g$ for γp interactions. In $\gamma\gamma$ interactions both photons are direct, i.e the main process is $\gamma\gamma \to f_i\bar{f}_i$.

`= 1 :` a photon is assumed to be resolved, i.e. can only interact through its constituent quarks and gluons, giving either high-$p_\perp$ parton–parton scatterings or low-$p_\perp$ events. Hard processes are calculated with the use of the full photon parton distributions. In $\gamma\gamma$ interactions both photons are resolved.

`= 2 :` a photon is assumed resolved, but only the VMD piece is included in the parton distributions, which therefore mainly are scaled-down versions of the ρ^0/π^0 ones. Both high-$p_\perp$ parton–parton scatterings and low-pT events are allowed. In $\gamma\gamma$ interactions both photons are VMD-like.

`= 3 :` a photon is assumed resolved, but only the anomalous piece of the photon parton distributions is included. (This event class is called either anomalous or GVMD; we will use both interchangeably, though the former is more relevant for high-$p_\perp$ phenomena and the latter for low-$p_\perp$ ones.) In $\gamma\gamma$ interactions both photons are anomalous.

`= 4 :` in $\gamma\gamma$ interactions one photon is direct and the other resolved. A typical process is thus $f_i\gamma \to f_i g$. Hard processes are calculated with the use of the full photon parton distributions for the resolved photon. Both possibilities of which photon is direct are included, in event topologies and in cross sections. This option cannot be used in configurations with only one incoming photon.

`= 5 :` in $\gamma\gamma$ interactions one photon is direct and the other VMD-like. Both possibilities of which photon is direct are included, in event topologies and in cross sections. This option cannot be used in configurations with only one incoming photon.

`= 6 :` in $\gamma\gamma$ interactions one photon is direct and the other anomalous. Both possibilities of which photon is direct are included, in event topologies and in cross sections. This option cannot be used in configurations with only one

incoming photon.

`= 7 :` in $\gamma\gamma$ interactions one photon is VMD-like and the other anomalous. Only high-$p_\perp$ parton–parton scatterings are allowed. Both possibilities of which photon is VMD-like are included, in event topologies and in cross sections. This option cannot be used in configurations with only one incoming photon.

`= 10 :` the VMD, direct and anomalous/GVMD components of the photon are automatically mixed. For γp interactions, this means an automatic mixture of the three classes 0, 2 and 3 above [Sch93, Sch93a], for $\gamma\gamma$ ones a mixture of the six classes 0, 2, 3, 5, 6 and 7 above [Sch94a]. Various restrictions exist for this option, as discussed in section 8.3.1.

`= 11 :` direct$\times$direct (see note 5); intended for virtual photons.

`= 12 :` direct$\times$VMD (i.e. first photon direct, second VMD); intended for virtual photons.

`= 13 :` direct$\times$anomalous; intended for virtual photons.

`= 14 :` VMD$\times$direct; intended for virtual photons.

`= 15 :` VMD$\times$VMD; intended for virtual photons.

`= 16 :` VMD$\times$anomalous; intended for virtual photons.

`= 17 :` anomalous$\times$direct; intended for virtual photons.

`= 18 :` anomalous$\times$VDM; intended for virtual photons.

`= 19 :` anomalous$\times$anomalous; intended for virtual photons.

`= 20 :` a mixture of the nine above components, 11–19, in the same spirit as `=10` provides a mixture for real gammas (or a virtual gamma on a hadron). For gamma-hadron, this option coincides with `=10`.

`= 21 :` direct$\times$direct (see note 5).

`= 22 :` direct$\times$resolved.

`= 23 :` resolved$\times$direct.

`= 24 :` resolved$\times$resolved.

`= 25 :` a mixture of the four above components, offering a simpler alternative to `=20` in cases where the parton distributions of the photon have not been split into VMD and anomalous components. For γ-hadron, only two components need be mixed.

`= 26 :` DIS$\times$VMD/p.

`= 27 :` DIS$\times$anomalous.

`= 28 :` VMD/p$\times$DIS.

`= 29 :` anomalous$\times$DIS.

`= 30 :` a mixture of all the 4 (for γ^*p) or 13 (for $\gamma^*\gamma^*$) components that are available, i.e. (the relevant ones of) 11–19 and 26–29 above; is as `=20` with the DIS processes mixed in.

Note 1: The `MSTP(14)` options apply for a photon defined by a `'gamma'` or `'gamma/`*lepton*`'` beam in the `PYINIT` call, but not to those photons implicitly obtained in a `'lepton'` beam with the `MSTP(12)=1` option. This latter approach to resolved photons is more primitive and is no longer recommended for QCD processes.

Note 2: for real photons our best understanding of how to mix event classes is provided by the option 10 above, which also can be obtained by combining three (for γp) or six (for $\gamma\gamma$) separate runs. In a simpler alternative the VMD and anomalous classes are joined into a single resolved class. Then γp physics only requires two separate runs, with 0 and 1, and $\gamma\gamma$ physics requires three, with 0, 1 and 4.

Note 3: most of the new options from 11 onwards are not needed and therefore not defined for ep collisions. The recommended 'best' value thus is `MSTP(14)=30`, which also is the new default value.

Note 4: as a consequence of the appearance of new event classes, the `MINT(122)` and

MSTI(9) codes are not the same for $\gamma^*\gamma^*$ events as for γp, γ^*p or $\gamma\gamma$ ones. Instead the code is $3(i_1-1)+i_2$, where i is 1 for direct, 2 for VMD and 3 for anomalous/GVMD and indices refer to the two incoming photons. For γ^*p code 4 is DIS, and for $\gamma^*\gamma^*$ codes 10–13 corresponds to the MSTP(14) codes 26–29. As before, MINT(122) and MSTI(9) are only defined when several processes are to be mixed, not when generating one at a time. Also the MINT(123) code is modified (not shown here).

Note 5: The direct$\times$direct event class excludes lepton pair production when run with the default MSEL=1 option (or MSEL=2), in order not to confuse users. You can obtain lepton pairs as well, e.g. by running with MSEL=0 and switching on the desired processes by hand.

Note 6: For all non-QCD processes, a photon is assumed unresolved when MSTP(14)= 10, 20 or 25. In principle, both the resolved and direct possibilities ought to be explored, but this mixing is not currently implemented, so picking direct at least will explore one of the two main alternatives rather than none. Resolved processes can be accessed by the more primitive machinery of having a lepton beam and MSTP(12)=1.

MSTP(15) : (D=0) possibility to modify the nature of the anomalous photon component (as used with the appropriate MSTP(14) options), in particular with respect to the scale choices and cut-offs of hard processes. These options are mainly intended for comparative studies and should not normally be touched. Some of the issues are discussed in [Sch93a], while others have only been used for internal studies and are undocumented.

- = 0 : none, i.e. the same treatment as for the VMD component.
- = 1 : evaluate the anomalous parton distributions at a scale $Q^2/\mathtt{PARP(17)}^2$.
- = 2 : as =1, but instead of PARP(17) use either PARP(81)/PARP(15) or PARP(82)/PARP(15), depending on MSTP(82) value.
- = 3 : evaluate the anomalous parton distribution functions of the photon as $f^{\gamma,\mathrm{anom}}(x,Q^2,p_0^2)-f^{\gamma,\mathrm{anom}}(x,Q^2,r^2Q^2)$ with $r=$PARP(17).
- = 4 : as =3, but instead of PARP(17) use either PARP(81)/PARP(15) or PARP(82)/PARP(15), depending on MSTP(82) value.
- = 5 : use larger $p_{\perp\mathrm{min}}$ for the anomalous component than for the VMD one, but otherwise no difference.

MSTP(16) : (D=1) choice of definition of the fractional momentum taken by a photon radiated off a lepton. Enters in the flux factor for the photon rate, and thereby in cross sections.

- = 0 : x, i.e. energy fraction in the rest frame of the event.
- = 1 : y, i.e. light-cone fraction.

MSTP(17) : (D=4) possibility of a extrafactor for processes involving resolved virtual photons, to approximately take into account the effects of longitudinal photons. Given on the form

$R = 1+\mathtt{PARP(165)}\, r(Q^2,\mu^2)\, f_L(y,Q^2)/f_T(y,Q^2)$.

Here the 1 represents the basic transverse contribution, PARP(165) is some arbitrary overall factor, and f_L/f_T the (known) ratio of longitudinal to transverse photon flux factors. The arbitrary function r depends on the photon virtuality Q^2 and the hard scale μ^2 of the process. See [Fri00] for a discussion of the options.

- = 0 : No contribution, i.e. $r=0$.
- = 1 : $r=4\mu^2Q^2/(\mu^2+Q^2)^2$.
- = 2 : $r=4Q^2/(\mu^2+Q^2)$.
- = 3 : $r=4Q^2/(m_\rho^2+Q^2)$.
- = 4 : $r=4m_V^2Q^2/(m_V^2+Q^2)^2$, where m_V is the vector meson mass for VMD and $2k_\perp$ for GVMD states. Since there is no μ dependence here (as well as for

=3 and =5) also minimum-bias cross sections are affected, where μ would be vanishing. Currently the actual vector meson mass in the VMD case is replaced by m_ρ, for simplicity.

= 5 : $r = 4Q^2/(m_V^2 + Q^2)$, with m_V and comments as above.

Note: For a photon given by the 'gamma/*lepton*' option in the PYINIT call, the y spectrum is dynamically generated and y is thus known from event to event. For a photon beam in the PYINIT call, y is unknown from the onset, and has to be provided by you if any longitudinal factor is to be included. So long as these values, in PARP(167) and PARP(168), are at their default values, 0, it is assumed they have not been set and thus the MSTP(17) and PARP(165) values are inactive.

MSTP(18) : (D=3) choice of $p_{\perp\mathrm{min}}$ for direct processes.

= 1 : same as for VMD and GVMD states, i.e. the $p_{\perp\mathrm{min}}(W^2)$ scale. Primarily intended for real photons.

= 2 : $p_{\perp\mathrm{min}}$ is chosen to be PARP(15), i.e. the original old behaviour proposed in [Sch93, Sch93a]. In that case, also parton distributions, jet cross sections and α_s values were dampened for small $p_\perp$, so it may not be easy to obtain full backwards compatibility with this option.

= 3 : as =1, but if the Q scale of the virtual photon is above the VMD/GVMD $p_{\perp\mathrm{min}}(W^2)$, $p_{\perp\mathrm{min}}$ is chosen equal to Q. This is part of the strategy to mix in DIS processes at $p_\perp$ below Q, e.g. in MSTP(14)=30.

MSTP(19) : (D=4) choice of partonic cross section in the DIS process 99.

= 0 : QPM answer $4\pi^2\alpha_{\mathrm{em}}/Q^2 \sum_{\mathrm{q}} e_{\mathrm{q}}^2 (xq(x,Q^2) + x\overline{q}(x,Q^2))$ (with parton distributions frozen below the lowest Q allowed in the parameterization). Note that this answer is divergent for $Q^2 \to 0$ and thus violates gauge invariance.

= 1 : QPM answer is modified by a factor $Q^2/(Q^2 + m_\rho^2)$ to provide a finite cross section in the $Q^2 \to 0$ limit. A minimal regularization recipe.

= 2 : QPM answer is modified by a factor $Q^4/(Q^2 + m_\rho^2)^2$ to provide a vanishing cross section in the $Q^2 \to 0$ limit. Appropriate if one assumes that the normal photoproduction description gives the total cross section for $Q^2 = 0$, without any DIS contribution.

= 3 : as = 2, but additionally suppression by a parameterized factor $f(W^2, Q^2)$ (different for γ^*p and $\gamma^*\gamma^*$) that avoids double-counting the direct-process region where $p_\perp > Q$. Shower evolution for DIS events is then also restricted to be at scales below Q, whereas evolution all the way up to W is allowed in the other options above.

= 4 : as = 3, but additionally include factor $1/(1-x)$ for conversion from F_2 to σ. This is formally required, but is only relevant for small W^2 and therefore often neglected.

MSTP(20) : (D=3) suppression of resolved (VMD or GVMD) cross sections, introduced to compensate for an overlap with DIS processes in the region of intermediate Q^2 and rather small W^2.

= 0 : no; used as is.

> 1 : yes, by a factor $(W^2/(W^2 + Q_1^2 + Q_2^2))^{\mathrm{MSTP(20)}}$ (where $Q_i^2 = 0$ for an incoming hadron).

Note: the suppression factor is joined with the dipole suppression stored in VINT(317) and VINT(318).

MSTP(21) : (D=1) nature of fermion–fermion scatterings simulated in process 10 by t-channel exchange.

= 0 : all off (!).

= 1 : full mixture of γ^*/Z^0 neutral current and $\mathrm{W}^\pm$ charged current.

= 2 : γ neutral current only.

= 3 : Z^0 neutral current only.
= 4 : γ^*/Z^0 neutral current only.
= 5 : $W^\pm$ charged current only.

MSTP(22) : (D=0) special override of normal Q^2 definition used for maximum of parton-shower evolution, intended for Deeply Inelastic Scattering in lepton–hadron events, see section 10.4.

MSTP(23) : (D=1) for Deeply Inelastic Scattering processes (10 and 83), this option allows the x and Q^2 of the original hard scattering to be retained by the final electron when showers are considered (with warnings as below; partly obsolete).

= 0 : no correction procedure, i.e. x and Q^2 of the scattered electron differ from the originally generated x and Q^2.

= 1 : post facto correction, i.e. the change of electron momentum, by initial and final QCD radiation, primordial $k_\perp$ and beam remnant treatment, is corrected for by a shuffling of momentum between the electron and hadron side in the final state. Only process 10 is corrected, while process 83 is not.

= 2 : as =1, except that both process 10 and 83 are treated. This option is dangerous, especially for top, since it may well be impossible to 'correct' in process 83: the standard DIS kinematics definitions are based on the assumption of massless quarks. Therefore infinite loops are not excluded.

Note 1: the correction procedure will fail for a fraction of the events, which are thus rejected (and new ones generated in their place). The correction option is not unambiguous, and should not be taken too seriously. For very small Q^2 values, the x is not exactly preserved even after this procedure.

Note 2: This switch does not affect the recommended DIS description obtained with a 'gamma/*lepton*' beam/target in PYINIT, where x and Q^2 are always conserved.

MSTP(31) : (D=1) parameterization of total, elastic and diffractive cross sections.

= 0 : everything is to be set by you yourself in the PYINT7 common block. For photoproduction, additionally you need to set VINT(281). Normally you would set these values once and for all before the PYINIT call, but if you run with variable energies (see MSTP(171)) you can also set it before each new PYEVNT call.

= 1 : Donnachie–Landshoff for total cross section [Don92], and Schuler–Sjöstrand for elastic and diffractive cross sections [Sch94, Sch93a].

MSTP(32) : (D=8) Q^2 definition in hard scattering for $2 \to 2$ processes. For resonance production Q^2 is always chosen to be $\hat{s} = m_R^2$, where m_R is the mass of the resonance. For gauge boson scattering processes $VV \to VV$ the W or Z^0 squared mass is used as scale in parton distributions. See PARP(34) for a possibility to modify the choice below by a multiplicative factor.

The newer options 6–10 are specifically intended for processes with incoming virtual photons. These are ordered from a 'minimal' dependence on the virtualities to a 'maximal' one, based on reasonable kinematics considerations. The old default value MSTP(32)=2 forms the starting point, with no dependence at all, and the new default is some intermediate choice. Notation is that P_1^2 and P_2^2 are the virtualities of the two incoming particles, $p_\perp$ the transverse momentum of the scattering process, and m_3 and m_4 the masses of the two outgoing partons. For a direct photon, P^2 is the photon virtuality and $x = 1$. For a resolved photon, P^2 still refers to the photon, rather than the unknown virtuality of the reacting parton in the photon, and x is the momentum fraction taken by this parton.

= 1 : $Q^2 = 2\hat{s}\hat{t}\hat{u}/(\hat{s}^2 + \hat{t}^2 + \hat{u}^2)$.

= 2 : $Q^2 = (m_{\perp 3}^2 + m_{\perp 4}^2)/2 = p_\perp^2 + (m_3^2 + m_4^2)/2$.

= 3 : $Q^2 = \min(-\hat{t}, -\hat{u})$.

= 4 : $Q^2 = \hat{s}$.
= 5 : $Q^2 = -\hat{t}$.
= 6 : $Q^2 = (1 + x_1 P_1^2/\hat{s} + x_2 P_2^2/\hat{s})(p_\perp^2 + m_3^2/2 + m_4^2/2)$.
= 7 : $Q^2 = (1 + P_1^2/\hat{s} + P_2^2/\hat{s})(p_\perp^2 + m_3^2/2 + m_4^2/2)$.
= 8 : $Q^2 = p_\perp^2 + (P_1^2 + P_2^2 + m_3^2 + m_4^2)/2$.
= 9 : $Q^2 = p_\perp^2 + P_1^2 + P_2^2 + m_3^2 + m_4^2$.
= 10 : $Q^2 = s$ (the full energy-squared of the process).

Note: options 6 and 7 are motivated by assuming that one wants a scale that interpolates between $\hat{t}$ for small $\hat{t}$ and $\hat{u}$ for small $\hat{u}$, such as $Q^2 = -\hat{t}\hat{u}/(\hat{t} + \hat{u})$. When kinematics for the $2 \to 2$ process is constructed as if an incoming photon is massless when it is not, it gives rise to a mismatch factor $1 + P^2/\hat{s}$ (neglecting the other masses) in this Q^2 definition, which is then what is used in option 7 (with the neglect of some small cross-terms when both photons are virtual). When a virtual photon is resolved, the virtuality of the incoming parton can be anything from xP^2 and upwards. So option 6 uses the smallest kinematically possible value, while 7 is more representative of the typical scale. Option 8 and 9 are more handwaving extensions of the default option, with 9 specially constructed to ensure that the Q^2 scale is always bigger than P^2.

`MSTP(33)` : (D=0) inclusion of K factors in hard cross sections for parton–parton interactions (i.e. for incoming quarks and gluons).
= 0 : none, i.e. $K = 1$.
= 1 : a common K factor is used, as stored in `PARP(31)`.
= 2 : separate factors are used for ordinary (`PARP(31)`) and colour annihilation graphs (`PARP(32)`).
= 3 : A K factor is introduced by a shift in the α_s Q^2 argument, $\alpha_s = \alpha_s(\texttt{PARP(33)}Q^2)$.

`MSTP(34)` : (D=1) use of interference term in matrix elements for QCD processes, see section 8.2.1.
= 0 : excluded (i.e. string-inspired matrix elements).
= 1 : included (i.e. conventional QCD matrix elements).

Note: for the option `MSTP(34)=1`, i.e. interference terms included, these terms are divided between the different possible colour configurations according to the pole structure of the (string-inspired) matrix elements for the different colour configurations.

`MSTP(35)` : (D=0) threshold behaviour for heavy-flavour production, i.e. `ISUB` = 81, 82, 84, 85, and also for Z and Z′ decays. The non-standard options are mainly intended for top, but can be used, with less theoretical reliability, also for charm and bottom (for Z and Z′ only top and heavier flavours are affected). The threshold factors are given in eqs. (137) and (138).
= 0 : naïve lowest-order matrix-element behaviour.
= 1 : enhancement or suppression close to threshold, according to the colour structure of the process. The α_s value appearing in the threshold factor (which is not the same as the α_s of the lowest-order $2 \to 2$ process) is taken to be fixed at the value given in `PARP(35)`. The threshold factor used in an event is stored in `PARI(81)`.
= 2 : as =1, but the α_s value appearing in the threshold factor is taken to be running, with argument $Q^2 = m_Q\sqrt{(\hat{m} - 2m_Q)^2 + \Gamma_Q^2}$. Here m_Q is the nominal heavy-quark mass, Γ_Q is the width of the heavy-quark-mass distribution, and $\hat{m}$ is the invariant mass of the heavy-quark pair. The Γ_Q value has to be stored by you in `PARP(36)`. The regularization of α_s at low Q^2 is given by `MSTP(36)`.

`MSTP(36)` : (D=2) regularization of α_s in the limit $Q^2 \to 0$ for the threshold factor obtainable

in the MSTP(35)=2 option; see MSTU(115) for a list of the possibilities.

MSTP(37) : (D=1) inclusion of running quark masses in Higgs production ($q\overline{q} \to h^0$) and decay ($h^0 \to q\overline{q}$) couplings, obtained by calls to the PYMRUN function. Also included for charged Higgs and technipion production and decay.

= 0 : not included, i.e. fixed quark masses are used according to the values in the PMAS array.

= 1 : included, with running starting from the value given in the PMAS array, at a Q_0 scale of PARP(37) times the quark mass itself, up to a Q scale given by the Higgs mass. This option only works when α_s is allowed to run (so one can define a Λ value). Therefore it is only applied if additionally MSTP(2)≥ 1.

MSTP(38) : (D=5) handling of quark loop masses in the box graphs $gg \to \gamma\gamma$ and $gg \to g\gamma$, and in the Higgs production loop graphs $q\overline{q} \to gh^0$, $qg \to qh^0$ and $gg \to gh^0$, and their equivalents with H^0 or A^0 instead of h^0.

= 0 : for $gg \to \gamma\gamma$ and $gg \to g\gamma$ the program will for each flavour automatically choose the massless approximation for light quarks and the full massive formulae for heavy quarks, with a dividing line between light and heavy quarks that depends on the actual $\hat{s}$ scale. For Higgs production, all quark loop contributions are included with the proper masses. This option is then correct only in the Standard Model Higgs scenario, and should not be used e.g. in the MSSM.

$\geq$1 : for $gg \to \gamma\gamma$ and $gg \to g\gamma$ the program will use the massless approximation throughout, assuming the presence of MSTP(38) effectively massless quark species (however, at most 8). Normally one would use =5 for the inclusion of all quarks up to bottom, and =6 to include top as well. For Higgs production, the approximate expressions derived in the $m_t \to \infty$ limit are used, rescaled to match the correct $gg \to h^0/H^0/A^0$ cross sections. This procedure should work, approximately, also for non-standard Higgs particles.

Warning: for =0, numerical instabilities may arise in $gg \to \gamma\gamma$ and $gg \to g\gamma$ for scattering at small angles. You are therefore recommended in this case to set CKIN(27) and CKIN(28) so as to exclude the range of scattering angles that is not of interest anyway. Numerical problems may also occur for Higgs production with =0, and additionally the lengthy expressions make the code error-prone.

MSTP(39) : (D=2) choice of Q^2 scale for parton distributions and initial state parton showers in processes $gg \to Q\overline{Q}h$ or $q\overline{q} \to Q\overline{Q}h$.

= 1 : m_Q^2.

= 2 : $\max(m_{\perp Q}^2, m_{\perp \overline{Q}}^2) = m_Q^2 + \max(p_{\perp Q}^2, p_{\perp \overline{Q}}^2)$.

= 3 : m_h^2, where m_h is the actual Higgs mass of the event, fluctuating from one event to the next.

= 4 : $\hat{s} = (p_h + p_Q + p_{\overline{Q}})^2$.

= 5 : m_h^2, where m_h is the nominal, fixed Higgs mass.

MSTP(40) : (D=0) option for Coulomb correction in process 25, W^+W^- pair production, see [Kho96]. The value of the Coulomb correction factor for each event is stored in VINT(95).

= 0 : 'no Coulomb'. Is the often-used reference point.

= 1 : 'unstable Coulomb', gives the correct first-order expression valid in the non-relativistic limit. Is the reasonable option to use as a 'best bet' description of LEP 2 physics.

= 2 : 'second-order Coulomb' gives the correct second-order expression valid in the non-relativistic limit. In principle this is even better than =1, but the differences are negligible and computer time does go up because of the need for a numerical integration in the weight factor.

= 3 : 'dampened Coulomb', where the unstable Coulomb expression has been mod-

ified by a $(1-\beta)^2$ factor in front of the arctan term. This is intended as an alternative to `=1` within the band of our uncertainty in the relativistic limit.

`= 4 :` 'stable Coulomb', i.e. effects are calculated as if the W's were stable. Is incorrect, and mainly intended for comparison purposes.

Note : Unfortunately the W's at LEP 2 are not in the non-relativistic limit, so the separation of Coulomb effects from other radiative corrections is not gauge invariant. The options above should therefore be viewed as indicative only, not as the ultimate answer.

`MSTP(41)` : (D=2) master switch for all resonance decays (Z^0, $W^\pm$, t, h^0, Z'^0, $W'^\pm$, H^0, A^0, $H^\pm$, L_Q, R^0, d^*, u^*, ...).

`= 0 :` all off.

`= 1 :` all on.

`= 2 :` on or off depending on their individual `MDCY` values.

Note: also for `MSTP(41)=1` it is possible to switch off the decays of specific resonances by using the `MDCY(KC,1)` switches in PYTHIA. However, since the `MDCY` values are overwritten in the `PYINIT` call when `MSTP(41)=1` (or 0), individual resonances should then be switched off after the `PYINIT` call.

Warning: for top, leptoquark and other colour-carrying resonances it is dangerous to switch off decays if one later on intends to let them decay (with `PYEXEC`); see section 8.6.4.

`MSTP(42)` : (D=1) mass treatment in $2 \to 2$ processes, where the final-state resonances have finite width (see `PARP(41)`). (Does not apply for the production of a single *s*-channel resonance, where the mass appears explicitly in the cross section of the process, and thus is always selected with width.)

`= 0 :` particles are put on the mass shell.

`= 1 :` mass generated according to a Breit–Wigner.

`MSTP(43)` : (D=3) treatment of Z^0/γ^* interference in matrix elements. So far implemented in subprocesses 1, 15, 19, 22, 30 and 35; in other processes what is called a Z^0 is really a Z^0 only, without the γ^* piece.

`= 1 :` only γ^* included.

`= 2 :` only Z^0 included.

`= 3 :` complete Z^0/γ^* structure (with interference) included.

`MSTP(44)` : (D=7) treatment of $Z'^0/Z^0/\gamma^*$ interference in matrix elements.

`= 1 :` only γ^* included.

`= 2 :` only Z^0 included.

`= 3 :` only Z'^0 included.

`= 4 :` only Z^0/γ^* (with interference) included.

`= 5 :` only Z'^0/γ^* (with interference) included.

`= 6 :` only Z'^0/Z^0 (with interference) included.

`= 7 :` complete $Z'^0/Z^0/\gamma^*$ structure (with interference) included.

`MSTP(45)` : (D=3) treatment of $WW \to WW$ structure (`ISUB` = 77).

`= 1 :` only $W^+W^+ \to W^+W^+$ and $W^-W^- \to W^-W^-$ included.

`= 2 :` only $W^+W^- \to W^+W^-$ included.

`= 3 :` all charge combinations $WW \to WW$ included.

`MSTP(46)` : (D=1) treatment of $VV \to V'V'$ structures (`ISUB` = 71–77), where V represents a longitudinal gauge boson.

`= 0 :` only *s*-channel Higgs exchange included (where existing). With this option, subprocesses 71–72 and 76–77 will essentially be equivalent to subprocesses 5 and 8, respectively, with the proper decay channels (i.e. only Z^0Z^0 or W^+W^-) set via `MDME`. The description obtained for subprocesses 5 and 8 in this case is more sophisticated, however; see section 8.5.2.

`= 1 :` all graphs contributing to $VV \to V'V'$ processes are included.

= 2 : only graphs not involving Higgs exchange (either in *s*, *t* or *u* channel) are included; this option then gives the naïve behaviour one would expect if no Higgs exists, including unphysical unitarity violations at high energies.

= 3 : the strongly interacting Higgs-like model of Dobado, Herrero and Terron [Dob91] with Padé unitarization. Note that to use this option it is necessary to set the Higgs mass to a large number like 20 TeV (i.e. PMAS(25,1)=20000). The parameter ν is stored in PARP(44), but should not have to be changed.

= 4 : as =3, but with K-matrix unitarization [Dob91].

= 5 : the strongly interacting QCD-like model of Dobado, Herrero and Terron [Dob91] with Padé unitarization. The parameter ν is stored in PARP(44), but should not have to be changed. The effective techni-ρ mass in this model is stored in PARP(45); by default it is 2054 GeV, which is the expected value for three technicolors, based on scaling up the ordinary ρ mass appropriately.

= 6 : as =5, but with K-matrix unitarization [Dob91]. While PARP(45) still is a parameter of the model, this type of unitarization does not give rise to a resonance at a mass of PARP(45).

MSTP(47) : (D=1) (C) angular orientation of decay products of resonances (Z^0, $W^\pm$, t, h^0, Z'^0, $W'^\pm$, etc.), either when produced singly or in pairs (also from an h^0 decay), or in combination with a single quark, gluon or photon.

= 0 : independent decay of each resonance, isotropic in c.m. frame of the resonance.

= 1 : correlated decay angular distributions according to proper matrix elements, to the extent these are implemented.

MSTP(48) : (D=0) (C) switch for the treatment of γ^*/Z^0 decay for process 1 in e^+e^- events.

= 0 : normal machinery.

= 1 : if the decay of the Z^0 is to either of the five lighter quarks, d, u, s, c or b, the special treatment of Z^0 decay is accessed, including matrix element options, according to section 6.1.

This option is based on the machinery of the PYEEVT and associated routines when it comes to the description of QCD multijet structure and the angular orientation of jets, but relies on the normal PYEVNT machinery for everything else: cross section calculation, initial state photon radiation, flavour composition of decays (i.e. information on channels allowed), etc.

The initial state has to be e^+e^-; forward-backward asymmetries would not come out right for quark-annihilation production of the γ^*/Z^0 and therefore the machinery defaults to the standard one in such cases.

You can set the behaviour for the MSTP(48) option using the normal matrix element related switches. This especially means MSTJ(101) for the selection of first- or second-order matrix elements (=1 and =2, respectively). Further selectivity is obtained with the switches and parameters MSTJ(102) (for the angular orientation part only), MSTJ(103) (except the production threshold factor part), MSTJ(106), MSTJ(108) - MSTJ(111), PARJ(121), PARJ(122), and PARJ(125) - PARJ(129). Information can be read from MSTJ(120), MSTJ(121), PARJ(150), PARJ(152) - PARJ(156), PARJ(168), PARJ(169), PARJ(171).

The other e^+e^- switches and parameters should not be touched. In most cases they are simply not accessed, since the related part is handled by the PYEVNT machinery instead. In other cases they could give incorrect or misleading results. Beam polarization as set by PARJ(131) - PARJ(134), for instance, is only included for the angular orientation, but is missing for the cross section information. PARJ(123) and PARJ(124) for the Z^0 mass and width are set in the PYINIT call based on the input mass and calculated widths.

The cross section calculation is unaffected by the matrix element machinery. Thus also for negative `MSTJ(101)` values, where only specific jet multiplicities are generated, the `PYSTAT` cross section is the full one.

`MSTP(49)` : (D=1) assumed variation of the Higgs width to boson pairs, as a function of the actual mass $\hat{m} = \sqrt{\hat{s}}$ and the nominal mass $m_{\rm h}$. The switch applies both to $\rm h^0$, $\rm H^0$, $\rm A^0$ and $\rm H^\pm$ decays.

- `= 0` : the width is proportional to $\hat{m}^3$; thus the high-mass tail of the Breit-Wigner is enhanced.
- `= 1` : the width is proportional to $m_{\rm h}^2\hat{m}$. For a fixed Higgs mass $m_{\rm h}$ this means a width variation across the Breit-Wigner more in accord with other resonances (such as the $\rm Z^0$). This alternative gives more emphasis to the low-mass tail, where the parton distributions are peaked (for hadron colliders). This option is favoured by resummation studies [Sey95a].

Note 1: the partial width of a Higgs to a fermion pair is always taken to be proportional to the actual Higgs mass $\hat{m}$, irrespectively of `MSTP(49)`.

Note 2: this switch does not affect processes 71–77, where a fixed Higgs width is used in order to control cancellation of divergences.

`MSTP(50)` : (D=0) Switch to allow or not longitudinally polarized incoming beams, with the two polarizations stored in `PARJ(131)` and `PARJ(132)`, respectively. Most cross section expressions with polarization reduce to the unpolarized behaviour for the default `PARJ(131)=PARJ(132)=0`, and then this switch is not implemented. Currently it is only used in process 25, $\rm f\bar{f} \to W^+W^-$, for reasons explained in subsection 8.8.

- `= 0` : no polarization effects, no matter what `PARJ(131)` and `PARJ(132)` values are set.
- `= 1` : include polarization information in the cross section of the process and for angular correlations.

`MSTP(51)` : (D=7) choice of proton parton-distribution set; see also `MSTP(52)`.

- `= 1` : CTEQ 3L (leading order).
- `= 2` : CTEQ 3M ($\overline{\rm MS}$).
- `= 3` : CTEQ 3D (DIS).
- `= 4` : GRV 94L (leading order).
- `= 5` : GRV 94M ($\overline{\rm MS}$).
- `= 6` : GRV 94D (DIS).
- `= 7` : CTEQ 5L (leading order).
- `= 8` : CTEQ 5M1 ($\overline{\rm MS}$; slightly update version of CTEQ 5M).
- `= 11` : GRV 92L (leading order).
- `= 12` : EHLQ set 1 (leading order; 1986 updated version).
- `= 13` : EHLQ set 2 (leading order; 1986 updated version).
- `= 14` : Duke–Owens set 1 (leading order).
- `= 15` : Duke–Owens set 2 (leading order).
- `= 16` : simple ansatz with all parton distributions of the form c/x, with c some constant; intended for internal debug use only.

Note 1: distributions 11–15 are obsolete and should not be used for current physics studies. They are only implemented to have some sets in common between PYTHIA 5 and 6, for cross-checks.

Note 2: since all parameterizations have some region of applicability, the parton distributions are assumed frozen below the lowest Q^2 covered by the parameterizations. In some cases, evolution is also frozed above the maximum Q^2.

`MSTP(52)` : (D=1) choice of proton parton-distribution-function library.

- `= 1` : the internal PYTHIA one, with parton distributions according to the `MSTP(51)` above.
- `= 2` : the PDFLIB one [Plo93], with the PDFLIB (version 4) `NGROUP` and `NSET` num-

bers to be given as `MSTP(51) = 1000×NGROUP + NSET`.

Note: to make use of option 2, it is necessary to link PDFLIB. Additionally, on most computers, the three dummy routines `PDFSET`, `STRUCTM` and (for virtual photons) `STRUCTP` at the end of the PYTHIA file should be removed or commented out.

Warning: For external parton distribution libraries, PYTHIA does not check whether `MSTP(51)` corresponds to a valid code, or if special x and Q^2 restrictions exist for a given set, such that crazy values could be returned. This puts an extra responsibility on you.

`MSTP(53)` : (D=3) choice of pion parton-distribution set; see also `MSTP(54)`.

- `= 1` : Owens set 1.
- `= 2` : Owens set 2.
- `= 3` : GRV LO (updated version).

`MSTP(54)` : (D=1) choice of pion parton-distribution-function library.

- `= 1` : the internal PYTHIA one, with parton distributions according to the `MSTP(53)` above.
- `= 2` : the PDFLIB one [Plo93], with the PDFLIB (version 4) `NGROUP` and `NSET` numbers to be given as `MSTP(53) = 1000×NGROUP + NSET`.

Note: to make use of option 2, it is necessary to link PDFLIB. Additionally, on most computers, the three dummy routines `PDFSET`, `STRUCTM` and `STRUCTP` at the end of the PYTHIA file should be removed or commented out.

Warning: For external parton distribution libraries, PYTHIA does not check whether `MSTP(53)` corresponds to a valid code, or if special x and Q^2 restrictions exist for a given set, such that crazy values could be returned. This puts an extra responsibility on you.

`MSTP(55)` : (D=5) choice of the parton-distribution set of the photon; see also `MSTP(56)` and `MSTP(60)`.

- `= 1` : Drees–Grassie.
- `= 5` : SaS 1D (in DIS scheme, with $Q_0 = 0.6$ GeV).
- `= 6` : SaS 1M (in $\overline{\mathrm{MS}}$ scheme, with $Q_0 = 0.6$ GeV).
- `= 7` : SaS 2D (in DIS scheme, with $Q_0 = 2$ GeV).
- `= 8` : SaS 2M (in $\overline{\mathrm{MS}}$ scheme, with $Q_0 = 2$ GeV).
- `= 9` : SaS 1D (in DIS scheme, with $Q_0 = 0.6$ GeV).
- `= 10` : SaS 1M (in $\overline{\mathrm{MS}}$ scheme, with $Q_0 = 0.6$ GeV).
- `= 11` : SaS 2D (in DIS scheme, with $Q_0 = 2$ GeV).
- `= 12` : SaS 2M (in $\overline{\mathrm{MS}}$ scheme, with $Q_0 = 2$ GeV).

Note 1: sets 5–8 use the parton distributions of the respective set, and nothing else. These are appropriate for most applications, e.g. jet production in γp and $\gamma\gamma$ collisions. Sets 9–12 instead are appropriate for $\gamma^*\gamma$ processes, i.e. DIS scattering on a photon, as measured in F_2^γ. Here the anomalous contribution for c and b quarks are handled by the Bethe-Heitler formulae, and the direct term is artificially lumped with the anomalous one, so that the event simulation more closely agrees with what will be experimentally observed in these processes. The agreement with the F_2^γ parameterization is still not perfect, e.g. in the treatment of heavy flavours close to threshold.

Note 2: Sets 5–12 contain both VMD pieces and anomalous pieces, separately parameterized. Therefore the respective piece is automatically called, whatever `MSTP(14)` value is used to select only a part of the allowed photon interactions. For other sets (set 1 above or PDFLIB sets), usually there is no corresponding subdivision. Then an option like `MSTP(14)=2` (VMD part of photon only) is based on a rescaling of the pion distributions, while `MSTP(14)=3` gives the SaS anomalous parameterization.

Note 3: Formally speaking, the k_0 (or p_0) cut-off in `PARP(15)` need not be set in any

relation to the Q_0 cut-off scales used by the various parameterizations. Indeed, due to the familiar scale choice ambiguity problem, there could well be some offset between the two. However, unless you know what you are doing, it is recommended that you let the two agree, i.e. set `PARP(15)=0.6` for the SaS 1 sets and `=2.` for the SaS 2 sets.

`MSTP(56)` : (D=1) choice of photon parton-distribution-function library.

`= 1` : the internal PYTHIA one, with parton distributions according to the `MSTP(55)` above.

`= 2` : the PDFLIB one [Plo93], with the PDFLIB (version 4) `NGROUP` and `NSET` numbers to be given as `MSTP(55) = 1000×NGROUP + NSET`. When the VMD and anomalous parts of the photon are split, like for `MSTP(14)=10`, it is necessary to specify pion set to be used for the VMD component, in `MSTP(53)` and `MSTP(54)`, while `MSTP(55)` here is irrelevant.

`= 3` : when the parton distributions of the anomalous photon are requested, the homogeneous solution is provided, evolved from a starting value `PARP(15)` to the requested Q scale. The homogeneous solution is normalized so that the net momentum is unity, i.e. any factors of $\alpha_{\rm em}/2\pi$ and charge have been left out. The flavour of the original q is given in `MSTP(55)` (1, 2, 3, 4 or 5 for d, u, s, c or b); the value 0 gives a mixture according to squared charge, with the exception that c and b are only allowed above the respective mass threshold ($Q > m_{\rm q}$). The four-flavour Λ value is assumed given in `PARP(1)`; it is automatically recalculated for 3 or 5 flavours at thresholds. This option is not intended for standard event generation, but is useful for some theoretical studies.

Note: to make use of option 2, it is necessary to link PDFLIB. Additionally, on most computers, the three dummy routines `PDFSET`, `STRUCTM` and `STRUCTP` at the end of the PYTHIA file should be removed or commented out.

Warning: For external parton-distribution libraries, PYTHIA does not check whether `MSTP(55)` corresponds to a valid code, or if special x and Q^2 restrictions exist for a given set, such that crazy values could be returned. This puts an extra responsibility on you.

`MSTP(57)` : (D=1) choice of Q^2 dependence in parton-distribution functions.

`= 0` : parton distributions are evaluated at nominal lower cut-off value Q_0^2, i.e. are made Q^2-independent.

`= 1` : the parameterized Q^2 dependence is used.

`= 2` : the parameterized parton-distribution behaviour is kept at large Q^2 and x, but modified at small Q^2 and/or x, so that parton distributions vanish in the limit $Q^2 \to 0$ and have a theoretically motivated small-x shape [Sch93a]. This option is only valid for the p and n. It is obsolete within the current `'gamma/`*lepton*`'` framework.

`= 3` : as `=2`, except that also the $\pi^\pm$ is modified in a corresponding manner. A VMD photon is not mapped to a pion, but is treated with the same photon parton distributions as for other `MSTP(57)` values, but with properly modified behaviour for small x or Q^2. This option is obsolete within the current `'gamma/`*lepton*`'` framework.

`MSTP(58)` : (D=min(5, 2×`MSTP(1)`)) maximum number of quark flavours used in parton distributions, and thus also for initial-state space-like showers. If some distributions (notably t) are absent in the parameterization selected in `MSTP(51)`, these are obviously automatically excluded.

`MSTP(59)` : (D=1) choice of electron-inside-electron parton distribution.

`= 1` : the recommended standard for LEP 1, next-to-leading exponentiated, see [Kle89], p. 34.

`= 2` : the recommended ‘β’ scheme for LEP 2, also next-to-leading exponentiated,

see [Bee96], p. 130.

`MSTP(60)` : (D=7) extension of the SaS real-photon distributions to off-shell photons, especially for the anomalous component. See [Sch96] for an explanation of the options. The starting point is the expression in eq. (48), which requires a numerical integration of the anomalous component, however, and therefore is not convenient. Approximately, the dipole damping factor can be removed and compensated by a suitably shifted lower integration limit, whereafter the integral simplifies. Different 'goodness' criteria for the choice of the shifted lower limit is represented by the options 2–7 below.

- `= 1` : dipole dampening by integration; very time-consuming.
- `= 2` : $P_0^2 = \max(Q_0^2, P^2)$.
- `= 3` : $P'^2_0 = Q_0^2 + P^2$.
- `= 4` : $P_{\rm eff}$ that preserves momentum sum.
- `= 5` : $P_{\rm int}$ that preserves momentum and average evolution range.
- `= 6` : $P_{\rm eff}$, matched to P_0 in $P^2 \to Q^2$ limit.
- `= 7` : $P_{\rm int}$, matched to P_0 in $P^2 \to Q^2$ limit.

`MSTP(61)` : (D=1) (C) master switch for initial-state QCD and QED radiation.

- `= 0` : off.
- `= 1` : on.

`MSTP(62) - MSTP(69)` : (C) further switches for initial-state radiation, see section 10.4.

`MSTP(71)` : (D=1) (C) master switch for final-state QCD and QED radiation.

- `= 0` : off.
- `= 1` : on.

Note: additional switches (e.g. for conventional/coherent showers) are available in `MSTJ(38) - MSTJ(50)` and `PARJ(80) - PARJ(90)`, see section 10.4.

`MSTP(81)` : (D=1) master switch for multiple interactions.

- `= 0` : off.
- `= 1` : on.

`MSTP(82) - MSTP(86)` : further switches for multiple interactions, see section 11.4.

`MSTP(91) - MSTP(94)` : switches for beam remnant treatment, see section 11.4.

`MSTP(101)` : (D=3) (C) structure of diffractive system.

- `= 1` : forward moving diquark + interacting quark.
- `= 2` : forward moving diquark + quark joined via interacting gluon ('hairpin' configuration).
- `= 3` : a mixture of the two options above, with a fraction `PARP(101)` of the former type.

`MSTP(102)` : (D=1) (C) decay of a ρ^0 meson produced by 'elastic' scattering of an incoming γ, as in $\gamma p \to \rho^0 p$, or the same with the hadron diffractively excited.

- `= 0` : the ρ^0 is allowed to decay isotropically, like any other ρ^0.
- `= 1` : the decay $\rho^0 \to \pi^+\pi^-$ is done with an angular distribution proportional to $\sin^2\theta$ in its rest frame, where the z axis is given by the direction of motion of the ρ^0. The ρ^0 decay is then done as part of the hard process, i.e. also when `MSTP(111)=0`.

`MSTP(110)` : (D=0) switch to allow some or all resonance widths to be modified by the factor `PARP(110)`. This is not intended for serious physics studies. The main application is rather to generate events with an artificially narrow resonance width in order to study the detector-related smearing effects on the mass resolution.

- `> 0` : rescale the particular resonance with `KF = MSTP(110)`. If the resonance has an antiparticle, this one is affected as well.
- `= -1` : rescale all resonances, except t, $\bar{\rm t}$, Z^0 and $W^\pm$.
- `= -2` : rescale all resonances.

Warning: Only resonances with a width evaluated by `PYWIDT` are affected, and prefer-

entially then those with MWID value 1 or 3. For other resonances the appearance of effects or not depends on how the cross sections have been implemented. So it is important to check that indeed the mass distribution is affected as expected. Also beware that, if a sequential decay chain is involved, the scaling may become more complicated. Furthermore, depending on implementational details, a cross section may or may not scale with PARP(110) (quite apart from differences related to the convolution with parton distributions etc.). All in all, it is then an option to be used only with open eyes, and for very specific applications.

MSTP(111) : (D=1) (C) master switch for fragmentation and decay, as obtained with a PYEXEC call.

= 0 : off.

= 1 : on.

= -1 : only choose kinematical variables for hard scattering, i.e. no jets are defined. This is useful, for instance, to calculate cross sections (by Monte Carlo integration) without wanting to simulate events; information obtained with PYSTAT(1) will be correct.

MSTP(112) : (D=1) (C) cuts on partonic events; only affects an exceedingly tiny fraction of events. Normally this concerns what happens in the PYPREP routine, if a colour singlet subsystem has a very small invariant mass and attempts to collapse it to a single particle fail, see section 12.4.1.

= 0 : no cuts (can be used only with independent fragmentation, at least in principle).

= 1 : string cuts (as normally required for fragmentation).

MSTP(113) : (D=1) (C) recalculation of energies of partons from their momenta and masses, to be done immediately before and after fragmentation, to partly compensate for some numerical problems appearing at high energies.

= 0 : not performed.

= 1 : performed.

MSTP(115) : (D=0) (C) choice of colour rearrangement scenario for process 25, W^+W^- pair production, when both W's decay hadronically. (Also works for process 22, Z^0Z^0 production, except when the Z's are allowed to fluctuate to very small masses.) See section 12.4.2 for details.

= 0 : no reconnection.

= 1 : scenario I, reconnection inspired by a type I superconductor, with the reconnection probability related to the overlap volume in space and time between the W^+ and W^- strings. Related parameters are found in PARP(115) - PARP(119), with PARP(117) of special interest.

= 2 : scenario II, reconnection inspired by a type II superconductor, with reconnection possible when two string cores cross. Related parameter in PARP(115).

= 3 : scenario II', as model II but with the additional requirement that a reconnection will only occur if the total string length is reduced by it.

= 5 : the GH scenario, where the reconnection can occur that reduces the total string length (λ measure) most. PARP(120) gives the fraction of such event where a reconnection is actually made; since almost all events could allow a reconnection that would reduce the string length, PARP(120) is almost the same as the reconnection probability.

= 11 : the intermediate scenario, where a reconnection is made at the 'origin' of events, based on the subdivision of all radiation of a $q\overline{q}$ system as coming either from the q or the $\overline{q}$. PARP(120) gives the assumed probability that a reconnection will occur. A somewhat simpleminded model, but not quite unrealistic.

= 12 : the instantaneous scenario, where a reconnection is allowed to occur before

the parton showers, and showering is performed inside the reconnected systems with maximum virtuality set by the mass of the reconnected systems. PARP(120) gives the assumed probability that a reconnection will occur. Is completely unrealistic, but useful as an extreme example with very large effects.

MSTP(121) : (D=0) calculation of kinematics selection coefficients and differential cross section maxima for included (by you or default) subprocesses.

= 0 : not known; to be calculated at initialization.

= 1 : not known; to be calculated at initialization; however, the maximum value then obtained is to be multiplied by PARP(121) (this may be useful if a violation factor has been observed in a previous run of the same kind).

= 2 : known; kinematics selection coefficients stored by you in COEF(ISUB,J) (J = 1–20) in common block PYINT2 and maximum of the corresponding differential cross section times Jacobians in XSEC(ISUB,1) in common block PYINT5. This is to be done for each included subprocess ISUB before initialization, with the sum of all XSEC(ISUB,1) values, except for ISUB = 95, stored in XSEC(0,1).

MSTP(122) : (D=1) initialization and differential cross section maximization print-out. Also, less importantly, level of information on where in phase space a cross section maximum has been violated during the run.

= 0 : none.

= 1 : short message at initialization; only when an error (i.e. not a warning) is generated during the run.

= 2 : detailed message, including full maximization., at initialization; always during run.

MSTP(123) : (D=2) reaction to violation of maximum differential cross section or to occurence of negative differential cross sections (except when allowed for external processes, i.e. when IDWTUP < 0).

= 0 : stop generation, print message.

= 1 : continue generation, print message for each subsequently larger violation.

= 2 : as =1, but also increase value of maximum.

MSTP(124) : (D=1) (C) frame for presentation of event.

= 1 : as specified in PYINIT.

= 2 : c.m. frame of incoming particles.

= 3 : hadronic c.m. frame for DIS events, with warnings as given for PYFRAM.

MSTP(125) : (D=1) (C) documentation of partonic process, see section 5.3.2 for details.

= 0 : only list ultimate string/particle configuration.

= 1 : additionally list short summary of the hard process.

= 2 : list complete documentation of intermediate steps of parton-shower evolution.

MSTP(126) : (D=100) number of lines at the beginning of event record that are reserved for event-history information; see section 5.3.2. This value should never be reduced, but may be increased at a later date if more complicated processes are included.

MSTP(127) : (D=0) possibility to continue run even if none of the requested processes have non-vanishing cross sections.

= 0 : no, the run will be stopped in the PYINIT call.

= 1 : yes, the PYINIT execution will finish normally, but with the flag MSTI(53)=1 set to signal the problem. If nevertheless PYEVNT is called after this, the run will be stopped, since no events can be generated. If instead a new PYINIT call is made, with changed conditions (e.g. modified supersymmetry parameters in a SUSY run), it may now become possible to initialize normally and generate events.

MSTP(128) : (D=0) storing of copy of resonance decay products in the documentation section of the event record, and mother pointer (K(I,3)) relation of the actual resonance

decay products (stored in the main section of the event record) to the documentation copy.

`= 0 :` products are stored also in the documentation section, and each product stored in the main section points back to the corresponding entry in the documentation section.

`= 1 :` products are stored also in the documentation section, but the products stored in the main section point back to the decaying resonance copy in the main section.

`= 2 :` products are not stored in the documentation section; the products stored in the main section point back to the decaying resonance copy in the main section.

`MSTP(129) :` (D=10) for the maximization of $2 \to 3$ processes (`ISET(ISUB)=5`) each phase-space point in τ, y and τ' is tested `MSTP(129)` times in the other dimensions (at randomly selected points) to determine the effective maximum in the (τ, y, τ') point.

`MSTP(131) :` (D=0) master switch for pile-up events, i.e. several independent hadron–hadron interactions generated in the same bunch–bunch crossing, with the events following one after the other in the event record. See subsection 11.3 for details.

`= 0 :` off, i.e. only one event is generated at a time.

`= 1 :` on, i.e. several events are allowed in the same event record. Information on the processes generated may be found in `MSTI(41) - MSTI(50)`.

`MSTP(132) - MSTP(134) :` further switches for pile-up events, see section 11.4.

`MSTP(141) :` (D=0) calling of `PYKCUT` in the event-generation chain, for inclusion of user-specified cuts.

`= 0 :` not called.

`= 1 :` called.

`MSTP(142) :` (D=0) calling of `PYEVWT` in the event-generation chain, either to give weighted events or to modify standard cross sections. See `PYEVWT` description in section 9.1 for further details.

`= 0 :` not called.

`= 1 :` called; the distribution of events among subprocesses and in kinematics variables is modified by the factor `WTXS`, set by you in the `PYEVWT` call, but events come with a compensating weight `PARI(10)=1./WTXS`, such that total cross sections are unchanged.

`= 2 :` called; the cross section itself is modified by the factor `WTXS`, set by you in the `PYEVWT` call.

`MSTP(151) :` (D=0) introduce smeared position of primary vertex of events.

`= 0 :` no, i.e. the primary vertex of each event is at the origin.

`= 1 :` yes, with Gaussian distributions separately in x, y, z and t. The respective widths of the Gaussians have to be given in `PARP(151) - PARP(154)`. Also pile-up events obtain separate primary vertices. No provisions are made for more complicated beam-spot shapes, e.g. with a spread in z that varies as a function of t. Note that a large beam spot combined with some of the `MSTJ(22)` options may lead to many particles not being allowed to decay at all.

`MSTP(171) :` (D=0) possibility of variable energies from one event to the next. For further details see section 9.8.

`= 0 :` no; i.e. the energy is fixed at the initialization call.

`= 1 :` yes; i.e. a new energy has to be given for each new event.

Warning: Variable energies cannot be used in conjunction with the internal generation of a virtual photon flux obtained by a `PYINIT` call with `'gamma/`*lepton*`'` argument. The reason is that a variable-energy machinery is now used internally for the γ-hadron or $\gamma\gamma$ subsystem, with some information saved at initialization for the full energy.

`MSTP(172)` : (D=2) options for generation of events with variable energies, applicable when `MSTP(171)=1`.

- `= 1` : an event is generated at the requested energy, i.e. internally a loop is performed over possible event configurations until one is accepted. If the requested c.m. energy of an event is below `PARP(2)` the run is aborted. Cross-section information can not be trusted with this option, since it depends on how you decided to pick the requested energies.
- `= 2` : only one event configuration is tried. If that is accepted, the event is generated in full. If not, no event is generated, and the status code `MSTI(61)=1` is returned. You are then expected to give a new energy, looping until an acceptable event is found. No event is generated if the requested c.m. energy is below `PARP(2)`, instead `MSTI(61)=1` is set to signal the failure. In principle, cross sections should come out correctly with this option.

`MSTP(173)` : (D=0) possibility for you to give in an event weight to compensate for a biased choice of beam spectrum.

- `= 0` : no, i.e. event weight is unity.
- `= 1` : yes; weight to be given for each event in `PARP(173)`, with maximum weight given at initialization in `PARP(174)`.

`MSTP(181)` : (R) PYTHIA version number.
`MSTP(182)` : (R) PYTHIA subversion number.
`MSTP(183)` : (R) last year of change for PYTHIA.
`MSTP(184)` : (R) last month of change for PYTHIA.
`MSTP(185)` : (R) last day of change for PYTHIA.

`PARP(1)` : (D=0.25 GeV) nominal Λ_{QCD} used in running α_s for hard scattering (see `MSTP(3)`).
`PARP(2)` : (D=10. GeV) lowest c.m. energy for the event as a whole that the program will accept to simulate.
`PARP(13)` : (D=1. GeV^2) Q^2_{max} scale, to be set by you for defining maximum scale allowed for photoproduction when using the option `MSTP(13)=2`.
`PARP(14)` : (D=0.01) in the numerical integration of quark and gluon parton distributions inside an electron, the successive halvings of evaluation-point spacing is interrupted when two values agree in relative size, $|\mathrm{new}-\mathrm{old}|/(\mathrm{new}+\mathrm{old})$, to better than `PARP(14)`. There are hardwired lower and upper limits of 2 and 8 halvings, respectively.
`PARP(15)` : (D=0.5 GeV) lower cut-off p_0 used to define minimum transverse momentum in branchings $\gamma \to \mathrm{q}\overline{\mathrm{q}}$ in the anomalous event class of γp interactions, i.e. sets the dividing line between the VMD and GVMD event classes.
`PARP(16)` : (D=1.) the anomalous parton-distribution functions of the photon are taken to have the charm and bottom flavour thresholds at virtuality `PARP(16)`$\times m_{\mathrm{q}}^2$.
`PARP(17)` : (D=1.) rescaling factor used for the Q argument of the anomalous parton distributions of the photon, see `MSTP(15)`.
`PARP(18)` : (D=0.4 GeV) scale k_ρ, such that the cross sections of a GVMD state of scale $k_\perp$ is suppressed by a factor $k_\rho^2/k_\perp^2$ relative to those of a VMD state. Should be of order $m_\rho/2$, with some finetuning to fit data.
`PARP(31)` : (D=1.5) common K factor multiplying the differential cross section for hard parton–parton processes when `MSTP(33)=1` or 2, with the exception of colour annihilation graphs in the latter case.
`PARP(32)` : (D=2.0) special K factor multiplying the differential cross section in hard colour annihilation graphs, including resonance production, when `MSTP(33)=2`.
`PARP(33)` : (D=0.075) this factor is used to multiply the ordinary Q^2 scale in α_s at the hard interaction for `MSTP(33)=3`. The effective K factor thus obtained is in accordance with the results in [Ell86], modulo the danger of doublecounting because of parton-shower corrections to jet rates.

PARP(34) : (D=1.) the Q^2 scale defined by MSTP(32) is multiplied by PARP(34) when it is used as argument for parton distributions and α_s at the hard interaction. It does not affect α_s when MSTP(33)=3, nor does it change the Q^2 argument of parton showers.

PARP(35) : (D=0.20) fix α_s value that is used in the heavy-flavour threshold factor when MSTP(35)=1.

PARP(36) : (D=0. GeV) the width Γ_Q for the heavy flavour studied in processes ISUB = 81 or 82; to be used for the threshold factor when MSTP(35)=2.

PARP(37) : (D=2.) for MSTP(37)=1 this regulates the point at which the reference on-shell quark mass in Higgs and technicolor couplings is assumed defined in PYMRUN calls; specifically the running quark mass is assumed to coincide with the fix one at an energy scale PARP(37) times the fix quark mass, i.e. $m_{\rm running}(\texttt{PARP(37)} \times m_{\rm fix}) = m_{\rm fix}$. See discussion at eq. 90 on ambiguity of PARP(37) choice.

PARP(38) : (D=0.70 GeV^3) the squared wave function at the origin, $|R(0)|^2$, of the J/ψ wave function. Used for processes 86 and 106–108. See ref. [Glo88].

PARP(39) : (D=0.006 GeV^3) the squared derivative of the wave function at the origin, $|R'(0)|^2/m^2$, of the χ_c wave functions. Used for processes 87–89 and 104–105. See ref. [Glo88].

PARP(41) : (D=0.020 GeV) in the process of generating mass for resonances, and optionally to force that mass to be in a given range, only resonances with a total width in excess of PARP(41) are generated according to a Breit–Wigner shape (if allowed by MSTP(42)), while narrower resonances are put on the mass shell.

PARP(42) : (D=2. GeV) minimum mass of resonances assumed to be allowed when evaluating total width of h^0 to Z^0Z^0 or W^+W^- for cases when the h^0 is so light that (at least) one Z/W is forced to be off the mass shell. Also generally used as safety check on minimum mass of resonance. Note that some CKIN values may provide additional constraints.

PARP(43) : (D=0.10) precision parameter used in numerical integration of width for a channel with at least one daughter off the mass shell.

PARP(44) : (D=1000.) the ν parameter of the strongly interacting Z/W model of Dobado, Herrero and Terron [Dob91].

PARP(45) : (D=2054. GeV) the effective techni-ρ mass parameter of the strongly interacting model of Dobado, Herrero and Terron [Dob91]; see MSTP(46)=5. On physical grounds it should not be chosen smaller than about 1 TeV or larger than about the default value.

PARP(46) : (D=123. GeV) the F_π decay constant that appears inversely quadratically in all techni-η partial decay widths [Eic84, App92].

PARP(47) : (D=246. GeV) vacuum expectation value v used in the DHT scenario [Dob91] to define the width of the techni-ρ; this width is inversely proportional v^2.

PARP(48) : (D=50.) the Breit-Wigner factor in the cross section is set to vanish for masses that deviate from the nominal one by more than PARP(48) times the nominal resonance width (i.e. the width evaluated at the nominal mass). Is used in most processes with a single s-channel resonance, but there are some exceptions, notably γ^*/Z^0 and $W^\pm$. The reason for this option is that the conventional Breit-Wigner description is at times not really valid far away from the resonance position, e.g. because of interference with other graphs that should then be included. The wings of the Breit-Wigner can therefore be removed.

PARP(50) : (D=0.054) dimensionless coupling, which enters quadratically in all partial widths of the excited graviton G^* resonance, is $\kappa m_{G^*} = \sqrt{2} x_1 k/\overline{M}_{\rm Pl}$, where $x_1 \approx 3.83$ is the first zero of the J_1 Bessel function and $\overline{M}_{\rm Pl}$ is the modified Planck mass scale [Ran99, Bij01].

PARP(61) - PARP(65) : (C) parameters for initial-state radiation, see section 10.4.

PARP(71) - PARP(72) : (C) parameter for final-state radiation, see section 10.4.

PARP(81) - PARP(90) : parameters for multiple interactions, see section 11.4.

PARP(91) - PARP(100) : parameters for beam remnant treatment, see section 11.4.

PARP(101) : (D=0.50) fraction of diffractive systems in which a quark is assumed kicked out by the pomeron rather than a gluon; applicable for option MSTP(101)=3.

PARP(102) : (D=0.28 GeV) the mass spectrum of diffractive states (in single and double diffractive scattering) is assumed to start PARP(102) above the mass of the particle that is diffractively excited. In this connection, an incoming γ is taken to have the selected VMD meson mass, i.e. m_ρ, m_ω, m_ϕ or $m_{J/\psi}$.

PARP(103) : (D=1.0 GeV) if the mass of a diffractive state is less than PARP(103) above the mass of the particle that is diffractively excited, the state is forced to decay isotropically into a two-body channel. In this connection, an incoming γ is taken to have the selected VMD meson mass, i.e. m_ρ, m_ω, m_ϕ or $m_{J/\psi}$. If the mass is higher than this threshold, the standard string fragmentation machinery is used. The forced two-body decay is always carried out, also when MSTP(111)=0.

PARP(104) : (D=0.8 GeV) minimum energy above threshold for which hadron–hadron total, elastic and diffractive cross sections are defined. Below this energy, an alternative description in terms of specific few-body channels would have been required, and this is not modelled in PYTHIA.

PARP(110) : (D=1.) a rescaling factor for resonance widths, applied when MSTP(110 is switched on.

PARP(111) : (D=2. GeV) used to define the minimum invariant mass of the remnant hadronic system (i.e. when interacting partons have been taken away), together with original hadron masses and extra parton masses. For a hadron or resolved photon beam, this also implies a further contraint that the x of an interacting parton be below $1 - 2 \times$ PARP(111)$/E_{\rm cm}$.

PARP(115) : (D=1.5 fm) (C) the average fragmentation time of a string, giving the exponential suppression that a reconnection cannot occur if strings decayed before crossing. Is implicitly fixed by the string constant and the fragmentation function parameters, and so a significant change is not recommended.

PARP(116) : (D=0.5 fm) (C) width of the type I string, giving the radius of the Gaussian distribution in x and y separately.

PARP(117) : (D=0.6) (C) $k_{\rm I}$, the main free parameter in the reconnection probability for scenario I; the probability is given by PARP(117) times the overlap volume, up to saturation effects.

PARP(118), PARP(119) : (D=2.5,2.0) (C) f_r and f_t, respectively, used in the Monte Carlo sampling of the phase space volume in scenario I. There is no real reason to change these numbers.

PARP(120) : (D=1.0) (D) (C) fraction of events in the GH, intermediate and instantaneous scenarios where a reconnection is allowed to occur. For the GH one a further suppression of the reconnection rate occurs from the requirement of reduced string length in a reconnection.

PARP(121) : (D=1.) the maxima obtained at initial maximization are multiplied by this factor if MSTP(121)=1; typically PARP(121) would be given as the product of the violation factors observed (i.e. the ratio of final maximum value to initial maximum value) for the given process(es).

PARP(122) : (D=0.4) fraction of total probability that is shared democratically between the COEF coefficients open for the given variable, with the remaining fraction distributed according to the optimization results of PYMAXI.

PARP(131) : parameter for pile-up events, see section 11.4.

PARP(137) : (D=200.000 GeV) M_V, vector mass parameter for technivector decays to transverse gauge bosons and technipions.

PARP(138) : (D=200.000 GeV) M_A, axial mass parameter for technivector decays to transverse gauge bosons and technipions.

PARP(139) : (D=0.33300) $\sin\chi'$, where χ' is the mixing angle between the $\pi'^0_{\rm tc}$ interaction and

mass eigenstates.

PARP(140) : (D=0.05000) isospin violating $\rho^0_{tc}/\omega^0_{tc}$ mixing amplitude.

PARP(141) : (D= 0.33333) $\sin\chi$, where χ is the mixing angle between technipion interaction and mass eigenstates.

PARP(142) : (D=82.0000 GeV) F_T, the technipion decay constant.

PARP(143) : (D= 1.3333) Q_U, charge of up-type technifermion; the down-type technifermion has a charge $Q_D = Q_U - 1$.

PARP(144) : (D= 4.0000) N_{TC}, number of technicolors; fixes the relative values of $g_{\rm em}$ and $g_{\rm etc}$.

PARP(145) : (D= 1.0000) $C_{\rm c}$, coefficient of the technipion decays to charm; appears squared in the decay rate.

PARP(146) : (D= 1.0000) $C_{\rm b}$, coefficient of the technipion decays to bottom; appears squared in the decay rate.

PARP(147) : (D= 0.0182) $C_{\rm t}$, coefficient of the technipion decays to top, estimated to be $m_{\rm b}/m_{\rm t}$; appears squared in the decay rate.

PARP(148) : (D= 1.0000) C_τ, coefficient of the technipion decays to τ; appears squared in the decay rate.

PARP(149) : (D=0.00000) C_π, coefficient of technipion decays to gg.

PARP(150) : (D=1.33333) $C_{\pi'}$, coefficient of $\pi'^0_{\rm tc}$ decays to gg.

PARP(151) - PARP(154) : (D=4*0.) (C) regulate the assumed beam-spot size. For MSTP(151)=1 the x, y, z and t coordinates of the primary vertex of each event are selected according to four independent Gaussians. The widths of these Gaussians are given by the four parameters, where the first three are in units of mm and the fourth in mm/c.

PARP(155) : (D=0.3651480) parameter in the scenario with coloured technihadrons described in subsection 8.6.7 and switched on with MSTP(5)=5. The sign of PARP(155) is used to fix one of two models. For PARP(155)>0, the coupling of the V_8 to light quarks is suppressed by PARP(155)^2 whereas the coupling to heavy (b and t) quarks is enhanced by 1/PARP(155)^2. For PARP(155)<0, the couplings to quarks is universal, and given by 1/PARP(155)^2.

PARP(156) : (D=200 GeV) mass parameter in the scenario with coloured technihadrons described in subsection 8.6.7 and switched on with MSTP(5)=5. It sets the scale for the decay of technirhos into technipions, thereby affecting the width of the resonances.

PARP(161) - PARP(164) : (D=2.20, 23.6, 18.4, 11.5) couplings $f_V^2/4\pi$ of the photon to the ρ^0, ω, ϕ and J/ψ vector mesons.

PARP(165) : (D=0.5) a simple multiplicative factor applied to the cross section for the transverse resolved photons to take into account the effects of longitudinal resolved photons, see MSTP(17). No preferred value, but typically one could use PARP(165)=1 as main contrast to the no-effect =0, with the default arbitrarily chosen in the middle.

PARP(167), PARP(168) : (D=2*0) the longitudinal energy fraction y of an incoming photon, side 1 or 2, used in the R expression given for MSTP(17) to evaluate $f_L(y,Q^2)/f_T(y,Q^2)$. Need not be supplied when a photon spectrum is generated inside a lepton beam, but only when a photon is directly given as argument in the PYINIT call.

PARP(171) : to be set, event-by-event, when variable energies are allowed, i.e. when MSTP(171)=1. If PYINIT is called with FRAME='CMS' (='FIXT'), PARP(171) multiplies the c.m. energy (beam energy) used at initialization. For the options '3MOM', '4MOM' and '5MOM', PARP(171) is dummy, since there the momenta are set in the P array. It is also dummy for the 'USER' option, where the choice of variable energies is beyond the control of PYTHIA.

PARP(173) : event weight to be given by you when MSTP(173)=1.

PARP(174) : (D=1.) maximum event weight that will be encountered in PARP(173) during the course of a run with MSTP(173)=1; to be used to optimize the efficiency of the

event generation. It is always allowed to use a larger bound than the true one, but with a corresponding loss in efficiency.

PARP(181) - PARP(189) : (D = 0.1, 0.01, 0.01, 0.01, 0.1, 0.01, 0.01, 0.01, 0.3) Yukawa couplings of leptons to H^{++}, assumed same for H_L^{++} and H_R^{++}. Is a symmetric 3×3 array, where PARP(177+3*i+j) gives the coupling to a lepton pair with generation indices i and j. Thus the default matrix is dominated by the diagonal elements and especially by the $\tau\tau$ one.

PARP(190) : (D=0.64) $g_L = e/\sin\theta_W$.

PARP(191) : (D=0.64) g_R, assumed same as g_L.

PARP(192) : (D=5 GeV) v_L vacuum expectation value of the left-triplet. The corresponding v_R is assumed given by $v_R = \sqrt{2}M_{W_R}/g_R$ and is not stored explicitly.

9.4 Further Couplings

In this section we collect information on the two routines for running α_s and α_{em}, and on other couplings of standard and non-standard particles found in the PYDAT1 common block. Although originally begun for applications within the traditional particle sector, this section has rapidly expanded towards the non-standard aspects, and is thus more of interest for applications to specific processes. It could therefore equally well have been put somewhere else in this manual. Several other couplings indeed appear in the PARP array in the PYPARS common block, see section 9.3, and the choice between the two has largely been dictated by availability of space.

```
ALEM = PYALEM(Q2)
```

Purpose: to calculate the running electromagnetic coupling constant α_{em}. Expressions used are described in ref. [Kle89]. See MSTU(101), PARU(101), PARU(103) and PARU(104).

Q2 : the momentum transfer scale Q^2 at which to evaluate α_{em}.

```
ALPS = PYALPS(Q2)
```

Purpose: to calculate the running strong coupling constant α_s, e.g. in matrix elements and resonance decay widths. (The function is not used in parton showers, however, where formulae rather are written in terms of the relevant Λ values.) The first- and second-order expressions are given by eqs. (27) and (32). See MSTU(111) - MSTU(118) and PARU(111) - PARU(118) for options.

Q2 : the momentum transfer scale Q^2 at which to evaluate α_s.

```
PM = PYMRUN(KF,Q2)
```

Purpose: to give running masses of d, u, s, c, b and t quarks according to eq. 90. For all other particles, the PYMASS function is called by PYMRUN to give the normal mass. Such running masses appear e.g. in couplings of fermions to Higgs and technipion states.

KF : flavour code.

Q2 : the momentum transfer scale Q^2 at which to evaluate α_s.

Note: The nominal values, valid at a reference scale
$Q^2_{ref} = \max((\mathtt{PARP}(37)m_{nominal})^2, 4\Lambda^2)$,
are stored in PARF(91)-PARF(96).

```
COMMON/PYDAT1/MSTU(200),PARU(200),MSTJ(200),PARJ(200)
```

Purpose: to give access to a number of status codes and parameters which regulate the performance of the program as a whole. Here only those related to couplings are described; the main description is found in section 14.3.

`MSTU(101)` : (D=1) procedure for $\alpha_{\rm em}$ evaluation in the `PYALEM` function.
- `= 0` : $\alpha_{\rm em}$ is taken fixed at the value `PARU(101)`.
- `= 1` : $\alpha_{\rm em}$ is running with the Q^2 scale, taking into account corrections from fermion loops (e, μ, τ, d, u, s, c, b).
- `= 2` : $\alpha_{\rm em}$ is fixed, but with separate values at low and high Q^2. For Q^2 below (above) `PARU(104)` the value `PARU(101)` (`PARU(103)`) is used. The former value is then intended for real photon emission, the latter for electroweak physics, e.g. of the W/Z gauge bosons.

`MSTU(111)` : (I, D=1) order of $\alpha_{\rm s}$ evaluation in the `PYALPS` function. Is overwritten in `PYEEVT`, `PYONIA` or `PYINIT` calls with the value desired for the process under study.
- `= 0` : $\alpha_{\rm s}$ is fixed at the value `PARU(111)`. As extra safety, Λ =`PARU(117)` is set in `PYALPS` so that the first-order running $\alpha_{\rm s}$ agrees with the desired fixed $\alpha_{\rm s}$ for the Q^2 value used.
- `= 1` : first-order running $\alpha_{\rm s}$ is used.
- `= 2` : second-order running $\alpha_{\rm s}$ is used.

`MSTU(112)` : (D=5) the nominal number of flavours assumed in the $\alpha_{\rm s}$ expression, with respect to which Λ is defined.

`MSTU(113)` : (D=3) minimum number of flavours that may be assumed in $\alpha_{\rm s}$ expression, see `MSTU(112)`.

`MSTU(114)` : (D=5) maximum number of flavours that may be assumed in $\alpha_{\rm s}$ expression, see `MSTU(112)`.

`MSTU(115)` : (D=0) treatment of $\alpha_{\rm s}$ singularity for $Q^2 \to 0$ in `PYALPS` calls. (Relevant e.g. for QCD $2 \to 2$ matrix elements in the $p_\perp \to 0$ limit, but not for showers, where `PYALPS` is not called.)
- `= 0` : allow it to diverge like $1/\ln(Q^2/\Lambda^2)$.
- `= 1` : soften the divergence to $1/\ln(1+Q^2/\Lambda^2)$.
- `= 2` : freeze Q^2 evolution below `PARU(114)`, i.e. the effective argument is $\max(Q^2,$`PARU(114)`$)$.

`MSTU(118)` : (I) number of flavours n_f found and used in latest `PYALPS` call.

`PARU(101)` : (D=0.00729735=1/137.04) $\alpha_{\rm em}$, the electromagnetic fine structure constant at vanishing momentum transfer.

`PARU(102)` : (D=0.232) $\sin^2\theta_W$, the weak mixing angle of the standard electroweak model.

`PARU(103)` : (D=0.007764=1/128.8) typical $\alpha_{\rm em}$ in electroweak processes; used for $Q^2 >$`PARU(104)` in the option `MSTU(101)=2` of `PYALEM`. Although it can technically be used also at rather small Q^2, this $\alpha_{\rm em}$ value is mainly intended for high Q^2, primarily Z^0 and $W^\pm$ physics.

`PARU(104)` : (D=1 GeV^2) dividing line between 'low' and 'high' Q^2 values in the option `MSTU(101)=2` of `PYALEM`.

`PARU(105)` : (D=1.16639E-5 GeV^{-2}) $G_{\rm F}$, the Fermi constant of weak interactions.

`PARU(108)` : (I) the $\alpha_{\rm em}$ value obtained in the latest call to the `PYALEM` function.

`PARU(111)` : (D=0.20) fix $\alpha_{\rm s}$ value assumed in `PYALPS` when `MSTU(111)=0` (and also in parton showers when $\alpha_{\rm s}$ is assumed fix there).

`PARU(112)` : (I, D=0.25 GeV) Λ used in running $\alpha_{\rm s}$ expression in `PYALPS`. Like `MSTU(111)`, this value is overwritten by the calling physics routines, and is therefore purely nominal.

`PARU(113)` : (D=1.) the flavour thresholds, for the effective number of flavours n_f to use in the $\alpha_{\rm s}$ expression, are assumed to sit at $Q^2 =$`PARU(113)`$\times m_{\rm q}^2$, where $m_{\rm q}$ is the quark mass. May be overwritten from the calling physics routine.

PARU(114) : (D=4 GeV^2) Q^2 value below which the α_s value is assumed constant for MSTU(115)=2.

PARU(115) : (D=10.) maximum α_s value that PYALPS will ever return; is used as a last resort to avoid singularities.

PARU(117) : (I) Λ value (associated with MSTU(118) effective flavours) obtained in latest PYALPS call.

PARU(118) : (I) α_s value obtained in latest PYALPS call.

PARU(121) - PARU(130) : couplings of a new Z'^0; for fermion default values are given by the Standard Model Z^0 values, assuming $\sin^2\theta_W = 0.23$. Since a generation dependence is now allowed for the Z'^0 couplings to fermions, the variables PARU(121) - PARU(128) only refer to the first generation, with the second generation in PARJ(180) - PARJ(187) and the third in PARJ(188) - PARJ(195) following exactly the same pattern. Note that e.g. the Z'^0 width contains squared couplings, and thus depends quadratically on the values below.

PARU(121), PARU(122) : (D=-0.693,-1.) vector and axial couplings of down type quarks to Z'^0.

PARU(123), PARU(124) : (D=0.387,1.) vector and axial couplings of up type quarks to Z'^0.

PARU(125), PARU(126) : (D=-0.08,-1.) vector and axial couplings of leptons to Z'^0.

PARU(127), PARU(128) : (D=1.,1.) vector and axial couplings of neutrinos to Z'^0.

PARU(129) : (D=1.) the coupling $Z'^0 \to W^+W^-$ is taken to be PARU(129)$\times$(the Standard Model $Z^0 \to W^+W^-$ coupling)$\times(m_W/m_{Z'})^2$. This gives a $Z'^0 \to W^+W^-$ partial width that increases proportionately to the Z'^0 mass.

PARU(130) : (D=0.) in the decay chain $Z'^0 \to W^+W^- \to 4$ fermions, the angular distribution in the W decays is supposed to be a mixture, with fraction 1-PARU(130) corresponding to the same angular distribution between the four final fermions as in $Z^0 \to W^+W^-$ (mixture of transverse and longitudinal W's), and fraction PARU(130) corresponding to $h^0 \to W^+W^-$ the same way (longitudinal W's).

PARU(131) - PARU(136) : couplings of a new $W'^\pm$; for fermions default values are given by the Standard Model $W^\pm$ values (i.e. $V-A$). Note that e.g. the $W'^\pm$ width contains squared couplings, and thus depends quadratically on the values below.

PARU(131), PARU(132) : (D=1.,-1.) vector and axial couplings of a quark–antiquark pair to $W'^\pm$; is further multiplied by the ordinary CKM factors.

PARU(133), PARU(134) : (D=1.,-1.) vector and axial couplings of a lepton-neutrino pair to $W'^\pm$.

PARU(135) : (D=1.) the coupling $W'^\pm \to Z^0W^\pm$ is taken to be PARU(135)$\times$(the Standard Model $W^\pm \to Z^0W^\pm$ coupling)$\times(m_W/m_{W'})^2$. This gives a $W'^\pm \to Z^0W^\pm$ partial width that increases proportionately to the W' mass.

PARU(136) : (D=0.) in the decay chain $W'^\pm \to Z^0W^\pm \to 4$ fermions, the angular distribution in the W/Z decays is supposed to be a mixture, with fraction 1-PARU(136) corresponding to the same angular distribution between the four final fermions as in $W^\pm \to Z^0W^\pm$ (mixture of transverse and longitudinal W/Z's), and fraction PARU(136) corresponding to $H^\pm \to Z^0W^\pm$ the same way (longitudinal W/Z's).

PARU(141) : (D=5.) $\tan\beta$ parameter of a two Higgs doublet scenario, i.e. the ratio of vacuum expectation values. This affects mass relations and couplings in the Higgs sector. If the Supersymmetry simulation is switched on, IMSS(5) nonvanishing, PARU(141) will be overwritten by RMSS(5) at initialization, so it is the latter varaible that should be set.

PARU(142) : (D=1.) the $Z^0 \to H^+H^-$ coupling is taken to be PARU(142)$\times$(the MSSM $Z^0 \to H^+H^-$ coupling).

PARU(143) : (D=1.) the $Z'^0 \to H^+H^-$ coupling is taken to be PARU(143)$\times$(the MSSM $Z^0 \to$

H^+H^- coupling).

PARU(145) : (D=1.) quadratically multiplicative factor in the $Z'^0 \to Z^0h^0$ partial width in left–right-symmetric models, expected to be unity (see [Coc91]).

PARU(146) : (D=1.) $\sin(2\alpha)$ parameter, enters quadratically as multiplicative factor in the $W'^{\pm} \to W^{\pm}h^0$ partial width in left–right-symmetric models (see [Coc91]).

PARU(151) : (D=1.) multiplicative factor in the $L_Q \to q\ell$ squared Yukawa coupling, and thereby in the L_Q partial width and the $q\ell \to L_Q$ and other cross sections. Specifically, $\lambda^2/(4\pi) =$PARU(151)$\times\alpha_{em}$, i.e. it corresponds to the *k* factor of [Hew88].

PARU(153) : (D=0.) anomalous magnetic moment of the $W^{\pm}$ in process 20; $\eta = \kappa - 1$, where $\eta = 0$ ($\kappa = 1$) is the Standard Model value.

PARU(155) : (D=1000. GeV) compositeness scale Λ.

PARU(156) : (D=1.) sign of interference term between standard cross section and composite term (η parameter); should be ± 1.

PARU(157) - PARU(159) : (D=3*1.) strength of **SU(2)**, **U(1)** and **SU(3)** couplings, respectively, in an excited fermion scenario; cf. f, f' and f_s of [Bau90].

PARU(161) - PARU(168) : (D=5*1.,3*0.) multiplicative factors that can be used to modify the default couplings of the h^0 particle in PYTHIA. Note that the factors enter quadratically in the partial widths. The default values correspond to the couplings given in the minimal one-Higgs-doublet Standard Model, and are therefore not realistic in a two-Higgs-doublet scenario. The default values should be changed appropriately by you. Also the last two default values should be changed; for these the expressions of the minimal supersymmetric Standard Model (MSSM) are given to show parameter normalization. Alternatively, the SUSY machinery can generate all the couplings for IMSS(1), see MSTP(4).

- PARU(161) : h^0 coupling to down type quarks.
- PARU(162) : h^0 coupling to up type quarks.
- PARU(163) : h^0 coupling to leptons.
- PARU(164) : h^0 coupling to Z^0.
- PARU(165) : h^0 coupling to $W^{\pm}$.
- PARU(168) : h^0 coupling to $H^{\pm}$ in $\gamma\gamma \to h^0$ loops, in MSSM $\sin(\beta-\alpha)+\cos(2\beta)\sin(\beta+\alpha)/(2\cos^2\theta_W)$.

PARU(171) - PARU(178) : (D=7*1.,0.) multiplicative factors that can be used to modify the default couplings of the H^0 particle in PYTHIA. Note that the factors enter quadratically in partial widths. The default values for PARU(171) - PARU(175) correspond to the couplings given to h^0 in the minimal one-Higgs-doublet Standard Model, and are therefore not realistic in a two-Higgs-doublet scenario. The default values should be changed appropriately by you. Also the last two default values should be changed; for these the expressions of the minimal supersymmetric Standard Model (MSSM) are given to show parameter normalization. Alternatively, the SUSY machinery can generate all the couplings for IMSS(1), see MSTP(4).

- PARU(171) : H^0 coupling to down type quarks.
- PARU(172) : H^0 coupling to up type quarks.
- PARU(173) : H^0 coupling to leptons.
- PARU(174) : H^0 coupling to Z^0.
- PARU(175) : H^0 coupling to $W^{\pm}$.
- PARU(176) : H^0 coupling to h^0h^0, in MSSM $\cos(2\alpha)\cos(\beta+\alpha)-2\sin(2\alpha)\sin(\beta+\alpha)$.
- PARU(177) : H^0 coupling to A^0A^0, in MSSM $\cos(2\beta)\cos(\beta+\alpha)$.
- PARU(178) : H^0 coupling to $H^{\pm}$ in $\gamma\gamma \to H^0$ loops, in MSSM $\cos(\beta-\alpha)-\cos(2\beta)\cos(\beta+\alpha)/(2\cos^2\theta_W)$.

PARU(181) - PARU(190) : (D=3*1.,2*0.,2*1.,3*0.) multiplicative factors that can be used to modify the default couplings of the A^0 particle in PYTHIA. Note that the factors enter quadratically in partial widths. The default values for PARU(181) -

PARU(183) correspond to the couplings given to h^0 in the minimal one-Higgs-doublet Standard Model, and are therefore not realistic in a two-Higgs-doublet scenario. The default values should be changed appropriately by you. PARU(184) and PARU(185) should be vanishing at the tree level, in the absence of CP violating phases in the Higgs sector, and are so set; normalization of these couplings agrees with what is used for h^0 and H^0. Also the other default values should be changed; for these the expressions of the Minimal Supersymmetric Standard Model (MSSM) are given to show parameter normalization. Alternatively, the SUSY machinery can generate all the couplings for IMSS(1), see MSTP(4).

PARU(181) : A^0 coupling to down type quarks.
PARU(182) : A^0 coupling to up type quarks.
PARU(183) : A^0 coupling to leptons.
PARU(184) : A^0 coupling to Z^0.
PARU(185) : A^0 coupling to $W^{\pm}$.
PARU(186) : A^0 coupling to Z^0h^0 (or Z^* to A^0h^0), in MSSM $\cos(\beta-\alpha)$.
PARU(187) : A^0 coupling to Z^0H^0 (or Z^* to A^0H^0), in MSSM $\sin(\beta-\alpha)$.
PARU(188) : As PARU(186), but coupling to Z'^0 rather than Z^0.
PARU(189) : As PARU(187), but coupling to Z'^0 rather than Z^0.
PARU(190) : A^0 coupling to $H^{\pm}$ in $\gamma\gamma \to A^0$ loops, 0 in MSSM.

PARU(191) - PARU(195) : (D=4*0.,1.) multiplicative factors that can be used to modify the couplings of the $H^{\pm}$ particle in PYTHIA. Currently only PARU(195) is in use. See above for related comments.

PARU(195) : $H^{\pm}$ coupling to $W^{\pm}h^0$ (or $W^{*\pm}$ to $h^{\pm}h^0$), in MSSM $\cos(\beta-\alpha)$.

PARJ(180) - PARJ(187) : couplings of the second generation fermions to the Z'^0, following the same pattern and with the same default values as the first one in PARU(121) - PARU(128).

PARJ(188) - PARJ(195) : couplings of the third generation fermions to the Z'^0, following the same pattern and with the same default values as the first one in PARU(121) - PARU(128).

9.5 Supersymmetry Common Blocks and Routines

The parameters available to the SUSY user are stored in the common block PYMSSM. In general, options are set by the IMSS array, while real valued parameters are set by RMSS. The entries IMSS(0) and RMSS(0) are not used, but are available for compatibility with the C programming language.

```
COMMON/PYMSSM/IMSS(0:99),RMSS(0:99)
```

Purpose: to give access to parameters that allow the simulation of the MSSM.

IMSS(1) : (D=0) level of MSSM simulation.

= 0 : No MSSM simulation.

= 1 : A general MSSM simulation. The parameters of the model are set by the array RMSS.

= 2 : An approximate SUGRA simulation using the analytic formulae of [Dre95] to reduce the number of free parameters. In this case, only five input parameters are used. RMSS(1) is the common gaugino mass $m_{1/2}$, RMSS(8) is the common scalar mass m_0, RMSS(4) fixes the sign of the higgsino mass μ, RMSS(16) is the common trilinear coupling A, and RMSS(5) is $\tan\beta = v_2/v_1$.

IMSS(2) : (D=0) treatment of **U(1)**, **SU(2)**, and **SU(3)** gaugino mass parameters.

= 0 : The gaugino parameters M_1, M_2 and M_3 are set by RMSS(1), RMSS(2), and

RMSS(3), i.e. there is no forced relation between them.

= 1 : The gaugino parameters are fixed by the relation $(3/5)M_1/\alpha_1 = M_2/\alpha_2 = M_3/\alpha_3 = X$ and the parameter RMSS(1). If IMSS(1)=2, then RMSS(1) is treated as the common gaugino mass $m_{1/2}$ and RMSS(20) is the GUT scale coupling constant α_{GUT}, so that $X = m_{1/2}/\alpha_{GUT}$.

= 2 : M_1 is set by RMSS(1), M_2 by RMSS(2) and $M_3 = M_2\alpha_3/\alpha_2$. In such a scenario, the **U(1)** gaugino mass behaves anomalously.

IMSS(3) : (D=0) treatment of the gluino mass parameter.

= 0 : The gluino mass parameter M_3 is used to calculate the gluino pole mass with the formulae of [Kol96]. The effects of squark loops can significantly shift the mass.

= 1 : M_3 is the gluino pole mass. The effects of squark loops are assumed to have been included in this value.

IMSS(4) : (D=1) treatment of the Higgs sector.

= 0 : The Higgs sector is determined by the approximate formulae of [Car95] and the pseudoscalar mass $M_{\rm A}$ set by RMSS(19).

= 1 : The Higgs sector is determined by the exact formulae of [Car95] and the pseudoscalar mass $M_{\rm A}$ set by RMSS(19). The pole mass for $M_{\rm A}$ is not the same as the input parameter.

= 2 : The Higgs sector is fixed by the mixing angle α set by RMSS(18) and the mass values PMAS(I,1), where I=25,35,36, and 37.

IMSS(5) : (D=0) allows you to set the $\tilde{\rm t}$, $\tilde{\rm b}$ and $\tilde{\tau}$ masses and mixing by hand.

= 0 : no, the program calculates itself.

= 1 : yes, calculate from given input. The parameters RMSS(26) - RMSS(28) specify the mixing angle (in radians) for the sbottom, stop, and stau. The parameters RMSS(10) - RMSS(14) specify the two stop masses, the one sbottom mass (the other being fixed by the other parameters) and the two stau masses. Note that the masses RMSS(10), RMSS(11) and RMSS(13) correspond to the left-left entries of the diagonalized matrices, while RMSS(12) and RMSS(14) correspond to the right-right entries. Note that these entries need not be ordered in mass.

IMSS(7) : (D=0) treatment of the scalar masses in an extension of SUGRA models. The presence of additional **U(1)** symmetries at high energy scales can modify the boundary conditions for the scalar masses at the unification scale.

= 0 : No additional D–terms are included. In SUGRA models, all scalars have the mass m_0 at the unification scale.

= 1 : RMSS(23) - RMSS(25) are the values of D_X,D_Y and D_S at the unification scale in the model of [Mar94]. The boundary conditions for the scalar masses are shifted based on their quantum numbers under the additional **U(1)** symmetries.

IMSS(8) : (D=1) treatment of the $\tilde{\tau}$ mass eigenstates.

= 0 : The $\tilde{\tau}$ mass eigenstates are calculated using the parameters RMSS(13,14,17).

= 1 : The $\tilde{\tau}$ mass eigenstates are identical to the interaction eigenstates, so they are treated identically to $\tilde{\rm e}$ and $\tilde{\mu}$.

IMSS(9) : (D=0) treatment of the right handed squark mass eigenstates for the first two generations.

= 0 : The $\tilde{\rm q}_R$ masses are fixed by RMSS(9). $\tilde{\rm d}_R$ and $\tilde{\rm u}_R$ are identical except for Electroweak D–term contributions.

= 1 : The masses of $\tilde{\rm d}_R$ and $\tilde{\rm u}_R$ are fixed by RMSS(9) and RMSS(22) respectively.

IMSS(10) : (D=0) allowed decays for $\tilde{\chi}_2$.

= 0 : The second lightest neutralino $\tilde{\chi}_2$ decays with a branching ratio calculated from the MSSM parameters.

= 1 : $\tilde{\chi}_2$ is forced to decay only to $\tilde{\chi}_1\gamma$, regardless of the actual branching ratio. This can be used for detailed studies of this particular final state.

IMSS(11) : (D=0) choice of the lightest superpartner (LSP).

= 0 : $\tilde{\chi}_1$ is the LSP.

= 1 : $\tilde{\chi}_1$ is the next to lightest superparter (NLSP) and the gravitino is the LSP. The $\tilde{\chi}_1$ decay length is calculated from the gravitino mass set by RMSS(21) and the $\tilde{\chi}_1$ mass and mixing.

IMSS(51) : (D=0) Lepton number violation on/off (LLE type couplings).

= 0 : All LLE couplings off. LLE decay channels off.

= 1 : All LLE couplings set to common value given by $10^{\text{-RMSS(51)}}$.

= 2 : LLE couplings set to generation-hierarchical 'natural' values with common normalization RMSS(51) (see section 8.7.2).

= 3 : All LLE couplings set to zero, but LLE decay channels not switched off. Non-zero couplings should be entered individually into the array RVLAM(I,J,K).

IMSS(52) : (D=0) Lepton number violation on/off (LQD type couplings).

= 0 : All LQD couplings off. LQD decay channels off.

= 1 : All LQD couplings set to common value given by $10^{\text{-RMSS(52)}}$.

= 2 : LQD couplings set to generation-hierarchical 'natural' values with common normalization RMSS(52) (see section 8.7.2).

= 3 : All LQD couplings set to zero, but LQD decay channels not switched off. Non-zero couplings should be entered individually into the array RVLAMP(I,J,K).

IMSS(53) : Reserved for baryon number violation.

RMSS(1) : (D=80. GeV) If IMSS(1)=1 M_1, then **U(1)** gaugino mass. If IMSS(1)=2, then the common gaugino mass $m_{1/2}$.

RMSS(2) : (D=160. GeV) M_2, the **SU(2)** gaugino mass.

RMSS(3) : (D=500. GeV) M_3, the **SU(3)** (gluino) mass parameter.

RMSS(4) : (D=800. GeV) μ, the higgsino mass parameter. If IMSS(1)=2, only the sign of μ is used.

RMSS(5) : (D=2.) $\tan\beta$, the ratio of Higgs expectation values.

RMSS(6) : (D=250. GeV) Left slepton mass $M_{\tilde{\ell}_L}$. The sneutrino mass is fixed by a sum rule.

RMSS(7) : (D=200. GeV) Right slepton mass $M_{\tilde{\ell}_R}$.

RMSS(8) : (D=800. GeV) Left squark mass $M_{\tilde{q}_L}$. If IMSS(1)=2, the common scalar mass m_0.

RMSS(9) : (D=700. GeV) Right squark mass $M_{\tilde{q}_R}$. $M_{\tilde{d}_R}$ when IMSS(9)=1.

RMSS(10) : (D=800. GeV) Left squark mass for the third generation $M_{\tilde{q}_L}$. When IMSS(5)=1, it is instead the $\tilde{t}_2$ mass, and $M_{\tilde{q}_L}$ is a derived quantity.

RMSS(11) : (D=700. GeV) Right sbottom mass $M_{\tilde{b}_R}$. When IMSS(5)=1, it is instead the $\tilde{b}_1$ mass.

RMSS(12) : (D=500. GeV) Right stop mass $M_{\tilde{t}_R}$ If negative, then it is assumed that $M^2_{\tilde{t}_R} < 0$. When IMSS(5)=1, it is instead the $\tilde{t}_1$ mass.

RMSS(13) : (D=250. GeV) Left stau mass $M_{\tilde{\tau}_L}$.

RMSS(14) : (D=200. GeV) Right stau mass $M_{\tilde{\tau}_R}$.

RMSS(15) : (D=800. GeV) Bottom trilinear coupling A_b. When IMSS(5)=1, it is a derived quantity.

RMSS(16) : (D=400. GeV) Top trilinear coupling A_t. If IMSS(1)=2, the common trilinear coupling A. When IMSS(5)=1, it is a derived quantity.

RMSS(17) : (D=0.) Tau trilinear coupling A_τ. When IMSS(5)=1, it is a derived quantity.

RMSS(18) : (D=0.1) Higgs mixing angle α. This is only used when all of the Higgs parameters are set by you, i.e IMSS(4)=2.

RMSS(19) : (D=850. GeV) Pseudoscalar Higgs mass parameter M_A.

RMSS(20) : (D=0.041) GUT scale coupling constant α_{GUT}.

RMSS(21) : (D=1.0 eV) The gravitino mass. Note nonconventional choice of units for this

particular mass.

`RMSS(22)` : (D=800. GeV) $\tilde{u}_R$ mass when `IMSS(9)=1`.

`RMSS(23)` : (D=10^4 GeV^2) D_X contribution to scalar masses when `IMSS(7)=1`.

`RMSS(24)` : (D=10^4 GeV^2) D_Y contribution to scalar masses when `IMSS(7)=1`.

`RMSS(25)` : (D=10^4 GeV^2) D_S contribution to scalar masses when `IMSS(7)=1`.

`RMSS(26)` : (D=0.0 radians) when `IMSS(5)=1` it is the sbottom mixing angle.

`RMSS(27)` : (D=0.0 radians) when `IMSS(5)=1` it is the stop mixing angle.

`RMSS(28)` : (D=0.0 radians) when `IMSS(5)=1` it is the stau mixing angle.

`RMSS(29)` : (D=2.4×10^{18} GeV) The Planck mass, used for calculating decays to light gravitinos.

`RMSS(30)` - `RMSS(33)` : (D=0.0,0.0,0.0,0.0) complex phases for the mass parameters in `RMSS(1)` - `RMSS(4)`, where the latter represent the moduli of the mass parameters for the case of nonvanishing phases.

`RMSS(40)`, `RMSS(41)` : used for temporary storage of the corrections $\Delta m_{\rm t}$ and $\Delta m_{\rm b}$, respectively, in the calculation of Higgs properties.

`RMSS(51)` : (D=0.0) when `IMSS(51)=1` it is the negative logarithm of the common value for all lepton number violating λ couplings (LLE). When `IMSS(51)=2` it is the constant of proportionality for generation-hierarchical λ couplings. See section 8.7.2.

`RMSS(52)` : (D=0.0) when `IMSS(52)=1` it is the negative logarithm of the common value for all lepton number violating λ couplings (LQD). When `IMSS(52)=2` it is the constant of proportionality for generation-hierarchical λ' couplings. See section 8.7.2.

`RMSS(53)` : (D=0.0) Reserved for baryon number violation.

```
 COMMON/PYSSMT/ZMIX(4,4),UMIX(2,2),VMIX(2,2),SMZ(4),SMW(2),
&SFMIX(16,4),ZMIXI(4,4),UMIXI(2,2),VMIXI(2,2)
```

Purpose: to provide information on the neutralino, chargino, and sfermion mixing parameters. The variables should not be changed by you.

`ZMIX(4,4)` : the real part of the neutralino mixing matrix in the Bino–neutral Wino–Up higgsino–Down higgsino basis.

`UMIX(2,2)` : the real part of the chargino mixing matrix in the charged Wino–charged higgsino basis.

`VMIX(2,2)` : the real part of the charged conjugate chargino mixing matrix in the wino–charged higgsino basis.

`SMZ(4)` : the signed masses of the neutralinos.

`SMW(2)` : the signed masses of the charginos.

`SFMIX(16,4)` : the sfermion mixing matrices **T** in the L–R basis, identified by the corresponding fermion, i.e. `SFMIX(6,I)` is the stop mixing matrix. The four entries for each sfermion are T_{11}, T_{12}, T_{21}, and T_{22}.

`ZMIXI(4,4)` : the imaginary part of the neutralino mixing matrix in the Bino–neutral Wino–Up higgsino–Down higgsino basis.

`UMIXI(2,2)` : the imaginary part of the chargino mixing matrix in the charged Wino–charged higgsino basis.

`VMIXI(2,2)` : the imaginary part of the charged conjugate chargino mixing matrix in the wino–charged higgsino basis.

```
 COMMON/PYMSRV/RVLAM(3,3,3), RVLAMP(3,3,3), RVLAMB(3,3,3)
```

Purpose: to provide information on lepton and baryon number violating couplings.

`RVLAM(3,3,3)` : the lepton number violating λ_{ijk} couplings. See `IMSS(51)`, `RMSS(51)`.

`RVLAMP(3,3,3)` : the lepton number violating λ'_{ijk} couplings. See `IMSS(52)`, `RMSS(52)`.

`RVLAMB(3,3,3)` : the baryon number violating λ''_{ijk} couplings. Currently not used.

The following subroutines and functions need not be accessed by the user, but are described for completeness.

`SUBROUTINE PYAPPS` : uses approximate analytic formulae to determine the full set of MSSM parameters from SUGRA inputs.

`SUBROUTINE PYGLUI` : calculates gluino decay modes.

`SUBROUTINE PYGQQB` : calculates three body decays of gluinos into neutralinos or charginos and third generation fermions. These routines are valid for large values of $\tan\beta$.

`SUBROUTINE PYCJDC` : calculates the chargino decay modes.

`SUBROUTINE PYHEXT` : calculates the non–Standard Model decay modes of the Higgs bosons.

`SUBROUTINE PYHGGM` : determines the Higgs boson mass spectrum using several inputs.

`SUBROUTINE PYINOM` : finds the mass eigenstates and mixing matrices for the charginos and neutralinos.

`SUBROUTINE PYMSIN` : initializes the MSSM simulation.

`SUBROUTINE PYNJDC` : calculates neutralino decay modes.

`SUBROUTINE PYPOLE` : computes the Higgs boson masses using a renormalization group improved leading–log approximation and two-loop leading–log corrections.

`SUBROUTINE PYRNMT` : determines the running mass of the top quark.

`SUBROUTINE PYSFDC` : calculates sfermion decay modes.

`SUBROUTINE PYSUBH` : computes the Higgs boson masses using only renormalization group improved formulae.

`SUBROUTINE PYTBDY` : samples the phase space for three body decays of neutralinos, charginos, and the gluino.

`SUBROUTINE PYTHRG` : computes the masses and mixing matrices of the third generation sfermions.

`SUBROUTINE PYRVSF` : R-violating sfermion decay widths.

`SUBROUTINE PYRVNE` : R-violating neutralino decay widths.

`SUBROUTINE PYRVCH` : R-violating chargino decay widths.

`SUBROUTINE PYRVGW` : calculates R-violating 3-body widths using `PYRVI1`, `PYRVI2`, `PYRVI3`, `PYRVG1`, `PYRVG2`, `PYRVG3`, `PYRVG4`, `PYRVR`, and `PYRVS`.

`FUNCTION PYRVSB` : calculates R-violating 2-body widths.

9.6 General Event Information

When an event is generated with `PYEVNT`, some information on it is stored in the `MSTI` and `PARI` arrays of the `PYPARS` common block (often copied directly from the internal `MINT` and `VINT` variables). Further information is stored in the complete event record; see section 5.2.

Part of the information is only relevant for some subprocesses; by default everything irrelevant is set to 0. Kindly note that, like the `CKIN` constraints described in section 9.2, kinematical variables normally (i.e. where it is not explicitly stated otherwise) refer to the naïve hard scattering, before initial- and final-state radiation effects have been included.

```
COMMON/PYPARS/MSTP(200),PARP(200),MSTI(200),PARI(200)
```

Purpose: to provide information on latest event generated or, in a few cases, on statistics accumulated during the run.

`MSTI(1)` : specifies the general type of subprocess that has occurred, according to the `ISUB` code given in section 8.1.

`MSTI(2)` : whenever `MSTI(1)` (together with `MSTI(15)` and `MSTI(16)`) are not enough to specify the type of process uniquely, `MSTI(2)` provides an ordering of the different possibilities. This is particularly relevant for the different colour-flow topologies possible in QCD $2 \to 2$ processes. With $i =$`MSTI(15)`, $j =$`MSTI(16)` and

$k =$`MSTI(2)`, the QCD possibilities are, in the classification scheme of [Ben84] (cf. section 8.2.1):

ISUB = 11, $i = j$, $q_i q_i \to q_i q_i$;
$k = 1$: colour configuration A.
$k = 2$: colour configuration B.

ISUB = 11, $i \neq j$, $q_i q_j \to q_i q_j$;
$k = 1$: only possibility.

ISUB = 12, $q_i \overline{q}_i \to q_l \overline{q}_l$;
$k = 1$: only possibility.

ISUB = 13, $q_i \overline{q}_i \to gg$;
$k = 1$: colour configuration A.
$k = 2$: colour configuration B.

ISUB = 28, $q_i g \to q_i g$;
$k = 1$: colour configuration A.
$k = 2$: colour configuration B.

ISUB = 53, $gg \to q_l \overline{q}_l$;
$k = 1$: colour configuration A.
$k = 2$: colour configuration B.

ISUB = 68, $gg \to gg$;
$k = 1$: colour configuration A.
$k = 2$: colour configuration B.
$k = 3$: colour configuration C.

ISUB = 83, $fq \to f'Q$ (by t-channel W exchange; does not distinguish colour flows but result of user selection);
$k = 1$: heavy flavour Q is produced on side 1.
$k = 2$: heavy flavour Q is produced on side 2.

`MSTI(3)` : the number of partons produced in the hard interactions, i.e. the number n of the $2 \to n$ matrix elements used; it is sometimes 3 or 4 when a basic $2 \to 1$ or $2 \to 2$ process has been folded with two $1 \to 2$ initial branchings (like $q_i q_j \to q_k q_l h^0$).

`MSTI(4)` : number of documentation lines at the beginning of the common block `PYJETS` that are given with `K(I,1)=21`; 0 for `MSTP(125)=0`.

`MSTI(5)` : number of events generated to date in current run. In runs with the variable-energy option, `MSTP(171)=1` and `MSTP(172)=2`, only those events that survive (i.e. that do not have `MSTI(61)=1`) are counted in this number. That is, `MSTI(5)` may be less than the total number of `PYEVNT` calls.

`MSTI(6)` : current frame of event, cf. `MSTP(124)`.

`MSTI(7), MSTI(8)` : line number for documentation of outgoing partons/particles from hard scattering for $2 \to 2$ or $2 \to 1 \to 2$ processes (else = 0).

`MSTI(9)` : event class used in current event for γp or $\gamma\gamma$ events. The code depends on which process is being studied.

`= 0` : for other processes than the ones listed above.

For γp or $\gamma^* p$ events, generated with the `MSTP(14)=10` or `MSTP(14)=30` options:

`= 1` : VMD.
`= 2` : direct.
`= 3` : anomalous.
`= 4` : DIS (only for $\gamma^* p$, i.e. `MSTP(14)=30`).

For real $\gamma\gamma$ events, i.e. `MSTP(14)=10`:

`= 1` : VMD$\times$VMD.
`= 2` : VMD$\times$direct.
`= 3` : VMD$\times$anomalous .
`= 4` : direct$\times$direct.
`= 5` : direct$\times$anomalous.
`= 6` : anomalous$\times$anomalous.

For virtual $\gamma^*\gamma^*$ events, i.e. `MSTP(14)=30`, where the two incoming photons are not equivalent and the order therefore matters:

`= 1 :` direct×direct.
`= 2 :` direct×VMD.
`= 3 :` direct×anomalous.
`= 4 :` VMD×direct.
`= 5 :` VMD×VMD.
`= 6 :` VMD×anomalous.
`= 7 :` anomalous×direct.
`= 8 :` anomalous×VMD.
`= 9 :` anomalous×anomalous.
`= 10 :` DIS×VMD.
`= 11 :` DIS×anomalous.
`= 12 :` VMD×DIS.
`= 13 :` anomalous×DIS.

`MSTI(10)` : is 1 if cross section maximum was violated in current event, and 0 if not.

`MSTI(11)` : `KF` flavour code for beam (side 1) particle.

`MSTI(12)` : `KF` flavour code for target (side 2) particle.

`MSTI(13), MSTI(14)` : `KF` flavour codes for side 1 and side 2 initial-state shower initiators.

`MSTI(15), MSTI(16)` : `KF` flavour codes for side 1 and side 2 incoming partons to the hard interaction.

`MSTI(17), MSTI(18)` : flag to signal if particle on side 1 or side 2 has been scattered diffractively; 0 if no, 1 if yes.

`MSTI(21) - MSTI(24)` : `KF` flavour codes for outgoing partons from the hard interaction. The number of positions actually used is process-dependent, see `MSTI(3)`; trailing positions not used are set = 0. For events with many outgoing partons, e.g. in external processes, also `MSTI(25)` and `MSTI(26)` could be used.

`MSTI(25), MSTI(26)` : `KF` flavour codes of the products in the decay of a single *s*-channel resonance formed in the hard interaction. Are thus only used when `MSTI(3)=1` and the resonance is allowed to decay.

`MSTI(31)` : number of hard or semi-hard scatterings that occurred in the current event in the multiple-interaction scenario; is = 0 for a low-$p_\perp$ event.

`MSTI(32)` : information on whether a reconnection occurred in the current event; is 0 normally but 1 in case of reconnection.

`MSTI(41)` : the number of pile-up events generated in the latest `PYEVNT` call (including the first, 'hard' event).

`MSTI(42) - MSTI(50)` : `ISUB` codes for the events 2–10 generated in the pile-up-events scenario. The first event `ISUB` code is stored in `MSTI(1)`. If `MSTI(41)` is less than 10, only as many positions are filled as there are pile-up events. If MSTI(41) is above 10, some `ISUB` codes will not appear anywhere.

`MSTI(51)` : normally 0 but set to 1 if a `UPEVNT` call did not return an event, such that `PYEVNT` could not generate an event. For further details, see section 9.9.

`MSTI(52)` : counter for the number of times the current event configuration failed in the generation machinery. For accepted events this is always 0, but the counter can be used inside `UPEVNT` to check on anomalous occurrences. For further details, see section 9.9.

`MSTI(53)` : normally 0, but 1 if no processes with non-vanishing cross sections were found in a `PYINIT` call, for the case that `MSTP(127)=1`.

`MSTI(61)` : status flag set when events are generated. It is only of interest for runs with variable energies, `MSTP(171)=1`, with the option `MSTP(172)=2`.

`= 0 :` an event has been generated.
`= 1 :` no event was generated, either because the c.m. energy was too low or because the Monte Carlo phase space point selection machinery rejected the trial point.

A new energy is to be picked by you.

`MSTI(71), MSTI(72)` : `KF` code for incoming lepton beam or target particles, when a flux of virtual photons are generated internally for `'gamma/`*lepton*`'` beams, while `MSTI(11)` and `MSTI(12)` is then the photon code.

`PARI(1)` : total integrated cross section for the processes under study, in mb. This number is obtained as a by-product of the selection of hard-process kinematics, and is thus known with better accuracy when more events have been generated. The value stored here is based on all events until the latest one generated.

`PARI(2)` : for unweighted events, `MSTP(142)=0` or `=2`, it is the ratio `PARI(1)/MSTI(5)`, i.e. the ratio of total integrated cross section and number of events generated. Histograms should then be filled with unit event weight and, at the end of the run, multiplied by `PARI(2)` and divided by the bin width to convert results to mb/(dimension of the horizontal axis). For weighted events, `MSTP(142)=1`, `MSTI(5)` is replaced by the sum of `PARI(10)` values. Histograms should then be filled with event weight `PARI(10)` and, as before, be multiplied by `PARI(2)` and divided by the bin width at the end of the run. In runs with the variable-energy option, `MSTP(171)=1` and `MSTP(172)=2`, only those events that survive (i.e. that do not have `MSTI(61)=1`) are counted.

`PARI(7)` : an event weight, normally 1 and thus uninteresting, but for external processes with `IDWTUP=-1, -2` or `-3` it can be -1 for events with negative cross section, with `IDWTUP=4` it can be an arbitrary non-negative weight of dimension mb, and with `IDWTUP=-4` it can be an arbitrary weight of dimension mb. (The difference being that in most cases a rejection step is involved to bring the accepted events to a common weight normalization, up to a sign, while no rejection need be involved in the last two cases.)

`PARI(9)` : is weight `WTXS` returned from `PYEVWT` call when `MSTP(142)`≥ 1, otherwise is 1.

`PARI(10)` : is compensating weight `1./WTXS` that should be associated to events when `MSTP(142)=1`, else is 1.

`PARI(11)` : $E_{\rm cm}$, i.e. total c.m. energy (except when using the `'gamma/`*lepton*`'` machinery, see `PARI(101)`.

`PARI(12)` : s, i.e. squared total c.m. energy (except when using the `'gamma/`*lepton*`'` machinery, see `PARI(102)`.

`PARI(13)` : $\hat{m} = \sqrt{\hat{s}}$, i.e. mass of the hard-scattering subsystem.

`PARI(14)` : $\hat{s}$ of the hard subprocess ($2 \to 2$ or $2 \to 1$).

`PARI(15)` : $\hat{t}$ of the hard subprocess ($2 \to 2$ or $2 \to 1 \to 2$).

`PARI(16)` : $\hat{u}$ of the hard subprocess ($2 \to 2$ or $2 \to 1 \to 2$).

`PARI(17)` : $\hat{p}_\perp$ of the hard subprocess ($2 \to 2$ or $2 \to 1 \to 2$), evaluated in the rest frame of the hard interaction.

`PARI(18)` : $\hat{p}_\perp^2$ of the hard subprocess; see `PARI(17)`.

`PARI(19)` : $\hat{m}'$, the mass of the complete three- or four-body final state in $2 \to 3$ or $2 \to 4$ processes (while $\hat{m}$, given in `PARI(13)`, here corresponds to the one- or two-body central system). Kinematically $\hat{m} \leq \hat{m}' \leq E_{\rm cm}$.

`PARI(20)` : $\hat{s}' = \hat{m}'^2$; see `PARI(19)`.

`PARI(21)` : Q of the hard-scattering subprocess. The exact definition is process-dependent, see `MSTP(32)`.

`PARI(22)` : Q^2 of the hard-scattering subprocess; see `PARI(21)`.

`PARI(23)` : Q of the outer hard-scattering subprocess. Agrees with `PARI(21)` for a $2 \to 1$ or $2 \to 2$ process. For a $2 \to 3$ or $2 \to 4$ W/Z fusion process, it is set by the W/Z mass scale, and for subprocesses 121 and 122 by the heavy-quark mass.

`PARI(24)` : Q^2 of the outer hard-scattering subprocess; see `PARI(23)`.

`PARI(25)` : Q scale used as maximum virtuality in parton showers. Is equal to `PARI(23)`, except for Deeply Inelastic Scattering processes when `MSTP(22)`≥ 1.

PARI(26) : Q^2 scale in parton showers; see PARI(25).

PARI(31), PARI(32) : the momentum fractions x of the initial-state parton-shower initiators on side 1 and 2, respectively.

PARI(33), PARI(34) : the momentum fractions x taken by the partons at the hard interaction, as used e.g. in the parton-distribution functions.

PARI(35) : Feynman-x, $x_{\rm F} = x_1 - x_2 =$PARI(33)$-$PARI(34).

PARI(36) : $\tau = \hat{s}/s = x_1 x_2 =$PARI(33)$\times$PARI(34).

PARI(37) : $y = (1/2)\ln(x_1/x_2)$, i.e. rapidity of the hard-interaction subsystem in the c.m. frame of the event as a whole.

PARI(38) : $\tau' = \hat{s}'/s =$PARI(20)/PARI(12).

PARI(39), PARI(40) : the primordial $k_\perp$ values selected in the two beam remnants.

PARI(41) : $\cos\hat{\theta}$, where $\hat{\theta}$ is the scattering angle of a $2 \to 2$ (or $2 \to 1 \to 2$) interaction, defined in the rest frame of the hard-scattering subsystem.

PARI(42) : $x_\perp$, i.e. scaled transverse momentum of the hard-scattering subprocess, $x_\perp = 2\hat{p}_\perp/E_{\rm cm} =$ 2PARI(17)/PARI(11).

PARI(43), PARI(44) : x_{L3} and x_{L4}, i.e. longitudinal momentum fractions of the two scattered partons, in the range $-1 < x_{\rm L} < 1$, in the c.m. frame of the event as a whole.

PARI(45), PARI(46) : x_3 and x_4, i.e. scaled energy fractions of the two scattered partons, in the c.m. frame of the event as a whole.

PARI(47), PARI(48) : y_3^* and y_4^*, i.e. rapidities of the two scattered partons in the c.m. frame of the event as a whole.

PARI(49), PARI(50) : η_3^* and η_4^*, i.e. pseudorapidities of the two scattered partons in the c.m. frame of the event as a whole.

PARI(51), PARI(52) : $\cos\theta_3^*$ and $\cos\theta_4^*$, i.e. cosines of the polar angles of the two scattered partons in the c.m. frame of the event as a whole.

PARI(53), PARI(54) : θ_3^* and θ_4^*, i.e. polar angles of the two scattered partons, defined in the range $0 < \theta^* < \pi$, in the c.m. frame of the event as a whole.

PARI(55), PARI(56) : azimuthal angles ϕ_3^* and ϕ_4^* of the two scattered partons, defined in the range $-\pi < \phi^* < \pi$, in the c.m. frame of the event as a whole.

PARI(61) : multiple interaction enhancement factor for current event. A large value corresponds to a central collision and a small value to a peripheral one.

PARI(65) : sum of the transverse momenta of partons generated at the hardest interaction of the event, excluding initial- and final-state radiation, i.e. 2$\times$PARI(17). Only intended for $2 \to 2$ or $2 \to 1 \to 2$ processes, i.e. not implemented for $2 \to 3$ ones.

PARI(66) : sum of the transverse momenta of all partons generated at the hardest interaction, including initial- and final-state radiation, resonance decay products, and primordial $k_\perp$.

PARI(67) : scalar sum of transverse momenta of partons generated at hard interactions, excluding the hardest one (see PARI(65)), and also excluding all initial- and final-state radiation. Is non-vanishing only in the multiple-interaction scenario.

PARI(68) : sum of transverse momenta of all partons generated at hard interactions, excluding the hardest one (see PARI(66)), but including initial- and final-state radiation associated with those further interactions. Is non-vanishing only in the multiple-interaction scenario. Since showering has not yet been added to those additional interactions, it currently coincides with PARI(67).

PARI(69) : sum of transverse momenta of all partons generated in hard interactions (PARI(66) + PARI(68)) and, additionally, of all beam remnant partons.

PARI(71), PARI(72) : sum of the momentum fractions x taken by initial-state parton-shower initiators on side 1 and and side 2, excluding those of the hardest interaction. Is non-vanishing only in the multiple-interaction scenario.

PARI(73), PARI(74) : sum of the momentum fractions x taken by the partons at the hard interaction on side 1 and side 2, excluding those of the hardest interaction. Is non-vanishing only in the multiple-interaction scenario.

`PARI(75)`, `PARI(76)` : the x value of a photon that branches into quarks or gluons, i.e. x at interface between initial-state QED and QCD cascades, for the old photoproduction machinery..

`PARI(77)`, `PARI(78)` : the χ values selected for beam remnants that are split into two objects, describing how the energy is shared (see `MSTP(92)` and `MSTP(94)`); is vanishing if no splitting is needed.

`PARI(81)` : size of the threshold factor (enhancement or suppression) in the latest event with heavy-flavour production; see `MSTP(35)`.

`PARI(91)` : average multiplicity $\overline{n}$ of pile-up events, see `MSTP(133)`. Only relevant for `MSTP(133)`= 1 or 2.

`PARI(92)` : average multiplicity $\langle n \rangle$ of pile-up events as actually simulated, i.e. with multiplicity = 0 events removed and the high-end tail truncated. Only relevant for `MSTP(133)`= 1 or 2.

`PARI(93)` : for `MSTP(133)=1` it is the probability that a beam crossing will produce a pile-up event at all, i.e. that there will be at least one hadron–hadron interaction; for `MSTP(133)=2` the probability that a beam crossing will produce a pile-up event with one hadron–hadron interaction of the desired rare type. See subsection 11.3.

`PARI(101)` : c.m. energy for the full collision, while `PARI(11)` gives the γ-hadron or $\gamma\gamma$ subsystem energy; used for virtual photons generated internally with the `'gamma/`*lepton*`'` option.

`PARI(102)` : full squared c.m. energy, while `PARI(12)` gives the subsystem squared energy; used for virtual photons generated internally with the `'gamma/`*lepton*`'` option.

`PARI(103)`, `PARI(104)` : x values, i.e. respective photon energy fractions of the incoming lepton in the c.m. frame of the event; used for virtual photons generated internally with the `'gamma/`*lepton*`'` option.

`PARI(105)`, `PARI(106)` : Q^2 or P^2, virtuality of the respective photon (thus the square of `VINT(3)`, `VINT(4)`); used for virtual photons generated internally with the `'gamma/`*lepton*`'` option.

`PARI(107)`, `PARI(108)` : y values, i.e. respective photon light-cone energy fraction of the incoming lepton; used for virtual photons generated internally with the `'gamma/`*lepton*`'` option.

`PARI(109)`, `PARI(110)` : θ, scattering angle of the respective lepton in the c.m. frame of the event; used for virtual photons generated internally with the `'gamma/`*lepton*`'` option.

`PARI(111)`, `PARI(112)` : ϕ, azimuthal angle of the respective scattered lepton in the c.m. frame of the event; used for virtual photons generated internally with the `'gamma/`*lepton*`'` option.

`PARI(113)`, `PARI(114)`: the R factor defined at `MSTP(17)`, giving a cross section enhancement from the contribution of resolved longitudinal photons.

9.7 How to Generate Weighted Events

By default PYTHIA generates unweighted events, i.e. all events in a run are on an equal footing. This means that corners of phase space with low cross sections are poorly populated, as it should be. However, sometimes one is interested in also exploring such corners, in order to gain a better undestanding of physics. A typical example would be the jet cross section in hadron collisions, which is dropping rapidly with increasing jet $p_\perp$, and where it is interesting to trace this drop over several orders of magnitude. Experimentally this may be solved by prescaling events rates already at the trigger level, so that all high-$p_\perp$ events are saved but only a fraction of the lower-$p_\perp$ ones. In this section we outline procedures to generate events in a similar manner.

Basically two approaches can be used. One is to piece together results from different subruns, where each subrun is restricted to some specific region of phase space. Within each subrun all events then have the same weight, but subruns have to be combined according to their rel-

ative cross sections. The other approach is to let each event come with an associated weight, that can vary smoothly as a function of $p_\perp$. These two alternatives correspond to stepwise or smoothly varying prescaling factors in the experimemental analogue. We describe them one after the other.

The phase space can be sliced in many different ways. However, for the jet rate and many other processes, the most natural variable would be $p_\perp$ itself. (For production of a lepton pair by s-channel resonances, the invariant mass would be a better choice.) It is not possible to specify beforehand the jet $p_\perp$'s an event will contain, since this is a combination of the $\hat{p}_\perp$ of the hard scattering process with additional showering activity, with hadronization, with underlying event and with the jet clustering approach actually used. However, one would expect a strong correlation between the $\hat{p}_\perp$ scale and the jet $p_\perp$'s. Therefore the full $\hat{p}_\perp$ range can be subdivided into a set of ranges by using the `CKIN(3)` and `CKIN(4)` variables as lower and upper limits. This could be done e.g. for adjacent non-overlapping bins 10–20,20–40,40–70, etc.

Only if one would like to cover also very small $p_\perp$ is there a problem with this strategy: since the naive jet cross section is divergent for $\hat{p}_\perp \to 0$, a unitarization procedure is implied by setting `CKIN(3)=0` (or some other low value). This unitarization then disregards the actual `CKIN(3)` and `CKIN(4)` values and generates events over the full phase space. In order not to doublecount, then events above the intended upper limit of the first bin have to be removed by brute force.

A simple but complete example of a code performing this task (with some primitive histogramming) is the following:

```
C...All real arithmetic in double precision.
      IMPLICIT DOUBLE PRECISION(A-H, O-Z)
C...Three Pythia functions return integers, so need declaring.
      INTEGER PYK,PYCHGE,PYCOMP
C...EXTERNAL statement links PYDATA on most platforms.
      EXTERNAL PYDATA
C...The event record.
      COMMON/PYJETS/N,NPAD,K(4000,5),P(4000,5),V(4000,5)
C...Selection of hard scattering subprocesses.
      COMMON/PYSUBS/MSEL,MSELPD,MSUB(500),KFIN(2,-40:40),CKIN(200)
C...Parameters.
      COMMON/PYPARS/MSTP(200),PARP(200),MSTI(200),PARI(200)
C...Bins of pT.
      DIMENSION PTBIN(10)
      DATA PTBIN/0D0,10D0,20D0,40D0,70D0,110D0,170D0,250D0,350D0,1000D0/

C...Main parameters of run: c.m.\ energy and number of events per bin.
      ECM=2000D0
      NEV=1000

C...Histograms.
      CALL PYBOOK(1,'dn_ev/dpThat',100,0D0,500D0)
      CALL PYBOOK(2,'dsigma/dpThat',100,0D0,500D0)
      CALL PYBOOK(3,'log10(dsigma/dpThat)',100,0D0,500D0)
      CALL PYBOOK(4,'dsigma/dpTjet',100,0D0,500D0)
      CALL PYBOOK(5,'log10(dsigma/dpTjet)',100,0D0,500D0)
      CALL PYBOOK(11,'dn_ev/dpThat, dummy',100,0D0,500D0)
      CALL PYBOOK(12,'dn/dpTjet, dummy',100,0D0,500D0)

C...Loop over pT bins and initialize.
      DO 300 IBIN=1,9
        CKIN(3)=PTBIN(IBIN)
```

```
        CKIN(4)=PTBIN(IBIN+1)
        CALL PYINIT('CMS','p','pbar',ECM)

C...Loop over events. Remove unwanted ones in first pT bin.
        DO 200 IEV=1,NEV
          CALL PYEVNT
          PTHAT=PARI(17)
          IF(IBIN.EQ.1.AND.PTHAT.GT.PTBIN(IBIN+1)) GOTO 200

C...Store pThat. Cluster jets and store variable number of pTjet.
          CALL PYFILL(1,PTHAT,1D0)
          CALL PYFILL(11,PTHAT,1D0)
          CALL PYCELL(NJET)
          DO 100 IJET=1,NJET
            CALL PYFILL(12,P(N+IJET,5),1D0)
  100     CONTINUE

C...End of event loop.
  200   CONTINUE

C...Normalize cross section to pb/GeV and add up.
        FAC=1D9*PARI(1)/(DBLE(NEV)*5D0)
        CALL PYOPER(2,'+',11,2,1D0,FAC)
        CALL PYOPER(4,'+',12,4,1D0,FAC)

C...End of loop over pT bins.
  300 CONTINUE

C...Take logarithm and plot.
      CALL PYOPER(2,'L',2,3,1D0,0D0)
      CALL PYOPER(4,'L',4,5,1D0,0D0)
      CALL PYNULL(11)
      CALL PYNULL(12)
      CALL PYHIST

      END
```

The alternative to slicing the phase space is to used weighted events. This is possible by making use of the `PYEVWT` routine:

```
CALL PYEVWT(WTXS)
```

Purpose: to allow you to reweight event cross sections, by process type and kinematics of the hard scattering. There exists two separate modes of usage, described in the following.

For `MSTP(142)=1`, it is assumed that the cross section of the process is correctly given by default in PYTHIA, but that one wishes to generate events biased to a specific region of phase space. While the `WTXS` factor therefore multiplies the naïve cross section in the choice of subprocess type and kinematics, the produced event comes with a compensating weight `PARI(10)=1./WTXS`, which should be used when filling histograms etc. In the `PYSTAT(1)` table, the cross sections are unchanged (up to statistical errors) compared with the standard cross sections, but the relative composition of events may be changed and need no longer be in propor-

tion to relative cross sections. A typical example of this usage is if one wishes to enhance the production of high-$p_\perp$ events; then a weight like WTXS= $(p_\perp/p_{\perp 0})^2$ (with $p_{\perp 0}$ some fixed number) might be appropriate. See PARI(2) for a discussion of overall normalization issues.

For MSTP(142)=2, on the other hand, it is assumed that the true cross section is really to be modified by the multiplicative factor WTXS. The generated events therefore come with unit weight, just as usual. This option is really equivalent to replacing the basic cross sections coded in PYTHIA, but allows more flexibility: no need to recompile the whole of PYTHIA.

The routine will not be called unless MSTP(142)≥ 1, and never if 'minimum-bias'-type events (including elastic and diffractive scattering) are to be generated as well. Further, cross sections for additional multiple interactions or pile-up events are never affected. A dummy routine PYEVWT is included in the program file, so as to avoid unresolved external references when the routine is not used.

WTXS: multiplication factor to ordinary event cross section; to be set (by you) in PYEVWT call.

Remark : at the time of selection, several variables in the MINT and VINT arrays in the PYINT1 common block contain information that can be used to make the decision. The routine provided in the program file explicitly reads the variables that have been defined at the time PYEVWT is called, and also calculates some derived quantities. The given list of information includes subprocess type ISUB, $E_{\rm cm}$, $\hat{s}$, $\hat{t}$, $\hat{u}$, $\hat{p}_\perp$, x_1, x_2, $x_{\rm F}$, τ, y, τ', $\cos\hat{\theta}$, and a few more. Some of these may not be relevant for the process under study, and are then set to zero.

Warning: the weights only apply to the hard scattering subprocesses. There is no way to reweight the shape of initial- and final-state showers, fragmentation, or other aspects of the event.

There are some limitations to the facility. PYEVWT is called at an early stage of the generation process, when the hard kinematics is selected, well before the full event is constructed. It then cannot be used for low-$p_\perp$, elastic or diffractive events, for which no hard kinematics has been defined. If such processes are included, the event weighting is switched off. Therefore it is no longer an option to run with CKIN(3)=0.

Which weight expression to use may take some trial and error. In the above case, a reasonable ansatz seems to be a weight behaving like $\hat{p}_\perp^6$, where four powers of $\hat{p}_\perp$ are motivated by the partonic cross section behaving like $1/\hat{p}_\perp^4$, and the remaining two by the fall-off of parton densities. An example for the same task as above one would then be:

```
C...All real arithmetic in double precision.
      IMPLICIT DOUBLE PRECISION(A-H, O-Z)
C...Three Pythia functions return integers, so need declaring.
      INTEGER PYK,PYCHGE,PYCOMP
C...EXTERNAL statement links PYDATA on most platforms.
      EXTERNAL PYDATA
C...The event record.
      COMMON/PYJETS/N,NPAD,K(4000,5),P(4000,5),V(4000,5)
C...Selection of hard scattering subprocesses.
      COMMON/PYSUBS/MSEL,MSELPD,MSUB(500),KFIN(2,-40:40),CKIN(200)
C...Parameters.
      COMMON/PYPARS/MSTP(200),PARP(200),MSTI(200),PARI(200)

C...Main parameters of run: c.m.\ energy, pTmin and number of events.
      ECM=2000D0
      CKIN(3)=5D0
      NEV=10000
```

```
C...Histograms.
      CALL PYBOOK(1,'dn_ev/dpThat',100,0D0,500D0)
      CALL PYBOOK(2,'dsigma/dpThat',100,0D0,500D0)
      CALL PYBOOK(3,'log10(dsigma/dpThat)',100,0D0,500D0)
      CALL PYBOOK(4,'dsigma/dpTjet',100,0D0,500D0)
      CALL PYBOOK(5,'log10(dsigma/dpTjet)',100,0D0,500D0)

C...Initialize with weighted events.
      MSTP(142)=1
      CALL PYINIT('CMS','p','pbar',ECM)

C...Loop over events; read out pThat and event weight.
      DO 200 IEV=1,NEV
        CALL PYEVNT
        PTHAT=PARI(17)
        WT=PARI(10)

C...Store pThat. Cluster jets and store variable number of pTjet.
        CALL PYFILL(1,PTHAT,1D0)
        CALL PYFILL(2,PTHAT,WT)
        CALL PYCELL(NJET)
        DO 100 IJET=1,NJET
          CALL PYFILL(4,P(N+IJET,5),WT)
  100   CONTINUE

C...End of event loop.
  200   CONTINUE

C...Normalize cross section to pb/GeV, take logarithm and plot.
      FAC=1D9*PARI(2)/5D0
      CALL PYFACT(2,FAC)
      CALL PYFACT(4,FAC)
      CALL PYOPER(2,'L',2,3,1D0,0D0)
      CALL PYOPER(4,'L',4,5,1D0,0D0)
      CALL PYHIST

      END

C*************************************************************

      SUBROUTINE PYEVWT(WTXS)

C...Double precision and integer declarations.
      IMPLICIT DOUBLE PRECISION(A-H, O-Z)
      IMPLICIT INTEGER(I-N)
      INTEGER PYK,PYCHGE,PYCOMP
C...Commonblock.
      COMMON/PYINT1/MINT(400),VINT(400)

C...Read out pThat^2 and set weight.
      PT2=VINT(48)
      WTXS=PT2**3
```

```
      RETURN
      END
```

Note that, in PYEVWT one cannot look for $\hat{p}_\perp$ in PARI(17), since this variable is only set at the end of the event generation. Instead the internal VINT(48) is used. The dummy copy of the PYEVWT routine found in the PYTHIA code shows what is available and how to access this.

9.8 How to Run with Varying Energies

It is possible to use PYTHIA in a mode where the energy can be varied from one event to the next, without the need to re-initialize with a new PYINIT call. This allows a significant speed-up of execution, although it is not as fast as running at a fixed energy. It can not be used for everything — we will come to the fine print at the end — but it should be applicable for most tasks.

The master switch to access this possibility is in MSTP(171). By default it is off, so you must set MSTP(171)=1 before initialization. There are two submodes of running, with MSTP(172) being 1 or 2. In the former mode, PYTHIA will generate an event at the requested energy. This means that you have to know which energy you want beforehand. In the latter mode, PYTHIA will often return without having generated an event — with flag MSTI(61)=1 to signal that — and you are then requested to give a new energy. The energy spectrum of accepted events will then, in the end, be your naive input spectrum weighted with the cross-section of the processes you study. We will come back to this.

The energy can be varied, whichever frame is given in the PYINIT call. (Except for 'USER', where such information is fed in via the HEPEUP common block and thus beyond the control of PYTHIA.) When the frame is 'CMS', PARP(171) should be filled with the fractional energy of each event, i.e. $E_{\rm cm}$ =PARP(171)×WIN, where WIN is the nominal c.m. energy of the PYINIT call. Here PARP(171) should normally be smaller than unity, i.e. initialization should be done at the maximum energy to be encountered. For the 'FIXT' frame, PARP(171) should be filled by the fractional beam energy of that one, i.e. $E_{\rm beam}$ =PARP(171)×WIN. For the '3MOM', '4MOM' and '5MOM' options, the two four-momenta are given in for each event in the same format as used for the PYINIT call. Note that there is a minimum c.m. energy allowed, PARP(2). If you give in values below this, the program will stop for MSTP(172)=1, and will return with MSTI(61)=1 for MSTP(172)=1.

To illustrate the use of the MSTP(172)=2 facility, consider the case of beamstrahlung in e^+e^- linear colliders. This is just for convenience; what is said here can be translated easily into other situations. Assume that the beam spectrum is given by $D(z)$, where z is the fraction retained by the original e after beamstrahlung. Therefore $0 \le z \le 1$ and the integral of $D(z)$ is unity. This is not perfectly general; one could imagine branchings $e^- \to e^-\gamma \to e^-e^+e^-$, which gives a multiplication in the number of beam particles. This could either be expressed in terms of a $D(z)$ with integral larger than unity or in terms of an increased luminosity. We will assume the latter, and use $D(z)$ properly normalized. Given a nominal $s = 4E^2_{\rm beam}$, the actual s' after beamstrahlung is given by $s' = z_1 z_2 s$. For a process with a cross section $\sigma(s)$ the total cross section is then

$$\sigma_{\rm tot} = \int_0^1 \int_0^1 D(z_1)\, D(z_2) \sigma(z_1 z_2 s)\, {\rm d}z_1\, {\rm d}z_2 \ . \tag{142}$$

The cross section σ may in itself be an integral over a number of additional phase space variables. If the maximum of the differential cross section is known, a correct procedure to generate events is

1. pick z_1 and z_2 according to $D(z_1)\,{\rm d}z_1$ and $D(z_2)\,{\rm d}z_2$, respectively;
2. pick a set of phase space variables of the process, for the given s' of the event;
3. evaluate $\sigma(s')$ and compare with $\sigma_{\rm max}$;

4. if event is rejected, then return to step 1 to generate new variables;
5. else continue the generation to give a complete event.

You as a user are assumed to take care of step 1, and present the resulting kinematics with incoming e^+ and e^- of varying energy. Thereafter PYTHIA will do steps 2–5, and either return an event or put `MSTI(61)=1` to signal failure in step 4.

The maximization procedure does search in phase space to find $\sigma_{\rm max}$, but it does not vary the s' energy in this process. Therefore the maximum search in the `PYINIT` call should be performed where the cross section is largest. For processes with increasing cross section as a function of energy this means at the largest energy that will ever be encountered, i.e. $s' = s$ in the case above. This is the 'standard' case, but often one encounters other behaviours, where more complicated procedures are needed. One such case would be the process $e^+e^- \to Z^{*0} \to Z^0h^0$, which is known to have a cross section that increases near the threshold but is decreasing asymptotically. If one already knows that the maximum, for a given Higgs mass, appears at 300 GeV, say, then the `PYINIT` call should be made with that energy, even if subsequently one will be generating events for a 500 GeV collider.

In general, it may be necessary to modify the selection of z_1 and z_2 and assign a compensating event weight. For instance, consider a process with a cross section behaving roughly like $1/s$. Then the $\sigma_{\rm tot}$ expression above may be rewritten as

$$\sigma_{\rm tot} = \int_0^1 \int_0^1 \frac{D(z_1)}{z_1} \frac{D(z_2)}{z_2} z_1 z_2 \sigma(z_1 z_2 s)\, {\rm d}z_1\, {\rm d}z_2 \; . \tag{143}$$

The expression $z_1 z_2 \sigma(s')$ is now essentially flat in s', i.e. not only can $\sigma_{\rm max}$ be found at a convenient energy such as the maximum one, but additionally the PYTHIA generation efficiency (the likelihood of surviving step 4) is greatly enhanced. The price to be paid is that z has to be selected according to $D(z)/z$ rather than according to $D(z)$. Note that $D(z)/z$ is not normalized to unity. One therefore needs to define

$$I_D = \int_0^1 \frac{D(z)}{z}\, {\rm d}z \; , \tag{144}$$

and a properly normalized

$$D'(z) = \frac{1}{I_D} \frac{D(z)}{z} \; . \tag{145}$$

Then

$$\sigma_{\rm tot} = \int_0^1 \int_0^1 D'(z_1)\, D'(z_2)\; I_D^2 z_1 z_2 \sigma(z_1 z_2 s)\, {\rm d}z_1\, {\rm d}z_2 \; . \tag{146}$$

Therefore the proper event weight is $I_D^2 z_1 z_2$. This weight should be stored by you, for each event, in `PARP(173)`. The maximum weight that will be encountered should be stored in `PARP(174)` before the `PYINIT` call, and not changed afterwards. It is not necessary to know the precise maximum; any value larger than the true maximum will do, but the inefficiency will be larger the cruder the approximation. Additionally you must put `MSTP(173)=1` for the program to make use of weights at all. Often $D(z)$ is not known analytically; therefore I_D is also not known beforehand, but may have to be evaluated (by you) during the course of the run. Then you should just use the weight $z_1 z_2$ in `PARP(173)` and do the overall normalization yourself in the end. Since `PARP(174)=1` by default, in this case you need not set this variable specially. Only the cross sections are affected by the procedure selected for overall normalization, the events themselves still are properly distributed in s' and internal phase space.

Above it has been assumed tacitly that $D(z) \to 0$ for $z \to 0$. If not, $D(z)/z$ is divergent, and it is not possible to define a properly normalized $D'(z) = D(z)/z$. If the cross section is truly diverging like $1/s$, then a $D(z)$ which is nonvanishing for $z \to 0$ does imply an infinite total cross section, whichever way things are considered. In cases like that, it is necessary to impose

a lower cut on z, based on some physics or detector consideration. Some such cut is anyway needed to keep away from the minimum c.m. energy required for PYTHIA events, see above.

The most difficult cases are those with a very narrow and high peak, such as the Z^0. One could initialize at the energy of maximum cross section and use $D(z)$ as is, but efficiency might turn out to be very low. One might then be tempted to do more complicated transforms of the kind illustrated above. As a rule it is then convenient to work in the variables $\tau_z = z_1 z_2$ and $y_z = (1/2)\ln(z_1/z_2)$, cf. section 7.2.

Clearly, the better the behaviour of the cross section can be modelled in the choice of z_1 and z_2, the better the overall event generation efficiency. Even under the best of circumstances, the efficiency will still be lower than for runs with fix energy. There is also a non-negligible time overhead for using variable energies in the first place, from kinematics reconstruction and (in part) from the phase space selection. One should therefore not use variable energies when not needed, and not use a large range of energies $\sqrt{s'}$ if in the end only a smaller range is of experimental interest.

This facility may be combined with most other aspects of the program. For instance, it is possible to simulate beamstrahlung as above and still include bremsstrahlung with `MSTP(11)=1`. Further, one may multiply the overall event weight of `PARP(173)` with a kinematics-dependent weight given by `PYEVWT`, although it is not recommended (since the chances of making a mistake are also multiplied). However, a few things do *not* work.

- It is not possible to use pile-up events, i.e. you must have `MSTP(131)=0`.
- The possibility of giving in your own cross-section optimization coefficients, option `MSTP(121)=2`, would require more input than with fixed energies, and this option should therefore not be used. You can still use `MSTP(121)=1`, however.
- The multiple interactions scenario with `MSTP(82)`≥ 2 only works approximately for energies different from the initialization one. If the c.m. energy spread is smaller than a factor 2, say, the approximation should be reasonable, but if the spread is larger one may have to subdivide into subruns of different energy bins. The initialization should be made at the largest energy to be encountered — whenever multiple interactions are possible (i.e. for incoming hadrons and resolved photons) this is where the cross sections are largest anyway, and so this is no further constraint. There is no simple possibility to change `PARP(82)` during the course of the run, i.e. an energy-independent $p_{\perp 0}$ must be assumed. The default option `MSTP(82)=1` works fine, i.e. does not suffer from the constraints above. If so desired, $p_{\perp\mathrm{min}}$ =`PARP(81)` can be set differently for each event, as a function of c.m. energy. Initialization should then be done with `PARP(81)` as low as it is ever supposed to become.

9.9 How to Include External Processes

Despite a large repertory of processes in PYTHIA, the number of interesting missing ones clearly is even larger, and with time this discrepancy is likely to increase. There are several reasons why it is not practicable to imagine a PYTHIA which has 'everything'. One is the amount of time it takes to implement a process for the few PYTHIA authors, compared with the rate of new cross section results produced by the rather larger matrix-element calculations community. Another is the length of currently produced matrix-element expressions, which would make the program very bulky. A third argument is that, whereas the phase space of $2 \to 1$ and $2 \to 2$ processes can be set up once and for all according to a reasonably flexible machinery, processes with more final-state particles are less easy to generate. To achieve a reasonable efficiency, it is necessary to tailor the phase-space selection procedure to the dynamics of the given process, and to the desired experimental cuts.

At times, simple solutions may be found. Some processes may be seen just as trivial modifications of already existing ones. For instance, you might want to add some extra term, corresponding to contact interactions, to the matrix elements of a PYTHIA $2 \to 2$ process. In that case it is not necessary to go through the machinery below, but instead you can use the `PYEVWT`

routine (subsection 9.7) to introduce an additional weight for the event, defined as the ratio of the modified to the unmodified differential cross sections. If you use the option `MSTP(142)=2`, this weight is considered as part of the 'true' cross section of the process, and the generation is changed accordingly.

A PYTHIA expert could also consider implementing a new process along the lines of the existing ones, hardwired in the code. Such a modification would have to be ported anytime the PYTHIA program is upgraded, however.

The recommended solution, if a desired process is missing, is instead to include it into PYTHIA as an 'external' process. In this section we will describe how it is possible to specify the parton-level state of some hard-scattering process in a common block. ('Parton-level' is not intended to imply a restriction to quarks and gluons as interacting particles, but only that quarks and gluons are given rather than the hadrons they will produce in the observable final state.) PYTHIA will read this common block, and add initial- and final-state showers, beam remnants and underlying events, fragmentation and decays, to build up an event in as much detail as an ordinary PYTHIA one. Another common block is to be filled with information relevant for the run as a whole, where beams and processes are specified.

Such a facility has been available since long, and has been used e.g. together with the COMPHEP package. COMPHEP [Puk99] is mainly intended for the automatic computation of matrix elements, but also allows the sampling of phase space according to these matrix elements and thereby the generation of weighted or unweighted events. These events can be saved on disk and thereafter read back in to PYTHIA for subsequent consideration [Bel00].

At the Les Houches 2001 workshop it was decided to develop a common standard, that could be used by all matrix-elements-based generators to feed information into any complete event generator [Boo01]. It is similar to, but in its details different from, the approach previously implemented in PYTHIA. Furthermore, it uses the same naming convention: all names in commonblocks end with `UP`, short for User(-defined) Process. This produces some clashes. Therefore the old facility, existing up to and including PYTHIA 6.1, has been completely removed and replaced by the new one. The new code is still under development, and not all particulars have yet been implemented. In the description below we will emphasize current restrictions to the standard, as well as the solutions to aspects not specified by the standard.

In particular, even with the common block contents defined, it is not clear where they are to be filled, i.e. how the external supplier of parton-level events should synchronize with PYTHIA. The solution adopted here — recommended in the standard — is to introduce two subroutines, `UPINIT` and `UPEVNT`. The first is called by `PYINIT` at initialization to obtain information about the run itself, and the other called by `PYEVNT` each time a new event configuration is to be fed in. We begin by describing these two steps and their related common blocks, before proceeding with further details and examples. The description is cast in a PYTHIA-oriented language, but for the common block contents it closely matches the generator-neutral standard in [Boo01]. Restrictions to or extensions of the standard should be easily recognized, but in case you are vitally dependent on following the standard exactly, you should of course check [Boo01].

If you want to provide routines based on this standard, free to be used by a larger community, please inform torbjorn@thep.lu.se. The intention is to create a list of links to such routines, accessible from the standard PYTHIA webpage, if there is interest.

9.9.1 Run information

When `PYINIT` is called in the main program, with `'USER'` as first argument (which makes the other arguments dummy), it signals that external processes are to be implemented. Then `PYINIT`, as part of its initialization tasks, will call the routine `UPINIT`.

```
CALL UPINIT
```

Purpose: routine to be provided by you when you want to implement external processes, wherein the contents of the HEPRUP common block are set. This information specifies the character of the run, both beams and processes, see further below.

Note 1: alternatively, the HEPRUP common block could be filled already before PYINIT is called, in which case UPINIT could be empty. We recommend UPINIT as the logical place to collect the relevant information, however.

Note 2: a dummy copy of UPINIT is distributed with the program, in order to avoid potential problems with unresolved external references. This dummy should not be linked when you supply your own UPINIT routine.

```
 INTEGER MAXPUP
 PARAMETER (MAXPUP=100)
 INTEGER IDBMUP,PDFGUP,PDFSUP,IDWTUP,NPRUP,LPRUP
 DOUBLE PRECISION EBMUP,XSECUP,XERRUP,XMAXUP
 COMMON/HEPRUP/IDBMUP(2),EBMUP(2),PDFGUP(2),PDFSUP(2),
&IDWTUP,NPRUP,XSECUP(MAXPUP),XERRUP(MAXPUP),XMAXUP(MAXPUP),
&LPRUP(MAXPUP)
```

Purpose: to contain the initial information necessary for the subsequent generation of complete events from externally provided parton configurations. The IDBMUP, EBMUP, PDFGUP and PDFSUP variables specify the nature of the two incoming beams. IDWTUP is a master switch, selecting the strategy to be used to mix different processes. NPRUP gives the number of different external processes to mix, and XSECUP, XERRUP, XMAXUP and LPRUP information on each of these. The contents in this common block must remain unchanged by the user during the course of the run, once set in the initialization stage.

This common block should be filled in the UPINIT routine or, alternatively, before the PYINIT call. During the run, PYTHIA may update the XMAXUP values as required.

MAXPUP : the maximum number of distinguishable processes that can be defined. (Each process in itself could consist of several subprocesses that have been distinguished in the parton-level generator, but where this distinction is not carried along.)

IDBMUP : the PDG codes of the two incoming beam particles (or, in alternative terminology, the beam and target particles).

In PYTHIA, this replaces the information normally provided by the BEAM and TARGET arguments of the PYINIT call. Only particles which are acceptable BEAM or TARGET arguments may also be used in IDBMUP. The 'gamma/*lepton*' options are not available.

EBMUP : the energies, in GeV, of the two incoming beam particles. The first (second) particle is taken to travel in the $+z$ ($-z$) direction.

The standard also allows non-collinear and varying-energy beams to be specified, see ISTUP = -9 below, but this is not yet implemented in PYTHIA.

PDFGUP, PDFSUP : the author group (PDFGUP) and set (PDFSUP) of the parton distributions of the two incoming beams, as used in the generation of the parton-level events. Numbers are based on the PDFLIB [Plo93] lists. This enumeration may not always be up to date, but it provides the only unique integer labels for parton distributions that we have. Where no codes are yet assigned to the parton distribution sets used, one should do as best as one can, and be prepared for more extensive user interventions to interpret the information. For lepton beams, or when the information is not provided for other reasons, one should put PDFGUP = PDFSUP = -1.

By knowing which set has been used, it is possible to reweight cross sections event by event, to correspond to another set.

Note that PYTHIA does not access the PDFGUP or PDFSUP values in its descrip-

tion of internal processes or initial-state showers. If you want this to happen, you have to manipulate the MSTP(51) - MSTP(56) switches. For instance, to access PDFLIB for protons, put MSTP(51) = 1000*PDFGUP + PDFSUP and MSTP(52) = 2 in UPINIT. (And remove the dummy PDFLIB routines, as described for MSTP(52).) Also note that PDFGUP and PDFSUP allow an independent choice of parton distributions on the two sides of the event, whereas PYTHIA only allows one single choice for all protons, another for all pions and a third for all photons.

IDWTUP : master switch dictating how event weights and cross sections should be interpreted. Several different models are presented in detail below. There will be tradeoffs between these, e.g. a larger flexibility to mix and re-mix several different processes could require a larger administrative machinery. Therefore the best strategy would vary, depending on the format of the input provided and the output desired. In some cases, parton-level configurations have already been generated with one specific model in mind, and then there may be no choice.

IDWTUP significantly affects the interpretation of XWGTUP, XMAXUP and XSECUP, as described below, but the basic nomenclature is the following. XWGTUP is the event weight for the current parton-level event, stored in the HEPEUP common block. For each allowed external process i, XMAXUP(i) gives the maximum event weight that could be encountered, while XSECUP(i) is the cross section of the process. Here i is an integer in the range between 1 and NPRUP; see the LPRUP description below for comments on alternative process labels.

= 1 : parton-level events come with a weight when input to PYTHIA, but are then accepted or rejected, so that fully generated events at output have a common weight, customarily defined as $+1$. The event weight XWGTUP is a non-negative dimensional quantity, in pb (converted to mb in PYTHIA), with a mean value converging to the total cross section of the respective process. For each process i, the XMAXUP(i) value provides an upper estimate of how large XWGTUP numbers can be encountered. There is no need to supply an XSECUP(i) value; the cross sections printed with PYSTAT(1) are based entirely on the averages of the XWGTUP numbers (with a small correction for the fraction of events that PYEVNT fails to generate in full for some reason).

The strategy is that PYEVNT selects which process i should be generated next, based on the relative size of the XMAXUP(i) values. The UPEVNT routine has to fill the HEPEUP common block with a parton-level event of the requested type, and give its XWGTUP event weight. The event is accepted by PYEVNT with probability XWGTUP/XMAXUP(i). In case of rejection, PYEVNT selects a new process i and asks for a new event. This ensures that processes are mixed in proportion to their average XWGTUP values.

This model presumes that UPEVNT is able to return a parton-level event of the process type requested by PYEVNT. It works well if each process is associated with an input stream of its own, either a subroutine generating events 'on the fly' or a file of already generated events. It works less well if parton-level events from different processes already are mixed in a single file, and therefore cannot easily be returned in the order wanted by PYEVNT. In the latter case one should either use another model or else consider reducing the level of ambition: even if you have mixed several different subprocesses on a file, maybe there is no need for PYTHIA to know this finer classification, in which case we may get back to a situation with one 'process' per external file. Thus the subdivision into processes should be a matter of convenience, not a straight jacket. Specifically, the shower and hadronization treatment of a parton-level event is independent of the process label assigned to it.

If the events of some process are already available unweighted, then a correct mixing of this process with others is ensured by putting XWGTUP =

XMAXUP(i), where both of these numbers now is the total cross section of the process.

Each XMAXUP(i) value must be known from the very beginning, e.g. from an earlier exploratory run. If a larger value is encountered during the course of the run, a warning message will be issued and the XMAXUP(i) value (and its copy in XSEC(ISUB,1)) increased. Events generated before this time will have been incorrectly distributed, both in the process composition and in the phase space of the affected process, so that a bad estimate of XMAXUP(i) may require a new run with a better starting value.

The model described here agrees with the one used for internal PYTHIA processes, and these can therefore freely be mixed with the external ones. Internal processes are switched on with MSUB(ISUB) = 1, as usual, either before the PYINIT call or in the UPINIT routine. One cannot use MSEL to select a predefined set of processes, for technical reasons, wherefore MSEL = 0 is hardcoded when external processes are included.

A reweighting of events is feasible, e.g. by including a kinematics-dependent K factor into XWGTUP, so long as XMAXUP(i) is also properly modified to take this into account. Optionally it is also possible to produce events with non-unit weight, making use the PYEVWT facility, see subsection 9.7. This works exactly the same way as for internal PYTHIA processes, except that the event information available inside PYEVWT would be different for external processes. You may therefore wish to access the HEPEUP common block inside your own copy of PYEVWT, where you calculate the event weight.

In summary, this option provides maximal flexibility, but at the price of potentially requiring the administration of several separate input streams of parton-level events.

= -1 : same as = 1 above, except that event weights may be either positive or negative on input, and therefore can come with an output weight of $+1$ or -1. This weight is uniquely defined by the sign of XWGTUP. It is also stored in PARI(7). The need for negative-weight events arises in some next-to-leading-order calculations, but there are inherent dangers, discussed in subsection 9.9.4 below. In order to allow a correct mixing between processes, a process of indeterminate cross section sign has to be split up in two, where one always gives a positive or vanishing XWGTUP, and the other always gives it negative or vanishing. The XMAXUP(i) value for the latter process should give the most negative XWGTUP that will be encountered. PYEVNT selects which process i that should be generated next, based on the relative size of the |XMAXUP(i)| values. A given event is accepted with probability |XWGTUP|/|XMAXUP(i)|.

= 2 : parton-level events come with a weight when input to PYTHIA, but are then accepted or rejected, so that events at output have a common weight, customarily defined as $+1$. The non-negative event weight XWGTUP and its maximum value XMAXUP(i) may or may not be dimensional quantities; it does not matter since only the ratio XWGTUP/XMAXUP(i) will be used. Instead XSECUP(i) contains the process cross section in pb (converted to mb in PYTHIA). It is this cross section that appears in the PYSTAT(1) table, only modified by the small fraction of events that PYEVNT fails to generate in full for some reason. The strategy is that PYEVNT selects which process i should be generated next, based on the relative size of the XSECUP(i) values. The UPEVNT routine has to fill the HEPEUP common block with a parton-level event of the requested type, and give its XWGTUP event weight. The event is accepted by PYEVNT with probability XWGTUP/XMAXUP(i). In case of rejection, the process number i is retained and PYEVNT asks for a new event of this kind. This ensures that processes are mixed in proportion to their XSECUP(i) values.

This model presumes that UPEVNT is able to return a parton-level event of the process type requested by PYEVNT, with comments exactly as for the = 1 option.

If the events of some process are already available unweighted, then a correct mixing of this process with others is ensured by putting XWGTUP = XMAXUP(i).

Each XMAXUP(i) and XSECUP(i) value must be known from the very beginning, e.g. from an earlier integration run. If a larger value is encountered during the course of the run, a warning message will be issued and the XMAXUP(i) value increased. This will not affect the process composition, but events generated before this time will have been incorrectly distributed in the phase space of the affected process, so that a bad estimate of XMAXUP(i) may require a new run with a better starting value.

While the generation model is different from the normal internal PYTHIA one, it is sufficiently close that internal processes can be freely mixed with the external ones, exactly as described for the = 1 option. In such a mix, internal processes are selected according to their equivalents of XMAXUP(i) and at rejection a new i is selected, whereas external ones are selected according to XSECUP(i) with i retained when an event is rejected.

A reweighting of individual events is no longer simple, since this would change the XSECUP(i) value nontrivially. Thus a new integration run with the modified event weights would be necessary to obtain new XSECUP(i) and XMAXUP(i) values. An overall rescaling of each process separately can be obtained by modifying the XSECUP(i) values accordingly, however, e.g. by a relevant K factor.

In summary, this option is similar to the = 1 one. The input of XSECUP(i) allows good cross section knowledge also in short test runs, but at the price of a reduced flexibility to reweight events.

= -2 : same as = 2 above, except that event weights may be either positive or negative on input, and therefore can come with an output weight of $+1$ or -1. This weight is uniquely defined by the sign of XWGTUP. It is also stored in PARI(7).

The need for negative-weight events arises in some next-to-leading-order calculations, but there are inherent dangers, discussed in subsection 9.9.4 below.

In order to allow a correct mixing between processes, a process of indeterminate cross section sign has to be split up in two, where one always gives a positive or vanishing XWGTUP, and the other always gives it negative or vanishing. The XMAXUP(i) value for the latter process should give the most negative XWGTUP that will be encountered, and XSECUP(i) should give the integrated negative cross section. PYEVNT selects which process i that should be generated next, based on the relative size of the |XSECUP(i)| values. A given event is accepted with probability |XWGTUP|/|XMAXUP(i)|.

= 3 : parton-level events come with unit weight when input to PYTHIA, XWGTUP = 1, and are thus always accepted. This makes the XMAXUP(i) superfluous, while XSECUP(i) should give the cross section of each process.

The strategy is that that the next process type i is selected by the user inside UPEVNT, at the same time as the HEPEUP common block is filled with information about the parton-level event. This event is then unconditionally accepted by PYEVNT, except for the small fraction of events that PYEVNT fails to generate in full for some reason.

This model allows UPEVNT to read events from a file where different processes already appear mixed. Alternatively, you are free to devise and implement your own mixing strategy inside UPEVNT, e.g. to mimic the ones already outlined for PYEVNT in = 1 and = 2 above.

The XSECUP(i) values should be known from the beginning, in order for PYSTAT(1) to produce a sensible cross section table. This is the only place where it matters, however. That is, the processing of events inside PYTHIA is independent of this information.

In this model it is not possible to mix with internal PYTHIA processes, since not enough information is available to perform such a mixing.

A reweighting of events is completely in the hands of the UPEVNT author. In the case that all events are stored in a single file, and all are to be handed on to PYEVNT, only a common K factor applied to all processes would be possible.

In summary, this option puts more power — and responsibility — in the hands of the author of the parton-level generator. It is very convenient for the processing of unweighted parton-level events stored in a single file. The price to be paid is a reduced flexibility in the reweighting of events, or in combining processes at will.

= -3 : same as = 3 above, except that event weights may be either $+1$ or -1. This weight is uniquely defined by the sign of XWGTUP. It is also stored in PARI(7).

The need for negative-weight events arises in some next-to-leading-order calculations, but there are inherent dangers, discussed in subsection 9.9.4 below.

Unlike the = -1 and = -2 options, there is no need to split a process in two, each with a definite XWGTUP sign, since PYEVNT is not responsible for the mixing of processes. It may well be that the parton-level-generator author has enforced such a split, however, to solve a corresponding mixing problem inside UPEVNT. Information on the relative cross section in the negative- and positive-weight regions may also be useful to understand the character and validity of the calculation (large cancellations means trouble!).

= 4 : parton-level events come with a weight when input to PYTHIA, and this weight is to be retained unchanged at output. The event weight XWGTUP is a non-negative dimensional quantity, in pb (converted to mb in PYTHIA, and as such also stored in PARI(7)), with a mean value converging to the total cross section of the respective process. When histogramming results, one of these event weights would have to be used.

The strategy is exactly the same as = 3 above, except that the event weight is carried along from UPEVNT to the PYEVNT output. Thus again all control is in the hands of the UPEVNT author.

A cross section can be calculated from the average of the XWGTUP values, as in the = 1 option, and is displayed by PYSTAT(1). Here it is of purely informative character, however, and does not influence the generation procedure. Neither XSECUP(i) or XMAXUP(i) needs to be known or supplied.

In this model it is not possible to mix with internal PYTHIA processes, since not enough information is available to perform such a mixing.

A reweighting of events is completely in the hands of the UPEVNT author, and is always simple, also when events appear sequentially stored in a single file.

In summary, this option allows maximum flexibility for the parton-level-generator author, but potentially at the price of spending a significant amount of time processing events of very small weight. Then again, in some cases it may be an advantage to have more events in the tails of a distribution in order to understand those tails better.

= -4 : same as = 4 above, except that event weights in XWGTUP may be either positive or negative. In particular, the mean value of XWGTUP is converging to the total cross section of the respective process. The need for negative-weight events arises in some next-to-leading-order calculations, but there are inherent dangers, discussed in subsection 9.9.4 below.

Unlike the = -1 and = -2 options, there is no need to split a process in two,

each with a definite XWGTUP sign, since PYEVNT does not have to mix processes. However, as for option = -3, such a split may offer advantages in understanding the character and validity of the calculation.

NPRUP : the number of different external processes, with information stored in the first NPRUP entries of the XSECUP, XERRUP, XMAXUP and LPRUP arrays.

XSECUP : cross section for each external process, in pb. This information is mandatory for IDWTUP = ± 2, helpful for ± 3, and not used for the other options.

XERRUP : the statistical error on the cross section for each external process, in pb.
PYTHIA will never make use of this information, but if it is available anyway it provides a helpful service to the user of parton-level generators.
Note that, if a small number $n_{\rm acc}$ of events pass the experimental selection cuts, the statistical error on this cross section is limited by $\delta\sigma/\sigma \approx 1/\sqrt{n_{\rm acc}}$, irrespectively of the quality of the original integration. Furthermore, at least in hadronic physics, systematic errors from parton distributions and higher orders typically are much larger than the statistical errors.

XMAXUP : the maximum event weight XWGTUP that is likely to be encountered for each external process. For IDWTUP = ± 1 it has dimensions pb, while the dimensionality need not be specified for ± 2. For the other IDWTUP options it is not used.

LPRUP : a unique integer identifier of each external process, free to be picked by you for your convenience. This code is used in the IDPRUP identifier of which process occured.
In PYTHIA, an external process is thus identified by three different integers. The first is the PYTHIA process number, ISUB. This number is assigned by PYINIT at the beginning of each run, by scanning the ISET array for unused process numbers, and reclaiming such in the order they are found. The second is the sequence number i, running from 1 through NPRUP, used to find information in the cross section arrays. The third is the LPRUP(i) number, which can be anything that the user wants to have as a unique identifier, e.g. in a larger database of processes. For PYTHIA to handle conversions, the two KFPR numbers of a given process ISUB are overwritten with the second and third numbers above. Thus the first external process will land in ISUB = 4 (currently), and could have LPRUP(1) = 13579. In a PYSTAT(1) call, it would be listed as User process 13579.

9.9.2 Event information

Inside the event loop of the main program, PYEVNT will be called to generate the next event, as usual. When this is to be an external process, the parton-level event configuration and the event weight is found by a call from PYEVNT to UPEVNT.

```
CALL UPEVNT
```

Purpose: routine to be provided by you when you want to implement external processes, wherein the contents of the HEPEUP common block are set. This information specifies the next parton-level event, and some additional event information, see further below. How UPEVNT is expected to solve its task depends on the model selected in IDWTUP, see above. Specifically, note that the process type IDPRUP has already been selected for some IDWTUP options (and then cannot be overwritten), while it remains to be chosen for others.

Note : a dummy copy of UPEVNT is distributed with the program, in order to avoid potential problems with unresolved external references. This dummy should not be linked when you supply your own UPEVNT routine.

```
 INTEGER MAXNUP
 PARAMETER (MAXNUP=500)
 INTEGER NUP,IDPRUP,IDUP,ISTUP,MOTHUP,ICOLUP
 DOUBLE PRECISION XWGTUP,SCALUP,AQEDUP,AQCDUP,PUP,VTIMUP,SPINUP
 COMMON/HEPEUP/NUP,IDPRUP,XWGTUP,SCALUP,AQEDUP,AQCDUP,IDUP(MAXNUP),
&ISTUP(MAXNUP),MOTHUP(2,MAXNUP),ICOLUP(2,MAXNUP),PUP(5,MAXNUP),
&VTIMUP(MAXNUP),SPINUP(MAXNUP)
```

Purpose : to contain information on the latest external process generated in `UPEVNT`. A part is one-of-a-kind numbers, like the event weight, but the bulk of the information is a listing of incoming and outgoing particles, with history, colour, momentum, lifetime and spin information.

`MAXNUP` : the maximum number of particles that can be specified by the external process. The maximum of 500 is more than PYTHIA is set up to handle. By default, `MSTP(126) = 100`, at most 96 particles could be specified, since 4 additional entries are needed in PYTHIA for the two beam particles and the two initiators of initial-state radiation. If this default is not sufficient, `MSTP(126)` would have to be increased at the beginning of the run.

`NUP` : the number of particle entries in the current parton-level event, stored in the `NUP` first entries of the `IDUP`, `ISTUP`, `MOTHUP`, `ICOLUP`, `PUP`, `VTIMUP` and `SPINUP` arrays. The special value `NUP = 0` is used to denote the case where `UPEVNT` is unable to provide an event, at least of the type requested by `PYEVNT`, e.g. because all events available in a file have already been read. For such an event also the error flag `MSTI(51) = 1` instead of the normal `= 0`.

`IDPRUP` : the identity of the current process, as given by the `LPRUP` codes. When `IDWTUP =` ± 1 or ± 2, `IDPRUP` is selected by `PYEVNT` and already set when entering `UPEVNT`. Then `UPEVNT` has to provide an event of the specified process type, but cannot change `IDPRUP`. When `IDWTUP =` ± 3 or ± 4, `UPEVNT` is free to select the next process, and then should set `IDPRUP` accordingly.

`XWGTUP` : the event weight. The precise definition of `XWGTUP` depends on the value of the `IDWTUP` master switch. For `IDWTUP = 1` or `= 4` it is a dimensional quantity, in pb, with a mean value converging to the total cross section of the respective process. For `IDWTUP = 2` the overall normalization is irrelevant. For `IDWTUP = 3` only the value `+1` is allowed. For negative `IDWTUP` also negative weights are allowed, although positive and negative weights cannot appear mixed in the same process for `IDWTUP = -1` or `= -2`.

`SCALUP` : scale Q of the event, as used in the calculation of parton distributions (factorization scale). If the scale has not been defined, this should be denoted by using the value `-1`.
In PYTHIA, this is input to `PARI(21) - PARI(26)` (and internally `VINT(51) - VINT(56)`) When `SCALUP` is non-positive, the invariant mass of the parton-level event is instead used as scale. Either of these comes to set the maximum virtuality in the initial-state parton showers. The same scale is also used for the first final-state shower, i.e. the one associated with the hard scattering. As in internal events, `PARP(67)` and `PARP(71)` offer multiplicative factors, whereby the respective initial- or final-state showering Q^2_{max} scale can be modified relative to the scale above. Any subsequent final-state showers are assumed to come from resonance decays, where the resonance mass always sets the scale.

`AQEDUP` : the QED coupling α_{em} used for this event. If α_{em} has not been defined, this should be denoted by using the value `-1`.
In PYTHIA, this value is stored in `VINT(57)`. It is not used anywhere, however.

`AQCDUP` : the QCD coupling α_{s} used for this event. If α_{s} has not been defined, this should be

denoted by using the value `-1`.
In PYTHIA, this value is stored in `VINT(58)`. It is not used anywhere, however.

`IDUP(i)` : particle identity code, according to the PDG convention, for particle `i`. As an extension to this standard, `IDUP(i) = 0` can be used to designate an intermediate state of undefined (and possible non-physical) character, e.g. a subsystem with a mass to be preserved by parton showers.
In the PYTHIA event record, this corresponds to the `KF = K(I,2)` code. But note that, here and in the following, the positions `i` in `HEPEUP` and `I` in `PYJETS` are likely to be different, since PYTHIA normally stores more information in the beginning of the event record. Since `K(I,2) = 0` is forbidden, the `IDUP(i) = 0` code is mapped to `K(I,2) = 90`.

`ISTUP(i)` : status code of particle `i`.

`= -1` : an incoming particle of the hard-scattering process.
In PYTHIA, currently it is presumed that the first two particles, `i = 1` and `= 2`, are of this character, and none of the others. If not, the program execution will stop. This is a restriction relative to the standard, which allows more possibilities. It is also presumed that these two particles are given with vanishing masses and parallel to the respective incoming beam direction, i.e. $E = p_z$ for the first and $E = -p_z$ for the second. The assignment of spacelike virtualities and nonvanishing $p_\perp$'s from initial-state radiation and primordial $k_\perp$'s is the prerogative of PYTHIA.

`= 1` : an outgoing final-state particle.
Such a particle can, of course, be processed further by PYTHIA, to add showers and hadronization, or perform decays of any leftover resonances.

`= 2` : an intermediate resonance, whose mass should be preserved by parton showers. For instance, in a process such as $\mathrm{e^+e^-} \to \mathrm{Z^0h^0} \to \mathrm{q\overline{q}b\overline{b}}$, the $\mathrm{Z^0}$ and $\mathrm{h^0}$ should both be flagged this way, to denote that the $\mathrm{q\overline{q}}$ and $\mathrm{b\overline{b}}$ systems should have their individual masses preserved. In a more complex example, $\mathrm{d\overline{u}} \to \mathrm{W^-Z^0g} \to \ell^-\overline{\nu}_\ell\,\mathrm{q\overline{q}g}$, both the $\mathrm{W^-}$ and $\mathrm{Z^0}$ particles and the $\mathrm{W^-Z^0}$ pseudoparticle (with `IDUP(i) = 0`) could be given with status 2.
Often mass preservation is correlated with colour singlet subsystems, but this need not be the case. In $\mathrm{e^+e^-} \to \mathrm{t\overline{t}} \to \mathrm{bW^+\overline{b}W^-}$, the b and $\overline{\mathrm{b}}$ would be in a colour singlet state, but *not* with a preserved mass. Instead the $\mathrm{t} = \mathrm{bW^+}$ and $\overline{\mathrm{t}} = \overline{\mathrm{b}}\mathrm{W^-}$ masses would be preserved, i.e. when b radiates $\mathrm{b} \to \mathrm{bg}$ the recoil is taken by the $\mathrm{W^+}$. Exact mass preservation also by the hadronization stage is only guaranteed for colour singlet subsystems, however, at least for string fragmentation, since it is not possible to define a subset of hadrons that uniquely belong only with a single coloured particle.
The assignment of intermediate states is not always quantum mechanically well-defined. For instance, $\mathrm{e^+e^-} \to \mu^-\mu^+\nu_\mu\overline{\nu}_\mu$ can proceed both through a $\mathrm{W^+W^-}$ and a $\mathrm{Z^0Z^0}$ intermediate state, as well as through other graphs, which can interfere with each other. It is here the responsibility of the matrix-element-generator author to pick one of the alternatives, according to some convenient recipe. One option might be to perform two calculations, one complete to select the event kinematics and calculate the event weight, and a second with all interference terms neglected to pick the event history according to the relative weight of each graph. Often one particular graph would dominate, because a certain pairing of the final-state fermions would give invariant masses on or close to some resonance peaks.
In PYTHIA, the identification of an intermediate resonance is not only a matter of preserving a mass, but also of improving the modelling of the final-state shower evolution, since matrix-element-correction factors have been calcu-

lated for a variety of possible resonance decays and implemented in the respective parton shower description, see subsection 10.2.6.

`= 3 :` an intermediate resonance, given for documentation only, without any demand that the mass should be preserved in subsequent showers.
In PYTHIA, currently particles defined with this option are not treated any differently from the ones with `= 2`.

`= -2 :` an intermediate space-like propagator, defining an x and a Q^2, in the Deeply Inelastic Scattering terminology, which should be preserved.
In PYTHIA, currently this option is not defined and should not be used. If it is, the program execution will stop.

`= -9 :` an incoming beam particle at time $t = -\infty$. Such beams are not required in most cases, since the `HEPRUP` common block normally contains the information. The exceptions are studies with non-collinear beams and with varying-energy beams (e.g. from beamstrahlung, subsection 7.1.3), where `HEPRUP` does not supply sufficient flexibility. Information given with `= -9` overwrites the one given in `HEPRUP`.
This is an optional part of the standard, since it may be difficult to combine with some of the `IDWTUP` options.
Currently it is not recognized by PYTHIA. If it is used, the program execution will stop.

`MOTHUP(1,i), MOTHUP(2,i) :` position of the first and last mother of particle `i`. Decay products will normally have only one mother. Then either `MOTHUP(2,i) = 0` or `MOTHUP(2,i) = MOTHUP(1,i)`. Particles in the outgoing state of a $2 \to n$ process have two mothers. This scheme does not limit the number of mothers, so long as these appear consecutively in the listing, but in practice there will likely never be more than two mothers per particle.
As has already been mentioned for `ISTUP(i) = 2`, the definition of history is not always unique. Thus, in a case like $e^+e^- \to \mu^+\mu^-\gamma$, proceeding via an intermediate γ^*/Z^0, the squared matrix element contains an interference term between initial- and final-state emission of the photon. This ambiguity has to be resolved by the matrix-elements-based generator.
In PYTHIA, only information on the first mother survives into `K(I,3)`. This is adequate for resonance decays, while particles produced in the primary $2 \to n$ process are given mother code 0, as is customary for internal processes. It implies that two particles are deemed to have the same mothers if the first one agrees; it is difficult to conceive of situations where this would not be the case. Furthermore, it is assumed that the `MOTHUP(1,i) < i`, i.e. that mothers are stored ahead of their daughters, and that all daughters of a mother are listed consecutively, i.e. without other particles interspersed.
PYTHIA has a limit of at most seven particles coming from the same mother, for the final-state parton shower algorithm to work. In fact, the shower is optimized for a primary $2 \to 2$ process followed by some sequence of $1 \to 2$ resonance decays. Then colour coherence with the initial state, matrix-element matching to gluon emission in resonance decays, and other sophisticated features are automatically included. By contrast, the description of emission in systems with three or more partons is less sophisticated. Apart from problems with the algorithm itself, more information would be needed to do a good job than is provided by the standard. Specifically, there is a significant danger of doublecounting or gaps between the radiation already covered by matrix elements and the one added by the shower. The omission from `HEPEUP` of intermediate resonances known to be there, so that e.g. two consecutive $1 \to 2$ decays are bookkept as a single $1 \to 3$ branching, is a simple way to reduce the reliability of your studies!

`ICOLUP(1,i), ICOLUP(2,i) :` integer tags for the colour flow lines passing through the

colour and anticolour, respectively, of the particle. Any particle with colour (anticolour), such as a quark (antiquark) or gluon, will have the first (second) number nonvanishing.

The tags can be viewed as a numbering of different colours in the $N_C \to \infty$ limit of QCD. Any nonzero integer can be used to represent a different colour, but the standard recommends to stay with positive numbers larger than `MAXNUP` to avoid confusion between colour tags and the position labels `i` of particles.

The colour and anticolour of a particle is defined with the respect to the physical time ordering of the process, so as to allow a unique definition of colour flow also through intermediate particles. That is, a quark always has a nonvanishing colour tag `ICOLUP(1,i)`, whether it is in the initial, intermediate or final state. A simple example would be $\mathrm{q}\overline{\mathrm{q}} \to \mathrm{t}\overline{\mathrm{t}} \to \mathrm{bW}^+\overline{\mathrm{b}}\mathrm{W}^-$, where the same colour label is to be used for the q, the t and the b. Correspondingly, the $\overline{\mathrm{q}}$, $\overline{\mathrm{t}}$ and $\overline{\mathrm{b}}$ share another colour label, now stored in the anticolour position `ICOLUP(2,i)`.

The colour label in itself does not distinguish between the colour or the anticolour of a given kind; that information appears in the usage either of the `ICOLUP(1,i)` or of the `ICOLUP(2,i)` position for the colour or anticolour, respectively. Thus, in a $\mathrm{W}^+ \to \mathrm{u}\overline{\mathrm{d}}$ decay, the u and $\overline{\mathrm{d}}$ would share the same colour label, but stored in `ICOLUP(1,i)` for the u and in `ICOLUP(2,i)` for the $\overline{\mathrm{d}}$.

In general, several colour flows are possible in a given subprocess. This leads to ambiguities, of a character similar to the ones for the history above, and as is discussed in subsection 8.2.1. Again it is up to the author of the matrix-elements-based generator to find a sensible solution. It is useful to note that all interference terms between different colour flow topologies vanish in the $N_C \to \infty$ limit of QCD. One solution would then be to use a first calculation in standard QCD to select the momenta and find the weight of the process, and a second with $N_C \to \infty$ to pick one specific colour topology according to the relative probabilities in this limit.

The above colour scheme also allows for baryon number violating processes. Such a vertex would show up by 'dangling' colour lines, when the `ICOLUP` and `MOTHUP` information is correlated. For instance, in $\tilde{\mathrm{u}} \to \overline{\mathrm{d}}\overline{\mathrm{d}}$ the $\tilde{\mathrm{u}}$ inherits an existing colour label, while the two $\overline{\mathrm{d}}$'s are produced with two different new labels.

Several examples of colour assignments, both with and without baryon number violation, are given in [Boo01].

In PYTHIA, baryon number violation is not yet implemented. It will require substantial extra work, and is not imminent.

`PUP(1,i), PUP(2,i), PUP(3,i), PUP(4,i), PUP(5,i) :` the particle momentum vector (p_x, p_y, p_z, E, m), with units of GeV. A spacelike virtuality is denoted by a negative sign on the mass.

Apart from the index order, this exactly matches the `P` momentum conventions in `PYJETS`.

PYTHIA is forgiving when it comes to using other masses than its own, e.g. for quarks. Thus the external process can be given directly with the m_b used in the calculation, without any worry how this matches the PYTHIA default. However, remember that the two incoming particles with `ISTUP(i) = -1` have to be massless.

`VTIMUP(i) :` invariant lifetime $c\tau$ in mm, i.e. distance from production to decay. Once the primary vertex has been selected, the subsequent decay vertex positions in space and time can be constructed step by step, by also making use of the momentum information. Propagation in vacuum, without any bending e.g. by magnetic fields, has to be assumed.

This exactly corresponds to the `V(I,5)` component in `PYJETS`. Note that it is used in PYTHIA to track colour singlet particles through distances that might be observable in a detector. It is not used to trace the motion of coloured partons at fm scales,

through the hadronization process. Also note that PYTHIA will only use this information for intermediate resonances, not for the initial- and final-state particles. For instance, for an undecayed τ^-, the lifetime is selected as part of the τ^- decay process, not based on the `VTIMUP(i)` value.

`SPINUP(i)` : cosine of the angle between the spin vector of a particle and its three-momentum, specified in the lab frame, i.e. the frame where the event as a whole is defined. This scheme is neither general nor complete, but it is chosen as a sensible compromise.
The main foreseen application is τ's with a specific helicity. Typically a relativistic τ^- (τ^+) coming from a W^- (W^+) decay would have helicity and `SPINUP(i)` $= -1$ ($+1$). This could be changed by the boost from the W rest frame to the labe frame, however. The use of a real number, rather than an integer, allows for an extension to the non-relativistic case.
Particles which are unpolarized or have unknown polarization should be given `SPINUP(i) = 9`.
Explicit spin information is not used anywhere in PYTHIA. It is implicit in many production and decay matrix elements, which often contain more correlation information than could be conveyed by the simple spin numbers discussed here. Correspondingly, it is to be expected that the external generator already performed the decays of the W's, the Z's and the other resonances, so as to include the full spin correlations. If this is not the case, such resonances will normally be decayed isotropically. Some correlations could appear in decay chains: the PYTHIA decay $t \to bW^+$ is isotropic, but the subequent $W^+ \to q_1\overline{q}_2$ decay contains implicit W helicity information from the t decay.
Also τ decays performed by PYTHIA would be isotropic. An interface routine `PYTAUD` (see subsection 14.2) can be used to link to external τ decay generators, but is based on defining the τ in the rest frame of the decay that produces it, and so is not directly applicable here. In some future, it will be rewritten to make use of the `SPINUP(i)` information. In the meantime, and of course also afterwards, a valid option is to perform the τ decays yourself before passing 'parton-level' events to PYTHIA.

9.9.3 An example

To exemplify the above discussion, consider the explicit case of $q\overline{q}$ or $gg \to t\overline{t} \to bW^+\overline{b}W^- \to bq_1\overline{q}_2\,\overline{b}q_3\overline{q}_4$. These two processes are already available in PYTHIA, but without full spin correlations. One might therefore wish to include them from some external generator. A physics analysis would then most likely involve angular correlations intended to set limits on (or reveal traces of) anomalous couplings. However, so as to give a simple self-contained example, instead consider the analysis of the charged multiplicity distribution. This actually offers a simple first cross-check between the internal and external implementations of the same process. The main program might then look something like

```
      IMPLICIT DOUBLE PRECISION(A-H, O-Z)
      IMPLICIT INTEGER(I-N)

C...User process event common block.
      INTEGER MAXNUP
      PARAMETER (MAXNUP=500)
      INTEGER NUP,IDPRUP,IDUP,ISTUP,MOTHUP,ICOLUP
      DOUBLE PRECISION XWGTUP,SCALUP,AQEDUP,AQCDUP,PUP,VTIMUP,SPINUP
      COMMON/HEPEUP/NUP,IDPRUP,XWGTUP,SCALUP,AQEDUP,AQCDUP,IDUP(MAXNUP),
     &ISTUP(MAXNUP),MOTHUP(2,MAXNUP),ICOLUP(2,MAXNUP),PUP(5,MAXNUP),
     &VTIMUP(MAXNUP),SPINUP(MAXNUP)
      SAVE /HEPEUP/
```

```
C...PYTHIA common block.
      COMMON/PYJETS/N,NPAD,K(4000,5),P(4000,5),V(4000,5)
      SAVE /PYJETS/

C...Initialize.
      CALL PYINIT('USER',' ',' ',0D0)

C...Book histogram. Reset event counter.
      CALL PYBOOK(1,'Charged multiplicity',100,-1D0,199D0)
      NACC=0

C...Event loop; check that not at end of run; list first events.
      DO 100 IEV=1,1000
        CALL PYEVNT
        IF(NUP.EQ.0) GOTO 110
        NACC=NACC+1
        IF(IEV.LE.3) CALL PYLIST(7)
        IF(IEV.LE.3) CALL PYLIST(2)

C...Analyze event; end event loop.
        CALL PYEDIT(3)
        CALL PYFILL(1,DBLE(N),1D0)
  100 CONTINUE

C...Statistics and histograms.
  110 CALL PYSTAT(1)
      CALL PYFACT(1,1D0/DBLE(NACC))
      CALL PYHIST

      END
```

There `PYINIT` is called with `'USER'` as first argument, implying that the rest is dummy. The event loop itself looks fairly familiar, but with two additions. One is that `NUP` is checked after each event, since `NUP = 0` would signal a premature end of the run, with the external generator unable to return more events. This would be the case e.g. if events have been stored on file, and the end of this file is reached. The other is that `CALL PYLIST(7)` can be used to list the particle content of the `HEPEUP` common block (with some information omitted, on vertices and spin), so that one can easily compare this input with the output after PYTHIA processing, `CALL PYLIST(2)`. An example of a `PYLIST(7)` listing would be

```
          Event listing of user process at input (simplified)

 I IST  ID Mothers   Colours    p_x      p_y      p_z       E        m
 1 -1   21   0   0  101  109    0.000    0.000  269.223  269.223    0.000
 2 -1   21   0   0  109  102    0.000    0.000 -225.566  225.566    0.000
 3  2    6   1   2  101    0   72.569  153.924  -10.554  244.347  175.030
 4  2   -6   1   2    0  102  -72.569 -153.924   54.211  250.441  175.565
 5  1    5   3   0  101    0   56.519   33.343   53.910   85.045    4.500
 6  2   24   3   0    0    0   16.050  120.581  -64.464  159.302   80.150
 7  1   -5   4   0    0  102   44.127  -60.882   25.507   79.527    4.500
 8  2  -24   4   0    0    0 -116.696  -93.042   28.705  170.914   78.184
 9  1    2   6   0  103    0   -8.667   11.859   16.063   21.766    0.000
10  1   -1   6   0    0  103   24.717  108.722  -80.527  137.536    0.000
```

```
11  1   -2   8   0    0  104  -33.709  -22.471  -26.877   48.617    0.000
12  1    1   8   0  104    0  -82.988  -70.571   55.582  122.297    0.000
```

Note the reverse listing of ID(UP) and IST(UP) relative to the HEPEUP order, to have better agreement with the PYJETS one. (The ID column is wider in real life, to allow for longer codes, but has here been reduced to fit the listing onto the page.)

The corresponding PYLIST(2) listing of course would be considerably longer, containing a complete event as it does. Also the particles above would there appear boosted by the effects of initial-state radiation and primordial $k_\perp$; copies of them further down in the event record would also include the effects of final-state radiation. The full story is available with MSTP(125)=2, while the default listing omits some of the intermediate steps.

The PYINIT call will generate a call to the user-supplied routine UPINIT. It is here that we need to specify the details of the generation model. Assume, for instance that $q\bar{q}$- and gg-initiated events have been generated in two separate runs for Tevatron Run II, with weighted events stored in two separate files. By the end of each run, cross section and maximum weight information has also been obtained, and stored on separate files. Then UPINIT could look like

```
      SUBROUTINE UPINIT

C...Double precision and integer declarations.
      IMPLICIT DOUBLE PRECISION(A-H, O-Z)
      IMPLICIT INTEGER(I-N)

C...User process initialization common block.
      INTEGER MAXPUP
      PARAMETER (MAXPUP=100)
      INTEGER IDBMUP,PDFGUP,PDFSUP,IDWTUP,NPRUP,LPRUP
      DOUBLE PRECISION EBMUP,XSECUP,XERRUP,XMAXUP
      COMMON/HEPRUP/IDBMUP(2),EBMUP(2),PDFGUP(2),PDFSUP(2),
     &IDWTUP,NPRUP,XSECUP(MAXPUP),XERRUP(MAXPUP),XMAXUP(MAXPUP),
     &LPRUP(MAXPUP)
      SAVE /HEPRUP/

C....Pythia common block - needed for setting PDF's; see below.
      COMMON/PYPARS/MSTP(200),PARP(200),MSTI(200),PARI(200)
      SAVE /PYPARS/

C...Set incoming beams: Tevatron Run II.
      IDBMUP(1)=2212
      IDBMUP(2)=-2212
      EBMUP(1)=1000D0
      EBMUP(2)=1000D0

C...Set PDF's of incoming beams: CTEQ 5L.
C...Note that Pythia will not look at PDFGUP and PDFSUP.
      PDFGUP(1)=4
      PDFSUP(1)=46
      PDFGUP(2)=PDFGUP(1)
      PDFSUP(2)=PDFSUP(1)

C...Set use of CTEQ 5L in internal Pythia code.
      MSTP(51)=7

C...If you want Pythia to use PDFLIB, you have to set it by hand.
```

```
C...(You also have to ensure that the dummy routines
C...PDFSET, STRUCTM and STRUCTP in Pythia are not linked.)
C      MSTP(52)=2
C      MSTP(51)=1000*PDFGUP(1)+PDFSUP(1)

C...Decide on weighting strategy: weighted on input, cross section known.
      IDWTUP=2

C...Number of external processes.
      NPRUP=2

C...Set up q qbar -> t tbar.
      OPEN(21,FILE='qqtt.info',FORM='unformatted',ERR=100)
      READ(21,ERR=100) XSECUP(1),XERRUP(1),XMAXUP(1)
      LPRUP(1)=661
      OPEN(22,FILE='qqtt.events',FORM='unformatted',ERR=100)

C...Set up g g -> t tbar.
      OPEN(23,FILE='ggtt.info',FORM='unformatted',ERR=100)
      READ(23,ERR=100) XSECUP(2),XERRUP(2),XMAXUP(2)
      LPRUP(2)=662
      OPEN(24,FILE='ggtt.events',FORM='unformatted',ERR=100)

      RETURN
C...Stop run if file operations fail.
  100 WRITE(*,*) 'Error! File open or read failed. Program stopped.'
      STOP

      END
```

Here unformatted read/write is used to reduce the size of the event files, but at the price of a platform dependence. Formatted files are preferred if they are to be shipped elsewhere. The rest should be self-explanatory.

Inside the event loop of the main program, `PYEVNT` will call `UPEVNT` to obtain the next parton-level event. In its simplest form, only a single `READ` statement would be necessary to read information on the next event, e.g. what is shown in the event listing earlier in this subsection, with a few additions. Then the routine could look like

```
      SUBROUTINE UPEVNT

C...Double precision and integer declarations.
      IMPLICIT DOUBLE PRECISION(A-H, O-Z)
      IMPLICIT INTEGER(I-N)

C...User process event common block.
      INTEGER MAXNUP
      PARAMETER (MAXNUP=500)
      INTEGER NUP,IDPRUP,IDUP,ISTUP,MOTHUP,ICOLUP
      DOUBLE PRECISION XWGTUP,SCALUP,AQEDUP,AQCDUP,PUP,VTIMUP,SPINUP
      COMMON/HEPEUP/NUP,IDPRUP,XWGTUP,SCALUP,AQEDUP,AQCDUP,IDUP(MAXNUP),
     &ISTUP(MAXNUP),MOTHUP(2,MAXNUP),ICOLUP(2,MAXNUP),PUP(5,MAXNUP),
     &VTIMUP(MAXNUP),SPINUP(MAXNUP)
      SAVE /HEPEUP/
```

```
C...Pick file to read from, based on requested event type.
      IUNIT=22
      IF(IDPRUP.EQ.662) IUNIT=24

C...Read event from this file. (Except that NUP and IDPRUP are known.)
      NUP=12
      READ(IUNIT,ERR=100,END=100) XWGTUP,SCALUP,AQEDUP,AQCDUP,
     &(IDUP(I),ISTUP(I),MOTHUP(1,I),MOTHUP(2,I),ICOLUP(1,I),
     &ICOLUP(2,I),(PUP(J,I),J=1,5),VTIMUP(I),SPINUP(I),I=1,NUP)

C...Return, with NUP=0 if read failed.
      RETURN
  100 NUP=0
      RETURN
      END
```

However, in reality one might wish to save disk space by not storing redundant information. The XWGTUP and SCALUP numbers are vital, while AQEDUP and AQCDUP are purely informational and can be omitted. In a $gg \to t\bar{t} \to bW^+\bar{b}W^- \to bq_1\bar{q}_2\,\bar{b}q_3\bar{q}_4$ event, only the q_1, $\bar{q}_2$, q_3 and $\bar{q}_4$ flavours need be given, assuming that the particles are always stored in the same order. For a $q\bar{q}$ initial state, the q flavour should be added to the list. The ISTUP, MOTHUP and ICOLUP information is the same in all events of a given process, except for a twofold ambiguity in the colour flow for gg initial states. All VTIMUP vanish and the SPINUP are uninteresting since spins have already been taken into account by the kinematics of the final fermions. (It would be different if one of the W's decayed leptonically to a τ.) Only the PUP values of the six final fermions need be given, since the other momenta and masses can be reconstructed from this, remembering that the two initial partons are massless and along the beam pipe. The final fermions are on the mass shell, so their masses need not be stored event by event, but can be read from a small table. The energy of a particle can be reconstructed from its momentum and mass. Overall transverse momentum conservation removes two further numbers. What remains is thus 5 integers and 18 real numbers, where the reals could well be stored in single precision. Of course, the code needed to unpack information stored this way would be lengthy but trivial. Even more compact storage strategies could be envisaged, e.g. only to save the weight and the seed of a dedicated random-number generator, to be used to generate the next parton-level event. It is up to you to find the optimal balance between disk space and coding effort.

9.9.4 Further comments

This section contains additional information on a few different topics: cross section interpretation, negative-weight events, relations with other PYTHIA switches and routines, and error conditions.

In several IDWTUP options, the XWGTUP variable is supposed to give the differential cross section of the current event, times the phase-space volume within which events are generated, expressed in picobarns. (Converted to millibarns inside PYTHIA.) This means that, in the limit that many events are generated, the average value of XWGTUP gives the total cross section of the simulated process. Of course, the tricky part is that the differential cross section usually is strongly peaked in a few regions of the phase space, such that the average probability to accept an event, $\langle$XWGTUP$\rangle$/XMAXUP(i) is small. It may then be necessary to find a suitable set of transformed phase-space coordinates, for which the correspondingly transformed differential cross section is better behaved.

To avoid unclarities, here is a more formal version of the above paragraph. Call $\mathrm{d}X$ the differential phase space, e.g. for a $2 \to 2$ process $\mathrm{d}X = \mathrm{d}x_1\,\mathrm{d}x_2\,\mathrm{d}\hat{t}$, where x_1 and x_2 are the momentum fractions carried by the two incoming partons and $\hat{t}$ the Mandelstam variable of the

scattering (see subsection 7.2). Call $\mathrm{d}\sigma/\mathrm{d}X$ the differential cross section of the process, e.g. for $2 \to 2$: $\mathrm{d}\sigma/\mathrm{d}X = \sum_{ij} f_i(x_1, Q^2)\, f_j(x_2, Q^2)\, \mathrm{d}\hat{\sigma}_{ij}/\mathrm{d}\hat{t}$, i.e. the product of parton distributions and hard-scattering matrix elements, summed over all allowed incoming flavours i and j. The physical cross section that one then wants to generate is $\sigma = \int (\mathrm{d}\sigma/\mathrm{d}X)\, \mathrm{d}X$, where the integral is over the allowed phase-space volume. The event generation procedure consists of selecting an X uniformly in $\mathrm{d}X$ and then evaluating the weight $\mathrm{d}\sigma/\mathrm{d}X$ at this point. `XWGTUP` is now simply `XWGTUP`$= \mathrm{d}\sigma/\mathrm{d}X \int \mathrm{d}X$, i.e. the differential cross section times the considered volume of phase space. Clearly, when averaged over many events, `XWGTUP` will correctly estimate the desired cross section. If `XWGTUP` fluctuates too much, one may try to transform to new variables X', where events are now picked accordingly to $\mathrm{d}X'$ and `XWGTUP`$= \mathrm{d}\sigma/\mathrm{d}X' \int \mathrm{d}X'$.

A warning. It is important that X is indeed uniformly picked within the allowed phase space, alternatively that any Jacobians are properly taken into account. For instance, in the case above, one approach would be to pick x_1, x_2 and $\hat{t}$ uniformly in the ranges $0 < x_1 < 1$, $0 < x_2 < 1$, and $-s < \hat{t} < 0$, with full phase space volume $\int \mathrm{d}X = s$. The cross section would only be non-vanishing inside the physical region given by $-s x_1 x_2 < \hat{t}$ (in the massless case), i.e. Monte Carlo efficiency is likely to be low. However, if one were to choose $\hat{t}$ values only in the range $-\hat{s} < \hat{t} < 0$, small $\hat{s}$ values would be favoured, since the density of selected $\hat{t}$ values would be larger there. Without the use of a compensating Jacobian $\hat{s}/s$, an incorrect answer would be obtained. Alternatively, one could start out with a phase space like $\mathrm{d}X = \mathrm{d}x_1\, \mathrm{d}x_2\, \mathrm{d}(\cos\hat{\theta})$, where the limits decouple. Of course, the $\cos\hat{\theta}$ variable can be translated back into a $\hat{t}$, which will then always be in the desired range $-\hat{s} < \hat{t} < 0$. The transformation itself here gives the necessary Jacobian.

At times, it is convenient to split a process into a discrete set of subprocesses for the parton-level generation, without retaining these in the `IDPRUP` classification. For instance, the cross section above contains a summation over incoming partons. An alternative would then have been to let each subprocess correspond to one unique combination of incoming flavours. When an event of process type i is to be generated, first a specific subprocess ik is selected with probability f^{ik}, where $\sum_k f^{ik} = 1$. For this subprocess an `XWGTUP`^{k} is generated as above, except that there is no longer a summation over incoming flavours. Since only a fraction f^{ik} of all events now contain this part of the cross section, a compensating factor $1/f^{ik}$ is introduced, i.e. `XWGTUP`=`XWGTUP`$^k/f^{ik}$. Further, one has to define `XMAXUP(i)`$= \max_k$ `XMAXUP`$^{ik}/f^{ik}$ and `XSECUP(i)`$= \sum_k$ `XSECUP`^{ik}. The generation efficiency will be maximized for the f^{ik} coefficients selected proportional to `XMAXUP`^{ik}, but this is no requirement.

The standard allows external parton-level events to come with negative weights, unlike the case for internal PYTHIA processes. In order to avoid indiscriminate use of this option, some words of caution are in place. In next-to-leading-order calculations, events with negative weights may occur as part of the virtual corrections. In any physical observable quantity, the effect of such events should cancel against the effect of real events with one more parton in the final state. For instance, the next-to-leading order calculation of gluon scattering contains the real process $\mathrm{gg} \to \mathrm{ggg}$, with a positive divergence in the soft and collinear regions, while the virtual corrections to $\mathrm{gg} \to \mathrm{gg}$ are negatively divergent. Neglecting the problems of outgoing gluons collinear with the beams, and those of soft gluons, two nearby outgoing gluons in the $\mathrm{gg} \to \mathrm{ggg}$ process can be combined into one effective one, such that the divergences can be cancelled.

If rather widely separated gluons can be combined, the remaining negative contributions are not particularly large. Different separation criteria could be considered; one example would be $\Delta R = \sqrt{(\Delta\eta)^2 + (\Delta\varphi)^2} \approx 1$. The recombination of well separated partons is at the price of an arbitrariness in the choice of clustering algorithm, when two gluons of nonvanishing invariant mass are to be combined into one massless one, as required to be able to compare with the kinematics of the massless $\mathrm{gg} \to \mathrm{gg}$ process when performing the divergence cancellations. Alternatively, if a smaller ΔR cut is used, where the combining procedure is less critical, there will be more events with large positive and negative weights that are to cancel.

Without proper care, this cancellation could easily be destroyed by the subsequent showering description, as follows. The standard for external processes does not provide any way to pass information on the clustering algorithm used, so the showering routine will have to make its own choice what region of phase space to populate with radiation. One choice could be to allow a cone defined by the nearest colour-connected parton (see subsection 10.1.3 for a discussion). There could then arise a significant mismatch in shower description between events where two gluons are just below or just above the ΔR cut for being recombined, equivalently between $gg \to gg$ and $gg \to ggg$ events. Most of the phase space may be open for the former, while only the region below ΔR may be it for the latter. Thus the average 'two-parton' events may end up containing significantly more jet activity than the corresponding 'three-parton' ones. The smaller the ΔR cut, the more severe the mismatch, both on an event-by-event basis and in terms of the event rates involved.

One solution would be to extend the standard also to specify which clustering algorithm has been used in the matrix-element calculation, and with what parameter values. Any shower emission that would give rise to an extra jet, according to this algorithm, would be vetoed. If a small ΔR cut is used, this is almost equivalent to allowing no shower activity at all. (That would still leave us with potential mismatch problems in the hadronization description. Fortunately the string fragmentation approach is very powerful in this respect, with a smooth transition between two almost parallel gluons and a single one with the full energy [Sjö84].) But we know that the unassisted matrix-element description cannot do a good job of the internal structure of jets on an event-by-event basis, since multiple-gluon emission is the norm in this region. Therefore a $\Delta R \sim 1$ will be required, to let the matrix elements describe the wide-angle emission and the showers the small-angle one. This again suggests a picture with only a small contribution from negative-weight events. In summary, the appearance of a large fraction of negative-weight events should be a sure warning sign that physics is likely to be incorrectly described.

The above example illustrates that it may, at times, be desirable to sidestep the standard and provide further information directly in the PYTHIA common blocks. (Currently there is no exact match to the clustering task mentioned above, but there are a few simpler ways to intervene in the shower evolution.) Then it is useful to note that, apart from the hard-process generation machinery itself, the external processes are handled almost exactly as the internal ones. Thus essentially all switches and parameter values related to showers, underlying events and hadronization can be modified at will. This even applies to alternative listing modes of events and history pointers, as set by `MSTP(128)`. Also some of the information on the hard scattering is available, such as `MSTI(3)`, `MSTI(21) - MSTI(26)`, and `PARI(33) - PARI(38)`. Before using them, however, it is prudent to check that your variables of interest do work as intended for the particular process you study. Several differences do remain between internal and external processes, in particular related to final-state showers and resonance decays. For internal processes, the `PYRESD` routine will perform a shower (if relevant) directly after each decay. A typical example would be that a $t \to bW$ decay is immediately followed by a shower, which could change the momentum of the W before it decays in its turn. For an external process, this decay chain would presumably already have been carried out. When the equivalent shower to the above is performed, it is therefore now necessary also to boost the decay products of the W. The special sequence of showers and boosts for external processes is administrated by the `PYADSH` routine.

You are free to make use of whatever tools you want in your `UPINIT` and `UPEVNT` routines, and normally there would be little or no contact with the rest of PYTHIA, except as described above. However, several PYTHIA tools can be used, if you so wish. One attractive possibility is to use `PYPDFU` for parton-distribution-function evaluation. Other possible tools could be `PYR` for random-number generation, `PYALPS` for α_s evaluation, `PYALEM` for evaluation of a running α_{em}, and maybe a few more.

We end with a few comments on anomalous situations. As already described, you may put `NUP=0` inside `UPEVNT`, e.g. to signal the end of the file from which events are read. If the program encounters this value at a return from `UPEVNT`, then it will also exit from `PYEVNT`,

without incrementing the counters for the number of events generated. It is then up to you to have a check on this condition in your main event-generation loop. This you do either by looking at NUP or at MSTI(51); the latter is set to 1 if no event was generated.

It may also happen that a parton-level configuration fails elsewhere in the PYEVNT call. For instance, the beam-remnant treatment occasionally encounters situations it cannot handle, wherefore the parton-level event is rejected and a new one generated. This happens also with ordinary (not user-defined) events, and usually comes about as a consequence of the initial-state radiation description leaving too little energy for the remnant. If the same hard scattering were to be used as input for a new initial-state radiation and beam-remnant attempt, it could then work fine. There is a possibility to give events that chance, as follows. MSTI(52) counts the number of times a hard-scattering configuration has failed to date. If you come in to UPEVNT with MSTI(52) non-vanishing, this means that the latest configuration failed. So long as the contents of the HEPEUP common block are not changed, such an event may be given another try. For instance, a line

```
IF(MSTI(52).GE.1.AND.MSTI(52).LE.4) RETURN
```

at the beginning of UPEVNT will give each event up to five tries; thereafter a new one would be generated as usual. Note that the counter for the number of events is updated at each new try. The fraction of failed configurations is given in the bottom line of the PYSTAT(1) table.

The above comment only refers to very rare occurrences (less than one in a hundred), which are not errors in a strict sense; for instance, they do not produce any error messages on output. If you get warnings and error messages that the program does not understand the flavour codes or cannot reconstruct the colour flows, it is due to faults of yours, and giving such events more tries is not going to help.

9.10 Interfaces to Other Generators

In the previous section an approach to including external processes in PYTHIA was explained. While general enough, it may not always be the optimal choice. In particular, for e^+e^- annihilation events one may envisage some standard cases where simpler approaches could be pursued. A few such standard interfaces are described in this section.

In e^+e^- annihilation events, a convenient classification of electroweak physics is by the number of fermions in the final state. Two fermions from Z^0 decay is LEP1 physics, four fermions can come e.g. from W^+W^- or Z^0Z^0 events at LEP2, and at higher energies six fermions are produced by three-gauge-boson production or top-antitop. Often interference terms are non-negligible, requiring much more complex matrix-element expressions than are normally provided in PYTHIA. Dedicated electroweak generators often exist, however, and the task is therefore to interface them to the generic parton showering and hadronization machinery available in PYTHIA. In the LEP2 workshop [Kno96] one possible strategy was outlined to allow reasonably standardized interfaces between the electroweak and the QCD generators. The LU4FRM routine was provided for the key four-fermion case. This routine is now included here, in slightly modified form, together with two new siblings for two and six fermions. The former is trivial and included mainly for completeness, while the latter is rather more delicate.

In final states with two or three quark–antiquark pairs, the colour connection is not unique. For instance, a $u\bar{d}\bar{u}d$ final state could either stem from a W^+W^- or a Z^0Z^0 intermediate state, or even from interference terms between the two. In order to shower and fragment the system, it is then necessary to pick one of the two alternatives, e.g. according to the relative matrix element weight of each alternative, with the interference term dropped. Some different such strategies are proposed as options below.

Note that here we discuss purely perturbative ambiguities. One can imagine colour reconnection at later stages of the process, e.g. if the intermediate state indeed is W^+W^-, a soft-gluon

exchange could still result in colour singlets $u\bar{u}$ and $d\bar{d}$. We are then no longer speaking of ambiguities related to the hard process itself but rather to the possibility of nonperturbative effects. This is an interesting topic in itself, addressed in section 12.4.2 but not here.

The fermion-pair routines are not set up to handle QCD four-jet events, i.e. events of the types $q\bar{q}gg$ and $q\bar{q}q'\bar{q}'$ (with $q'\bar{q}'$ coming from a gluon branching). Such events are generated in normal parton showers, but not necessarily at the right rate (a problem that may be especially interesting for massive quarks like b). Therefore one would like to start a QCD final-state parton shower from a given four-parton configuration. Already some time ago, a machinery was developed to handle this kind of occurrences [And98a]. This approach has now been adapted to PYTHIA, in a somewhat modified form, see section 10.2.7. The main change is that, in the original work, the colour flow was picked in a separate first step (not discussed in the publication, since it is part of the standard 4-parton configuration machinery of PYEEVT), which reduces the number of allowed $q\bar{q}gg$ parton-shower histories. In the current implementation, more geared towards completely external generators, no colour flow assumptions are made, meaning a few more possible shower histories to pick between. Another change is that mass effects are better respected by the z definition. The code contains one new user routine, PY4JET, two new auxiliary ones, PY4JTW and PY4JTS, and significant additions to the PYSHOW showering routine.

```
CALL PY2FRM(IRAD,ITAU,ICOM)
```

Purpose: to allow a parton shower to develop and partons to hadronize from a two-fermion starting point. The initial list is supposed to be ordered such that the fermion precedes the antifermion. In addition, an arbitrary number of photons may be included, e.g. from initial-state radiation; these will not be affected by the operation and can be put anywhere. The scale for QCD (and QED) final-state radiation is automatically set to be the mass of the fermion-antifermion pair. (It is thus not suited for Bhabha scattering.)

IRAD : final-state QED radiation.
- = 0 : no final-state photon radiation, only QCD showers.
- = 1 : photon radiation inside each final fermion pair, also leptons, in addition to the QCD one for quarks.

ITAU : handling of τ lepton decay (where PYTHIA does not include spin effects, although some generators provide the helicity information that would allow a more sophisticated modelling).
- = 0 : τ's are considered stable (and can therefore be decayed afterwards).
- = 1 : τ's are allowed to decay.

ICOM : place where information about the event (flavours, momenta etc.) is stored at input and output.
- = 0 : in the HEPEVT commonblock (meaning that information is automatically translated to PYJETS before treatment and back afterwards).
- = 1 : in the PYJETS commonblock. All fermions and photons can be given with status code K(I,1)=1, flavour code in K(I,2) and five-momentum (momentum, energy, mass) in P(I,J). The V vector and remaining components in the K one are best put to zero. Also remember to set the total number of entries N.

```
CALL PY4FRM(ATOTSQ,A1SQ,A2SQ,ISTRAT,IRAD,ITAU,ICOM)
```

Purpose: to allow a parton shower to develop and partons to hadronize from a four-fermion starting point. The initial list of fermions is supposed to be ordered in the sequence fermion (1) – antifermion (2) – fermion (3) – antifermion (4). The flavour pairs should be arranged so that, if possible, the first two could come from a W^+ and the second two from a W^-; else each pair should have flavours consistent with a Z^0.

In addition, an arbitrary number of photons may be included, e.g. from initial-state radiation; these will not be affected by the operation and can be put anywhere. Since the colour flow need not be unique, three real and one integer numbers are providing further input. Once the colour pairing is determined, the scale for final-state QCD (and QED) radiation is automatically set to be the mass of the fermion-antifermion pair. (This is the relevant choice for normal fermion pair production from resonance decay, but is not suited e.g. for $\gamma\gamma$ processes dominated by small-t propagators.) The pairing is also meaningful for QED radiation, in the sense that a four-lepton final state is subdivided into two radiating subsystems in the same way. Only if the event consists of one lepton pair and one quark pair is the information superfluous.

`ATOTSQ` : total squared amplitude for the event, irrespective of colour flow.

`A1SQ` : squared amplitude for the configuration with fermions $1+2$ and $3+4$ as the two colour singlets.

`A2SQ` : squared amplitude for the configuration with fermions $1+4$ and $3+2$ as the two colour singlets.

`ISTRAT` : the choice of strategy to select either of the two possible colour configurations. Here 0 is supposed to represent a reasonable compromize, while 1 and 2 are selected so as to give the largest reasonable spread one could imagine.

- `= 0` : pick configurations according to relative probabilities `A1SQ` : `A2SQ`.
- `= 1` : assign the interference contribution to maximize the $1+2$ and $3+4$ pairing of fermions.
- `= 2` : assign the interference contribution to maximize the $1+4$ and $3+2$ pairing of fermions.

`IRAD` : final-state QED radiation.

- `= 0` : no final-state photon radiation, only QCD showers.
- `= 1` : photon radiation inside each final fermion pair, also leptons, in addition to the QCD one for quarks.

`ITAU` : handling of τ lepton decay (where PYTHIA does not include spin effects, although some generators provide the helicity information that would allow a more sophisticated modelling).

- `= 0` : τ's are considered stable (and can therefore be decayed afterwards).
- `= 1` : τ's are allowed to decay.

`ICOM` : place where information about the event (flavours, momenta etc.) is stored at input and output.

- `= 0` : in the `HEPEVT` commonblock (meaning that information is automatically translated to `PYJETS` before treatment and back afterwards).
- `= 1` : in the `PYJETS` commonblock. All fermions and photons can be given with status code `K(I,1)=1`, flavour code in `K(I,2)` and five-momentum (momentum, energy, mass) in `P(I,J)`. The `V` vector and remaining components in the `K` one are best put to zero. Also remember to set the total number of entries `N`.

Comment : Also colour reconnection phenomena can be studied with the `PY4FRM` routine. `MSTP(115)` can be used to switch between the scenarios, with default being no reconnection. Other reconnection parameters also work as normally, including that `MSTI(32)` can be used to find out whether a reconnection occured or not. In order for the reconnection machinery to work, the event record is automatically compelemented with information on the W^+W^- or Z^0Z^0 pair that produced the four fermions, based on the rules described above.

We remind that the four first parameters of the `PY4FRM` call are supposed to parameterize an ambiguity on the perturbative level of the process, which has to be resolved before parton showers are performed. The colour reconnection discussed here is (in most scenarios) occuring on the nonperturbative level, after the parton showers.

```
CALL PY6FRM(P12,P13,P21,P23,P31,P32,PTOP,IRAD,ITAU,ICOM)
```

Purpose: to allow a parton shower to develop and partons to hadronize from a six-fermion starting point. The initial list of fermions is supposed to be ordered in the sequence fermion (1) – antifermion (2) – fermion (3) – antifermion (4) – fermion (5) – antifermion (6). The flavour pairs should be arranged so that, if possible, the first two could come from a Z^0, the middle two from a W^+ and the last two from a W^-; else each pair should have flavours consistent with a Z^0. Specifically, this means that in a $t\bar{t}$ event, the t decay products would be found in 1 (b) and 3 and 4 (from the W^+ decay) and the $\bar{t}$ ones in 2 ($\bar{b}$) and 5 and 6 (from the W^- decay). In addition, an arbitrary number of photons may be included, e.g. from initial-state radiation; these will not be affected by the operation and can be put anywhere. Since the colour flow need not be unique, further input is needed to specify this. The number of possible interference contributions being much larger than for the four-fermion case, we have not tried to implement different strategies. Instead six probabilities may be input for the different pairings, that you e.g. could pick as the six possible squared amplitudes, or according to some more complicated scheme for how to handle the interference terms. The treatment of final-state cascades must be quite different for top events and the rest. For a normal three-boson event, each fermion pair would form one radiating system, with scale set equal to the fermion-antifermion invariant mass. (This is the relevant choice for normal fermion pair production from resonance decay, but is not suited e.g. for $\gamma\gamma$ processes dominated by small-*t* propagators.) In the top case, on the other hand, the b ($\bar{b}$) would be radiating with a recoil taken by the W^+ (W^-) in such a way that the t ($\bar{t}$) mass is preserved, while the W dipoles would radiate as normal. Therefore you need also supply a probability for the event to be a top one, again e.g. based on some squared amplitude.

`P12, P13, P21, P23, P31, P32` : relative probabilities for the six possible pairings of fermions with antifermions. The first (second) digit tells which antifermion the first (second) fermion is paired with, with the third pairing given by elimination. Thus e.g. `P23` means the first fermion is paired with the second antifermion, the second fermion with the third antifermion and the third fermion with the first antifermion. Pairings are only possible between quarks and leptons separately. The sum of probabilities for allowed pairings is automatically normalized to unity.

`PTOP` : the probability that the configuration is a top one; a number between 0 and 1. In this case, it is important that the order described above is respected, with the b and $\bar{b}$ coming first. No colour ambiguity exists if the top interpretation is selected, so then the `P12 - P32` numbers are not used.

`IRAD` : final-state QED radiation.

- `= 0` : no final-state photon radiation, only QCD showers.
- `= 1` : photon radiation inside each final fermion pair, also leptons, in addition to the QCD one for quarks.

`ITAU` : handling of τ lepton decay (where PYTHIA does not include spin effects, although some generators provide the helicity information that would allow a more sophisticated modelling).

- `= 0` : τ's are considered stable (and can therefore be decayed afterwards).
- `= 1` : τ's are allowed to decay.

`ICOM` : place where information about the event (flavours, momenta etc.) is stored at input and output.

- `= 0` : in the `HEPEVT` commonblock (meaning that information is automatically translated to `PYJETS` before treatment and back afterwards).
- `= 1` : in the `PYJETS` commonblock. All fermions and photons can be given with status code `K(I,1)=1`, flavour code in `K(I,2)` and five-momentum (momen-

tum, energy, mass) in P(I,J). The V vector and remaining components in the K one are best put to zero. Also remember to set the total number of entries N.

```
CALL PY4JET(PMAX,IRAD,ICOM)
```

Purpose: to allow a parton shower to develop and partons to hadronize from a $q\bar{q}gg$ or $q\bar{q}q'\bar{q}'$ original configuration. The partons should be ordered exactly as indicated above, with the primary $q\bar{q}$ pair first and thereafter the two gluons or the secondary $q'\bar{q}'$ pair. (Strictly speaking, the definition of primary and secondary fermion pair is ambiguous. In practice, however, differences in topological variables like the pair mass should make it feasible to have some sensible criterion on an event by event basis.) Within each pair, fermion should precede antifermion. In addition, an arbitrary number of photons may be included, e.g. from initial-state radiation; these will not be affected by the operation and can be put anywhere. The program will select a possible parton shower history from the given parton configuration, and then continue the shower from there on. The history selected is displayed in lines NOLD+1 to NOLD+6, where NOLD is the N value before the routine is called. Here the masses and energies of intermediate partons are clearly displayed. The lines NOLD+7 and NOLD+8 contain the equivalent on-mass-shell parton pair from which the shower is started.

PMAX : the maximum mass scale (in GeV) from which the shower is started in those branches that are not already fixed by the matrix-element history. If PMAX is set zero (actually below PARJ(82), the shower cutoff scale), the shower starting scale is instead set to be equal to the smallest mass of the virtual partons in the reconstructed shower history. A fixed PMAX can thus be used to obtain a reasonably exclusive set of four-jet events (to that PMAX scale), with little five-jet contamination, while the PMAX=0 option gives a more inclusive interpretation, with five- or more-jet events possible. Note that the shower is based on evolution in mass, meaning the cut is really one of mass, not of $p_\perp$, and that it may therefore be advantageous to set up the matrix elements cuts accordingly if one wishes to mix different event classes. This is not a requirement, however.

IRAD : final-state QED radiation.

- = 0 : no final-state photon radiation, only QCD showers.
- = 1 : photon radiation inside each final fermion pair, also leptons, in addition to the QCD one for quarks.

ICOM : place where information about the event (flavours, momenta etc.) is stored at input and output.

- = 0 : in the HEPEVT commonblock (meaning that information is automatically translated to PYJETS before treatment and back afterwards).
- = 1 : in the PYJETS commonblock. All fermions and photons can be given with status code K(I,1)=1, flavour code in K(I,2) and five-momentum (momentum, energy, mass) in P(I,J). The V vector and remaining components in the K one are best put to zero. Also remember to set the total number of entries N.

9.11 Other Routines and Common Blocks

The subroutines and common blocks that you will come in direct contact with have already been described. A number of other routines and common blocks exist, and those not described elsewhere are here briefly listed for the sake of completeness. The PYG*** routines are slightly modified versions of the SAS*** ones of the SASGAM library. The common block SASCOM is renamed PYINT8. If you want to use the parton distributions for standalone purposes, you are encouraged to use the original SASGAM routines rather than going the way via the PYTHIA

adaptations.

`SUBROUTINE PYINRE :` to initialize the widths and effective widths of resonances.

`SUBROUTINE PYINBM(CHFRAM,CHBEAM,CHTARG,WIN) :` to read in and identify the beam (`CHBEAM`) and target (`CHTARG`) particles and the frame (`CHFRAM`) as given in the `PYINIT` call; also to save the original energy (`WIN`).

`SUBROUTINE PYINKI(MODKI) :` to set up the event kinematics, either at initialization (`MODKI=0`) or for each separate event, the latter when the program is run with varying kinematics (`MODKI=1`).

`SUBROUTINE PYINPR :` to set up the partonic subprocesses selected with `MSEL`. For γp and γγ, also the `MSTP(14)` value affects the choice of processes. In particular, options such as `MSTP(14)=10` and `=30` sets up the several different kinds of processes that need to be mixed, with separate cuts for each.

`SUBROUTINE PYXTOT :` to give the parameterized total, double diffractive, single diffractive and elastic cross sections for different energies and colliding hadrons or photons.

`SUBROUTINE PYMAXI :` to find optimal coefficients `COEF` for the selection of kinematical variables, and to find the related maxima for the differential cross section times Jacobian factors, for each of the subprocesses included.

`SUBROUTINE PYPILE(MPILE) :` to determine the number of pile-up events, i.e. events appearing in the same beam–beam crossing.

`SUBROUTINE PYSAVE(ISAVE,IGA) :` saves and restores parameters and cross section values between the several γp and γγ components of mixing options such as `MSTP(14)=10` and `=30`. The options for `ISAVE` are (1) a complete save of all parameters specific to a given component, (2) a partial save of cross-section information, (3) a restoration of all parameters specific to a given component, (4) as 3 but preceded by a random selection of component, and (5) a summation of component cross sections (for `PYSTAT`). The subprocess code in `IGA` is the one described for `MSTI(9)`; it is input for options 1, 2 and 3 above, output for 4 and dummy for 5.

`SUBROUTINE PYGAGA(IGA) :` to generate photons according to the virtual photon flux around a lepton beam, for the `'gamma/`*lepton*`'` option in `PYINIT`.

- `IGA = 1 :` call at initialization to set up x and Q^2 limits etc.
- `IGA = 2 :` call at maximization step to give estimate of maximal photon flux factor.
- `IGA = 3 :` call at the beginning of the event generation to select the kinematics of the photon emission and to give the flux factor.
- `IGA = 4 :` call at the end of the event generation to set up the full kinematics of the photon emission.

`SUBROUTINE PYRAND :` to generate the quantities characterizing a hard scattering on the parton level, according to the relevant matrix elements.

`SUBROUTINE PYSCAT :` to find outgoing flavours and to set up the kinematics and colour flow of the hard scattering.

`SUBROUTINE PYRESD(IRES) :` to allow resonances to decay, including chains of successive decays and parton showers. Normally only two-body decays of each resonance, but a few three-body decays are also implemented.

- `IRES :` The standard call from `PYEVNT`, for the hard process, has `IRES=0`, and then finds resonances to be treated based on the subprocess number `ISUB`. In case of a nonzero `IRES` only the resonance in position `IRES` of the event record is considered. This is used by `PYEVNT` and `PYEXEC` to decay leftover resonances. (Example: a $\mathrm{b} \to \mathrm{W} + \mathrm{t}$ branching may give a t quark as beam remnant.)

`SUBROUTINE PYMULT(MMUL) :` to generate semi-hard interactions according to the multiple interaction formalism.

`SUBROUTINE PYREMN(IPU1,IPU2) :` to add on target remnants and include primordial $k_\perp$.

`SUBROUTINE PYDIFF :` to handle diffractive and elastic scattering events.

`SUBROUTINE PYDISG :` to set up kinematics, beam remnants and showers in the $2 \to 1$ DIS process $\gamma^* \mathrm{f} \to \mathrm{f}$. Currently initial-state radiation is not yet implemented, while final-

state is.

`SUBROUTINE PYDOCU` : to compute cross sections of processes, based on current Monte Carlo statistics, and to store event information in the `MSTI` and `PARI` arrays.

`SUBROUTINE PYWIDT(KFLR,SH,WDTP,WDTE)` : to calculate widths and effective widths of resonances. Everything is given in dimensions of GeV.

`SUBROUTINE PYOFSH(MOFSH,KFMO,KFD1,KFD2,PMMO,RET1,RET2)` : to calculate partial widths into channels off the mass shell, and to select correlated masses of resonance pairs.

`SUBROUTINE PYKLIM(ILIM)` : to calculate allowed kinematical limits.

`SUBROUTINE PYKMAP(IVAR,MVAR,VVAR)` : to calculate the value of a kinematical variable when this is selected according to one of the simple pieces.

`SUBROUTINE PYSIGH(NCHN,SIGS)` : to give the differential cross section (multiplied by the relevant Jacobians) for a given subprocess and kinematical setup.

`SUBROUTINE PYPDFL(KF,X,Q2,XPQ)` : to give parton distributions for p and n in the option with modified behaviour at small Q^2 and x, see `MSTP(57)`.

`SUBROUTINE PYPDFU(KF,X,Q2,XPQ)` : to give parton-distribution functions (multiplied by x, i.e. $xf_i(x,Q^2)$) for an arbitrary particle (of those recognized by PYTHIA). Generic driver routine for the following, specialized ones.

- `KF` : flavour of probed particle, according to `KF` code.
- `X` : x value at which to evaluate parton distributions.
- `Q2` : Q^2 scale at which to evaluate parton distributions.
- `XPQ` : array of dimensions `XPQ(-25:25)`, which contains the evaluated parton distributions $xf_i(x,Q^2)$. Components i ordered according to standard `KF` code; additionally the gluon is found in position 0 as well as 21 (for historical reasons).
- **Note:** the above set of calling arguments is enough for a real photon, but has to be complemented for a virtual one. This is done by `VINT(120)`.

`SUBROUTINE PYPDEL(KFA,X,Q2,XPEL)` : to give $e/\mu/\tau$ parton distributions.

`SUBROUTINE PYPDGA(X,Q2,XPGA)` : to give the photon parton distributions for sets other than the SaS ones.

`SUBROUTINE PYGGAM(ISET,X,Q2,P2,IP2,F2GM,XPDFGM)` : to construct the SaS F_2 and parton distributions of the photon by summing homogeneous (VMD) and inhomogeneous (anomalous) terms. For F_2, c and b are included by the Bethe-Heitler formula; in the '$\overline{\text{MS}}$' scheme additionally a C^γ term is added. `IP2` sets treatment of virtual photons, with same code as `MSTP(60)`. Calls `PYGVMD`, `PYGANO`, `PYGBEH`, and `PYGDIR`.

`SUBROUTINE PYGVMD(ISET,KF,X,Q2,P2,ALAM,XPGA,VXPGA)` : to evaluate the parton distributions of a VMD photon, evolved homogeneously from an initial scale P^2 to Q^2.

`SUBROUTINE PYGANO(KF,X,Q2,P2,ALAM,XPGA,VXPGA)` : to evaluate the parton distributions of the anomalous photon, inhomogeneously evolved from a scale P^2 (where it vanishes) to Q^2.

`SUBROUTINE PYGBEH(KF,X,Q2,P2,PM2,XPBH)` : to evaluate the Bethe-Heitler cross section for heavy flavour production.

`SUBROUTINE PYGDIR(X,Q2,P2,AK0,XPGA)` : to evaluate the direct contribution, i.e. the C^γ term, as needed in '$\overline{\text{MS}}$' parameterizations.

`SUBROUTINE PYPDPI(X,Q2,XPPI)` : to give pion parton distributions.

`SUBROUTINE PYPDPR(X,Q2,XPPR)` : to give proton parton distributions. Calls several auxiliary routines: `PYCTEQ`, `PYGRVL`, `PYGRVM`, `PYGRVD`, `PYGRVV`, `PYGRVW`, `PYGRVS`, `PYCT5L`, `PYCT5M` and `PYPDPO`.

`FUNCTION PYHFTH(SH,SQM,FRATT)` : to give heavy-flavour threshold factor in matrix elements.

`SUBROUTINE PYSPLI(KF,KFLIN,KFLCH,KFLSP)` : to give hadron remnant or remnants left

behind when the reacting parton is kicked out.

`FUNCTION PYGAMM(X)` : to give the value of the ordinary $\Gamma(x)$ function (used in some parton-distribution parameterizations).

`SUBROUTINE PYWAUX(IAUX,EPS,WRE,WIM)` : to evaluate the two auxiliary functions W_1 and W_2 appearing in some cross section expressions in `PYSIGH`.

`SUBROUTINE PYI3AU(EPS,RAT,Y3RE,Y3IM)` : to evaluate the auxiliary function I_3 appearing in some cross section expressions in `PYSIGH`.

`FUNCTION PYSPEN(XREIN,XIMIN,IREIM)` : to calculate the real and imaginary part of the Spence function [Hoo79].

`SUBROUTINE PYQQBH(WTQQBH)` : to calculate matrix elements for the two processes $gg \to Q\overline{Q}h^0$ and $q\overline{q} \to Q\overline{Q}h^0$.

`SUBROUTINE PYRECO(IW1,IW2,NSD1,NAFT1))` : to perform nonperturbative reconnection among strings in W^+W^- and Z^0Z^0 events. The physics of this routine is described as part of the fragmentation story, section 12.4.2, but for technical reasons the code is called directly in the event generation sequence.

`BLOCK DATA PYDATA` : to give sensible default values to all status codes and parameters.

```
COMMON/PYINT1/MINT(400),VINT(400)
```

Purpose: to collect a host of integer- and real-valued variables used internally in the program during the initialization and/or event generation stage. These variables must not be changed by you.

`MINT(1)` : specifies the general type of subprocess that has occurred, according to the `ISUB` code given in section 8.1.

`MINT(2)` : whenever `MINT(1)` (together with `MINT(15)` and `MINT(16)`) are not sufficient to specify the type of process uniquely, `MINT(2)` provides an ordering of the different possibilities, see `MSTI(2)`. Internally and temporarily, in process 53 `MINT(2)` is increased by 2 or 4 for c or b, respectively.

`MINT(3)` : number of partons produced in the hard interactions, i.e. the number n of the $2 \to n$ matrix elements used; is sometimes 3 or 4 when a basic $2 \to 1$ or $2 \to 2$ process has been convoluted with two $1 \to 2$ initial branchings (like $qq' \to q''q'''h^0$).

`MINT(4)` : number of documentation lines at the beginning of the common block `PYJETS` that are given with `K(I,1)=21`; 0 for `MSTP(125)=0`.

`MINT(5)` : number of events generated to date in current run. In runs with the variable-energy option, `MSTP(171)=1` and `MSTP(172)=2`, only those events that survive (i.e. that do not have `MSTI(61)=1`) are counted in this number. That is, `MINT(5)` may be less than the total number of `PYEVNT` calls.

`MINT(6)` : current frame of event (see `MSTP(124)` for possible values).

`MINT(7), MINT(8)` : line number for documentation of outgoing partons/particles from hard scattering for $2 \to 2$ or $2 \to 1 \to 2$ processes (else = 0).

`MINT(10)` : is 1 if cross section maximum was violated in current event, and 0 if not.

`MINT(11)` : `KF` flavour code for beam (side 1) particle.

`MINT(12)` : `KF` flavour code for target (side 2) particle.

`MINT(13), MINT(14)` : `KF` flavour codes for side 1 and side 2 initial-state shower initiators.

`MINT(15), MINT(16)` : `KF` flavour codes for side 1 and side 2 incoming partons to the hard interaction. (For use in `PYWIDT` calls, occasionally `MINT(15)=1` signals the presence of an as yet unspecified quark, but the original value is then restored afterwards.)

`MINT(17), MINT(18)` : flag to signal if particle on side 1 or side 2 has been scattered diffractively; 0 if no, 1 if yes.

`MINT(19), MINT(20)` : flag to signal initial-state structure with parton inside photon inside electron on side 1 or side 2; 0 if no, 1 if yes.

`MINT(21) - MINT(24)` : `KF` flavour codes for outgoing partons from the hard interaction.

The number of positions actually used is process-dependent, see `MINT(3)`; trailing positions not used are set = 0. For events with many outgoing partons, e.g. in external processes, also `MINT(25)` and `MINT(26)` could be used.

`MINT(25), MINT(26)` : `KF` flavour codes of the products in the decay of a single s-channel resonance formed in the hard interaction. Are thus only used when `MINT(3)=1` and the resonance is allowed to decay.

`MINT(31)` : number of hard or semi-hard scatterings that occurred in the current event in the multiple-interaction scenario; is = 0 for a low-$p_\perp$ event.

`MINT(32)` : information on whether a nonperturbative colour reconnection occurred in the current event; is 0 normally but 1 in case of reconnection.

`MINT(41), MINT(42)` : type of incoming beam or target particle; 1 for lepton and 2 for hadron. A photon counts as a lepton if it is not resolved (direct or DIS) and as a hadron if it is resolved (VMD or GVMD).

`MINT(43)` : combination of incoming beam and target particles. A photon counts as a hadron.

- `= 1` : lepton on lepton.
- `= 2` : lepton on hadron.
- `= 3` : hadron on lepton.
- `= 4` : hadron on hadron.

`MINT(44)` : as `MINT(43)`, but a photon counts as a lepton.

`MINT(45), MINT(46)` : structure of incoming beam and target particles.

- `= 1` : no internal structure, i.e. a lepton or photon carrying the full beam energy.
- `= 2` : defined with parton distributions that are not peaked at $x = 1$, i.e. a hadron or a resolved (VMD or GVMD) photon.
- `= 3` : defined with parton distributions that are peaked at $x = 1$, i.e. a resolved lepton.

`MINT(47)` : combination of incoming beam- and target-particle parton-distribution function types.

- `= 1` : no parton distribution either for beam or target.
- `= 2` : parton distributions for target but not for beam.
- `= 3` : parton distributions for beam but not for target.
- `= 4` : parton distributions for both beam and target, but not both peaked at $x = 1$.
- `= 5` : parton distributions for both beam and target, with both peaked at $x = 1$.
- `= 6` : parton distribution is peaked at $x = 1$ for target and no distribution at all for beam.
- `= 7` : parton distribution is peaked at $x = 1$ for beam and no distribution at all for target.

`MINT(48)` : total number of subprocesses switched on.

`MINT(49)` : number of subprocesses that are switched on, apart from elastic scattering and single, double and central diffractive.

`MINT(50)` : combination of incoming particles from a multiple interactions point of view.

- `= 0` : the total cross section is not known; therefore no multiple interactions are possible.
- `= 1` : the total cross section is known; therefore multiple interactions are possible if switched on. Requires beams of hadrons, VMD photons or GVMD photons.

`MINT(51)` : internal flag that event failed cuts.

- `= 0` : no problem.
- `= 1` : event failed; new one to be generated.
- `= 2` : event failed; no new event is to be generated but instead control is to be given back to used. Is intended for user-defined processes, when `NUP=0`.

`MINT(52)` : internal counter for number of lines used (in `/PYJETS/`) before multiple interactions are considered.

`MINT(53)` : internal counter for number of lines used (in `/PYJETS/`) before beam remnants

are considered.

`MINT(55)` : the heaviest new flavour switched on for QCD processes, specifically the flavour to be generated for `ISUB` = 81, 82, 83 or 84.

`MINT(56)` : the heaviest new flavour switched on for QED processes, specifically for `ISUB` = 85. Note that, unlike `MINT(55)`, the heaviest flavour may here be a lepton, and that heavy means the one with largest `KF` code.

`MINT(57)` : number of times the beam remnant treatment has failed, and the same basic kinematical setup is used to produce a new parton shower evolution and beam remnant set. Mainly used in leptoproduction, for the option when x and Q^2 are to be preserved.

`MINT(61)` : internal switch for the mode of operation of resonance width calculations in `PYWIDT` for γ^*/Z^0 or $\gamma^*/Z^0/Z'^0$.

- `= 0` : without reference to initial-state flavours.
- `= 1` : with reference to given initial-state flavours.
- `= 2` : for given final-state flavours.

`MINT(62)` : internal switch for use at initialization of h^0 width.

- `= 0` : use widths into ZZ^* or WW^* calculated before.
- `= 1` : evaluate widths into ZZ^* or WW^* for current Higgs mass.

`MINT(63)` : internal switch for use at initialization of the width of a resonance defined with `MWID(KC)=3`.

- `= 0` : use existing widths, optionally with simple energy rescaling.
- `= 1` : evaluate widths at initialization, to be used subsequently.

`MINT(65)` : internal switch to indicate initialization without specified reaction.

- `= 0` : normal initialization.
- `= 1` : initialization with argument `'none'` in `PYINIT` call.

`MINT(71)` : switch to tell whether current process is singular for $p_\perp \to 0$ or not.

- `= 0` : non-singular process, i.e. proceeding via an s-channel resonance or with both products having a mass above `CKIN(6)`.
- `= 1` : singular process.

`MINT(72)` : number of s-channel resonances that may contribute to the cross section.

`MINT(73)` : `KF` code of first s-channel resonance; 0 if there is none.

`MINT(74)` : `KF` code of second s-channel resonance; 0 if there is none.

`MINT(81)` : number of selected pile-up events.

`MINT(82)` : sequence number of currently considered pile-up event.

`MINT(83)` : number of lines in the event record already filled by previously considered pile-up events.

`MINT(84)` : `MINT(83) + MSTP(126)`, i.e. number of lines already filled by previously considered events plus number of lines to be kept free for event documentation.

`MINT(91)` : is 1 for a lepton–hadron event and 0 else. Used to determine whether a `PYFRAM(3)` call is possible.

`MINT(92)` : is used to denote region in (x, Q^2) plane when `MSTP(57)=2`, according to numbering in [Sch93a]. Simply put, 0 means that the modified proton parton distributions were not used, 1 large x and Q^2, 2 small Q^2 but large x, 3 small x but large Q^2 and 4 small x and Q^2.

`MINT(93)` : is used to keep track of parton distribution set used in the latest `STRUCTM` call to PDFLIB. The code for this set is stored in the form `MINT(93) = 1000000`$\times$`NPTYPE + 1000`$\times$`NGROUP + NSET`. The stored previous value is compared with the current new value to decide whether a `PDFSET` call is needed to switch to another set.

`MINT(101), MINT(102)` : is normally 1, but is 4 when a resolved photon (appearing on side 1 or 2) can be represented by either of the four vector mesons ρ^0, ω, ϕ and J/ψ.

`MINT(103), MINT(104)` : `KF` flavour code for the two incoming particles, i.e. the same as `MINT(11)` and `MINT(12)`. The exception is when a resolved photon is represented by a vector meson (a ρ^0, ω, ϕ or J/ψ). Then the code of the vector meson is given.

`MINT(105)` : is either `MINT(103)` or `MINT(104)`, depending on which side of the event currently is being studied.

`MINT(107), MINT(108)` : if either or both of the two incoming particles is a photon, then the respective value gives the nature assumed for that photon. The code follows the one used for `MSTP(14)`:

- `= 0` : direct photon.
- `= 1` : resolved photon.
- `= 2` : VMD-like photon.
- `= 3` : anomalous photon.
- `= 4` : DIS photon.

`MINT(109)` : is either `MINT(107)` or `MINT(108)`, depending on which side of the event currently is being studied.

`MINT(111)` : the frame given in `PYINIT` call, 0–5 for `'NONE'`, `'CMS'`, `'FIXT'`, `'3MOM'`, `'4MOM'` and `'5MOM'`, respectively, and 11 for `'USER'`.

`MINT(121)` : number of separate event classes to initialize and mix.

- `= 1` : the normal value.
- `= 2 - 13` : for $\gamma p/\gamma^* p/\gamma\gamma/\gamma^*\gamma/\gamma^*\gamma^*$ interaction when `MSTP(14)` is set to mix different photon components.

`MINT(122)` : event class used in current event for γp or $\gamma\gamma$ events generated with one of the `MSTP(14)` options mixing several event classes; code as described for `MSTI(9)`.

`MINT(123)` : event class used in the current event, with the same list of possibilities as for `MSTP(14)`, except that options 1, 4 or 10 do not appear. `= 8` denotes DIS$\times$VMD/p or vice verse, `= 9` DIS*GVMD or vice versa. Apart from a different coding, this is exactly the same information as is available in `MINT(122)`.

`MINT(141), MINT(142)` : for `'gamma/`*lepton*`'` beams, `KF` code for incoming lepton beam or target particles, while `MINT(11)` and `MINT(12)` is then the photon code. A nonzero value is the main check whether the photon emission machinery should be called at all.

`MINT(143)` : the number of tries before a successful kinematics configuration is found in `PYGAGA`, used for `'gamma/`*lepton*`'` beams. Used for the cross section updating in `PYRAND`.

`VINT(1)` : E_{cm}, c.m. energy.

`VINT(2)` : s $(= E_{\mathrm{cm}}^2)$ squared mass of complete system.

`VINT(3)` : mass of beam particle. Can be negative to denote a spacelike particle, e.g. a γ^*.

`VINT(4)` : mass of target particle. Can be negative to denote a spacelike particle, e.g. a γ^*.

`VINT(5)` : absolute value of momentum of beam (and target) particle in c.m. frame.

`VINT(6) - VINT(10)` : θ, φ and β for rotation and boost from c.m. frame to user-specified frame.

`VINT(11)` : τ_{min}.

`VINT(12)` : y_{min}.

`VINT(13)` : $\cos\hat{\theta}_{\mathrm{min}}$ for $\cos\hat{\theta} \leq 0$.

`VINT(14)` : $\cos\hat{\theta}_{\mathrm{min}}$ for $\cos\hat{\theta} \geq 0$.

`VINT(15)` : $x_{\perp\mathrm{min}}^2$.

`VINT(16)` : τ'_{min}.

`VINT(21)` : τ.

`VINT(22)` : y.

`VINT(23)` : $\cos\hat{\theta}$.

`VINT(24)` : φ (azimuthal angle).

`VINT(25)` : $x_{\perp}^2$.

`VINT(26)` : τ'.

`VINT(31)` : τ_{max}.

`VINT(32)` : y_{max}.

`VINT(33)` : $\cos\hat{\theta}_{\max}$ for $\cos\hat{\theta} \leq 0$.
`VINT(34)` : $\cos\hat{\theta}_{\max}$ for $\cos\hat{\theta} \geq 0$.
`VINT(35)` : $x^2_{\perp\max}$.
`VINT(36)` : $\tau'_{\max}$.
`VINT(41), VINT(42)` : the momentum fractions x taken by the partons at the hard interaction, as used e.g. in the parton-distribution functions.
`VINT(43)` : $\hat{m} = \sqrt{\hat{s}}$, mass of hard-scattering subsystem.
`VINT(44)` : $\hat{s}$ of the hard subprocess ($2 \to 2$ or $2 \to 1$).
`VINT(45)` : $\hat{t}$ of the hard subprocess ($2 \to 2$ or $2 \to 1 \to 2$).
`VINT(46)` : $\hat{u}$ of the hard subprocess ($2 \to 2$ or $2 \to 1 \to 2$).
`VINT(47)` : $\hat{p}_\perp$ of the hard subprocess ($2 \to 2$ or $2 \to 1 \to 2$), i.e. transverse momentum evaluated in the rest frame of the scattering.
`VINT(48)` : $\hat{p}^2_\perp$ of the hard subprocess; see `VINT(47)`.
`VINT(49)` : $\hat{m}'$, the mass of the complete three- or four-body final state in $2 \to 3$ or $2 \to 4$ processes.
`VINT(50)` : $\hat{s}' = \hat{m}'^2$; see `VINT(49)`.
`VINT(51)` : Q of the hard subprocess. The exact definition is process-dependent, see `MSTP(32)`.
`VINT(52)` : Q^2 of the hard subprocess; see `VINT(51)`.
`VINT(53)` : Q of the outer hard-scattering subprocess, used as scale for parton distribution function evaluation. Agrees with `VINT(51)` for a $2 \to 1$ or $2 \to 2$ process. For a $2 \to 3$ or $2 \to 4$ W/Z fusion process, it is set by the W/Z mass scale, and for subprocesses 121 and 122 by the heavy-quark mass.
`VINT(54)` : Q^2 of the outer hard-scattering subprocess; see `VINT(53)`.
`VINT(55)` : Q scale used as maximum virtuality in parton showers. Is equal to `VINT(53)`, except for DIS processes when `MSTP(22)`> 0.
`VINT(56)` : Q^2 scale in parton showers; see `VINT(55)`.
`VINT(57)` : $\alpha_{\rm em}$ value of hard process.
`VINT(58)` : $\alpha_{\rm s}$ value of hard process.
`VINT(59)` : $\sin\hat{\theta}$ (cf. `VINT(23)`); used for improved numerical precision in elastic and diffractive scattering.
`VINT(63), VINT(64)` : nominal m^2 values, i.e. without final-state radiation effects, for the two (or one) partons/particles leaving the hard interaction. For elastic VMD and GVMD events, this equals `VINT(69)`^{2} or `VINT(70)`^{2}, and for diffractive events it is above that.
`VINT(65)` : $\hat{p}_{\rm init}$, i.e. common nominal absolute momentum of the two partons entering the hard interaction, in their rest frame.
`VINT(66)` : $\hat{p}_{\rm fin}$, i.e. common nominal absolute momentum of the two partons leaving the hard interaction, in their rest frame.
`VINT(67), VINT(68)` : mass of beam and target particle, as `VINT(3)` and `VINT(4)`, except that an incoming γ is assigned the ρ^0, ω or ϕ mass. (This also applies for a GVMD photon, where the mass of the VMD state with the equivalent flavour content is chosen.) Used for elastic scattering $\gamma p \to \rho^0 p$ and other similar processes.
`VINT(69), VINT(70)` : the actual mass of a VMD or GVMD state; agrees with the above for VMD but is selected as a larger number for GVMD, using the approximate association $m = 2k_\perp$. Thus the mass selection for a GVMD state is according to $dm^2/(m^2+Q^2)^2$ between limits $2k_0 < m < 2k_1 = 2p_{\perp\min}(W^2)$. Required for elastic and diffractive events.
`VINT(71)` : $p_{\perp\min}$ of process, i.e. `CKIN(3)` or `CKIN(5)`, depending on which is larger, and whether the process is singular in $p_\perp \to 0$ or not.
`VINT(73)` : $\tau = m^2/s$ value of first resonance, if any; see `MINT(73)`.
`VINT(74)` : $m\Gamma/s$ value of first resonance, if any; see `MINT(73)`.

VINT(75) : $\tau = m^2/s$ value of second resonance, if any; see MINT(74).
VINT(76) : $m\Gamma/s$ value of second resonance, if any; see MINT(74).
VINT(80) : correction factor (evaluated in PYOFSH) for the cross section of resonances produced in $2 \to 2$ processes, if only some mass range of the full Breit–Wigner shape is allowed by user-set mass cuts (CKIN(2), CKIN(45) - CKIN(48)).
VINT(95) : the value of the Coulomb factor in the current event, see MSTP(40). For MSTP(40)=0 it is $= 1$, else it is > 1.
VINT(97) : an event weight, normally 1 and thus uninteresting, but for external processes with IDWTUP=-1, -2 or -3 it can be -1 for events with negative cross section, with IDWTUP=4 it can be an arbitrary non-negative weight of dimension mb, and with IDWTUP=-4 it can be an arbitrary weight of dimension mb. (The difference being that in most cases a rejection step is involved to bring the accepted events to a common weight normalization, up to a sign, while no rejection need be involved in the last two cases.)
VINT(98) : is sum of VINT(100) values for current run.
VINT(99) : is weight WTXS returned from PYEVWT call when MSTP(142)≥ 1, otherwise is 1.
VINT(100) : is compensating weight 1./WTXS that should be associated with events when MSTP(142)=1, otherwise is 1.
VINT(108) : ratio of maximum differential cross section observed to maximum differential cross section assumed for the generation; cf. MSTP(123).
VINT(109) : ratio of minimal (negative!) cross section observed to maximum differential cross section assumed for the generation; could only become negative if cross sections are incorrectly included.
VINT(111) - VINT(116) : for MINT(61)=1 gives kinematical factors for the different pieces contributing to γ^*/Z^0 or $\gamma^*/Z^0/Z'^0$ production, for MINT(61)=2 gives sum of final-state weights for the same; coefficients are given in the order pure γ^*, γ^*–Z^0 interference, γ^*–Z'^0 interference, pure Z^0, Z^0–Z'^0 interference and pure Z'^0.
VINT(117) : width of Z^0; needed in $\gamma^*/Z^0/Z'^0$ production.
VINT(120) : mass of beam or target particle, i.e. coincides with VINT(3) or VINT(4), depending on which side of the event is considered. Is used to bring information on the user-defined virtuality of a photon beam to the parton distributions of the photon.
VINT(131) : total cross section (in mb) for subprocesses allowed in the pile-up events scenario according to the MSTP(132) value.
VINT(132) : $\overline{n}$ =VINT(131)$\times$PARP(131) of pile-up events, cf. PARI(91).
VINT(133) : $\langle n \rangle = \sum_i i\mathcal{P}_i / \sum_i \mathcal{P}_i$ of pile-up events as actually simulated, i.e. $1 \leq i \leq 200$ (or smaller), see PARI(92).
VINT(134) : number related to probability to have an event in a beam–beam crossing; is $\exp(-\overline{n}) \sum_i \overline{n}^i / i!$ for MSTP(133)=1 and $\exp(-\overline{n}) \sum_i \overline{n}^i / (i-1)!$ for MSTP(133)=2, cf. PARI(93).
VINT(138) : size of the threshold factor (enhancement or suppression) in the latest event with heavy-flavour production; see MSTP(35).
VINT(141), VINT(142) : x values for the parton-shower initiators of the hardest interaction; used to find what is left for multiple interactions.
VINT(143), VINT(144) : $1 - \sum_i x_i$ for all scatterings; used for rescaling each new x-value in the multiple-interaction parton-distribution-function evaluation.
VINT(145) : estimate of total parton–parton cross section for multiple interactions; used for MSTP(82)≥ 2.
VINT(146) : common correction factor f_c in the multiple-interaction probability; used for MSTP(82)≥ 2 (part of $e(b)$, see eq. (202)).
VINT(147) : average hadronic matter overlap; used for MSTP(82)≥ 2 (needed in evaluation of $e(b)$, see eq. (202)).
VINT(148) : enhancement factor for current event in the multiple-interaction probability, defined as the actual overlap divided by the average one; used for MSTP(82)≥ 2 (is

$e(b)$ of eq. (202)).

`VINT(149)` : $x_\perp^2$ cut-off or turn-off for multiple interactions. For `MSTP(82)`≤ 1 it is $4p_{\perp\mathrm{min}}^2/W^2$, for `MSTP(82)`$\geq 2$ it is $4p_{\perp 0}^2/W^2$. For hadronic collisions, $W^2 = s$, but in photoproduction or $\gamma\gamma$ physics the W^2 scale refers to the hadronic subsystem squared energy. This may vary from event to event, so `VINT(149)` needs to be recalculated.

`VINT(150)` : probability to keep the given event in the multiple-interaction scenario with varying impact parameter, as given by the exponential factor in eq. (204).

`VINT(151), VINT(152)` : sum of x values for all the multiple-interaction partons.

`VINT(153)` : current differential cross section value obtained from `PYSIGH`; used in multiple interactions only.

`VINT(154)` : current $p_{\perp\mathrm{min}}(s)$ or $p_{\perp\mathrm{min}}(W^2)$, used for multiple interactions and also as upper cut-off k_1 if the GVMD $k_\perp$ spectrum. See comments at `VINT(149)`.

`VINT(155), VINT(156)` : the x value of a photon that branches into quarks or gluons, i.e. x at interface between initial-state QED and QCD cascades, in the old photoproduction machinery.

`VINT(157), VINT(158)` : the primordial $k_\perp$ values selected in the two beam remnants.

`VINT(159), VINT(160)` : the χ values selected for beam remnants that are split into two objects, describing how the energy is shared (see `MSTP(92)` and `MSTP(94)`); is 0 if no splitting is needed.

`VINT(161) - VINT(200)` : sum of Cabibbo–Kobayashi–Maskawa squared matrix elements that a given flavour is allowed to couple to. Results are stored in format `VINT(180+KF)` for quark and lepton flavours and antiflavours (which need not be the same; see `MDME(IDC,2)`). For leptons, these factors are normally unity.

`VINT(201) - VINT(220)` : additional variables needed in phase-space selection for $2 \to 3$ processes with `ISET(ISUB)=5`. Below indices 1, 2 and 3 refer to scattered partons 1, 2 and 3, except that the q four-momentum variables are $q_1 + q_2 \to q_1' + q_2' + q_3'$. All kinematical variables refer to the internal kinematics of the 3-body final state — the kinematics of the system as a whole is described by τ' and y, and the mass distribution of particle 3 (a resonance) by τ.

- `VINT(201)` : m_1.
- `VINT(202)` : $p_{\perp 1}^2$.
- `VINT(203)` : φ_1.
- `VINT(204)` : M_1 (mass of propagator particle).
- `VINT(205)` : weight for the $p_{\perp 1}^2$ choice.
- `VINT(206)` : m_2.
- `VINT(207)` : $p_{\perp 2}^2$.
- `VINT(208)` : φ_2.
- `VINT(209)` : M_2 (mass of propagator particle).
- `VINT(210)` : weight for the $p_{\perp 2}^2$ choice.
- `VINT(211)` : y_3.
- `VINT(212)` : $y_{3\mathrm{max}}$.
- `VINT(213)` : $\varepsilon = \pm 1$; choice between two mirror solutions $1 \leftrightarrow 2$.
- `VINT(214)` : weight associated to ε-choice.
- `VINT(215)` : $t_1 = (q_1 - q_1')^2$.
- `VINT(216)` : $t_2 = (q_2 - q_2')^2$.
- `VINT(217)` : $q_1 q_2'$ four-product.
- `VINT(218)` : $q_2 q_1'$ four-product.
- `VINT(219)` : $q_1' q_2'$ four-product.
- `VINT(220)` : $\sqrt{(m_{\perp 12}^2 - m_{\perp 1}^2 - m_{\perp 2}^2)^2 - 4m_{\perp 1}^2 m_{\perp 2}^2}$, where $m_{\perp 12}$ is the transverse mass of the $q_1' q_2'$ system.

`VINT(221) - VINT(225)` : θ, φ and β for rotation and boost from c.m. frame to hadronic

c.m. frame of a lepton–hadron event.

`VINT(231)` : $Q^2_{\rm min}$ scale for current parton-distribution function set.

`VINT(232)` : valence quark distribution of a VMD photon; set in `PYPDFU` and used in `PYPDFL`.

`VINT(281)` : for resolved photon events, it gives the ratio between the total γX cross section and the total $\pi^0 X$ cross section, where X represents the target particle.

`VINT(283), VINT(284)` : virtuality scale at which a GVMD/anomalous photon on the beam or target side of the event is being resolved. More precisely, it gives the $k_\perp^2$ of the $\gamma \to q\overline{q}$ vertex. For elastic and diffractive scatterings, $m^2/4$ is stored, where m is the mass of the state being diffracted. For clarity, we point out that elastic and diffractive events are characterized by the mass of the diffractive states but without any primordial $k_\perp$, while jet production involves a primordial $k_\perp$ but no mass selection. Both are thus not used at the same time, but for GVMD/anomalous photons, the standard (though approximate) identification $k_\perp^2 = m^2/4$ ensures agreement between the two applications.

`VINT(285)` : the `CKIN(3)` value provided by you at initialization; subsequently `CKIN(3)` may be overwritten (for `MSTP(14)=10`) but `VINT(285)` stays.

`VINT(289)` : squared c.m. energy found in `PYINIT` call.

`VINT(290)` : the `WIN` argument of a `PYINIT` call.

`VINT(291) - VINT(300)` : the two five-vectors of the two incoming particles, as reconstructed in `PYINKI`. These may vary from one event to the next.

`VINT(301) - VINT(320)` : used when a flux of virtual photons is being generated by the `PYGAGA` routine, for `'gamma/`*lepton*`'` beams.

- `VINT(301)` : c.m. energy for the full collision, while `VINT(1)` gives the γ-hadron or $\gamma\gamma$ subsystem energy.
- `VINT(302)` : full squared c.m. energy, while `VINT(2)` gives the subsystem squared energy.
- `VINT(303), VINT(304)` : mass of the beam or target lepton, while `VINT(3)` or `VINT(4)` give the mass of a photon emitted off it.
- `VINT(305), VINT(306)` : x values, i.e. respective photon energy fractions of the incoming lepton in the c.m. frame of the event.
- `VINT(307), VINT(308)` : Q^2 or P^2, virtuality of the respective photon (thus the square of `VINT(3), VINT(4)`).
- `VINT(309), VINT(310)` : y values, i.e. respective photon light-cone energy fraction of the lepton.
- `VINT(311), VINT(312)` : θ, scattering angle of the respective lepton in the c.m. frame of the event.
- `VINT(315), VINT(316)`: the R factor defined at `MSTP(17)`, giving a cross section enhancement from the contribution of resolved longitudinal photons.
- `VINT(317)` : dipole suppression factor in `PYXTOT` for current event.
- `VINT(318)` : dipole suppression factor in `PYXTOT` at initialization.
- `VINT(313), VINT(314)` : ϕ, azimuthal angle of the respective scattered lepton in the c.m. frame of the event.
- `VINT(319)` : photon flux factor in `PYGAGA` for current event.
- `VINT(320)` : photon flux factor in `PYGAGA` at initialization.

```
COMMON/PYINT2/ISET(500),KFPR(500,2),COEF(500,20),ICOL(40,4,2)
```

Purpose: to store information necessary for efficient generation of the different subprocesses, specifically type of generation scheme and coefficients of the Jacobian. Also to store allowed colour-flow configurations. These variables must not be changed by you.

`ISET(ISUB)` : gives the type of kinematical-variable selection scheme used for subprocess

ISUB.

= 0 : elastic, diffractive and low-$p_\perp$ processes.
= 1 : $2 \to 1$ processes (irrespective of subsequent decays).
= 2 : $2 \to 2$ processes (i.e. the bulk of processes).
= 3 : $2 \to 3$ processes (like $\mathrm{qq}' \to \mathrm{q}''\mathrm{q}'''\mathrm{h}^0$).
= 4 : $2 \to 4$ processes (like $\mathrm{qq}' \to \mathrm{q}''\mathrm{q}'''\mathrm{W}^+\mathrm{W}^-$).
= 5 : 'true' $2 \to 3$ processes, one method.
= 8 : $2 \to 1$ process $\gamma^*\mathrm{f}_i \to f_i$ where, unlike the $2 \to 1$ processes above, $\hat{s} = 0$.
= 9 : $2 \to 2$ in multiple interactions ($p_\perp$ as kinematics variable).
= 11 : a user-defined process.
= -1 : legitimate process which has not yet been implemented.
= -2 : ISUB is an undefined process code.

KFPR(ISUB,J) : give the KF flavour codes for the products produced in subprocess ISUB. If there is only one product, the J=2 position is left blank. Also, quarks and leptons assumed massless in the matrix elements are denoted by 0. The main application is thus to identify resonances produced (Z^0, $\mathrm{W}^\pm$, h^0, etc.). For external processes, KFPR instead stores information on process numbers in the two external classifications, see subsection 9.9.

COEF(ISUB,J) : factors used in the Jacobians in order to speed up the selection of kinematical variables. More precisely, the shape of the cross section is given as the sum of terms with different behaviour, where the integral over the allowed phase space is unity for each term. COEF gives the relative strength of these terms, normalized so that the sum of coefficients for each variable used is unity. Note that which coefficients are indeed used is process-dependent.

ISUB : standard subprocess code.
J = 1 : τ selected according $1/\tau$.
J = 2 : τ selected according to $1/\tau^2$.
J = 3 : τ selected according to $1/(\tau(\tau+\tau_R))$, where $\tau_R = m_R^2/s$ is τ value of resonance; only used for resonance production.
J = 4 : τ selected according to Breit–Wigner of form $1/((\tau-\tau_R)^2+\gamma_R^2)$, where $\tau_R = m_R^2/s$ is τ value of resonance and $\gamma_R = m_R\Gamma_R/s$ is its scaled mass times width; only used for resonance production.
J = 5 : τ selected according to $1/(\tau(\tau+\tau_{R'}))$, where $\tau_{R'} = m_{R'}^2/s$ is τ value of second resonance; only used for simultaneous production of two resonances.
J = 6 : τ selected according to second Breit–Wigner of form $1/((\tau-\tau_{R'})^2+\gamma_{R'}^2)$, where $\tau_{R'} = m_{R'}^2/s$ is τ value of second resonance and $\gamma_{R'} = m_{R'}\Gamma_{R'}/s$ is its scaled mass times width; is used only for simultaneous production of two resonances, like $\gamma^*/\mathrm{Z}^0/\mathrm{Z}'^0$.
J = 7 : τ selected according to $1/(1-\tau)$; only used when both parton distributions are peaked at $x = 1$.
J = 8 : y selected according to $y - y_{\min}$.
J = 9 : y selected according to $y_{\max} - y$.
J = 10 : y selected according to $1/\cosh(y)$.
J = 11 : y selected according to $1/(1-\exp(y-y_{\max}))$; only used when beam parton distribution is peaked close to $x = 1$.
J = 12 : y selected according to $1/(1-\exp(y_{\min}-y))$; only used when target parton distribution is peaked close to $x = 1$.
J = 13 : $z = \cos\hat{\theta}$ selected evenly between limits.
J = 14 : $z = \cos\hat{\theta}$ selected according to $1/(a-z)$, where $a = 1 + 2m_3^2m_4^2/\hat{s}^2$, m_3 and m_4 being the masses of the two final-state particles.
J = 15 : $z = \cos\hat{\theta}$ selected according to $1/(a+z)$, with a as above.
J = 16 : $z = \cos\hat{\theta}$ selected according to $1/(a-z)^2$, with a as above.

J = 17 : $z = \cos\hat{\theta}$ selected according to $1/(a+z)^2$, with a as above.
J = 18 : τ' selected according to $1/\tau'$.
J = 19 : τ' selected according to $(1-\tau/\tau')^3/\tau'^2$.
J = 20 : τ' selected according to $1/(1-\tau')$; only used when both parton distributions are peaked close to $x = 1$.

ICOL : contains information on different colour-flow topologies in hard $2 \to 2$ processes.

```
COMMON/PYINT3/XSFX(2,-40:40),ISIG(1000,3),SIGH(1000)
```

Purpose: to store information on parton distributions, subprocess cross sections and different final-state relative weights. These variables must not be changed by you.

XSFX : current values of parton-distribution functions $xf(x)$ on beam and target side.

ISIG(ICHN,1) : incoming parton/particle on the beam side to the hard interaction for allowed channel number ICHN. The number of channels filled with relevant information is given by NCHN, one of the arguments returned in a PYSIGH call. Thus only $1 \leq$ICHN$\leq$NCHN is filled with relevant information.

ISIG(ICHN,2) : incoming parton/particle on the target side to the hard interaction for allowed channel number ICHN. See also comment above.

ISIG(ICHN,3) : colour-flow type for allowed channel number ICHN; see MSTI(2) list. See also above comment. For 'subprocess' 96 uniquely, ISIG(ICHN,3) is also used to translate information on what is the correct subprocess number (11, 12, 13, 28, 53 or 68); this is used for reassigning subprocess 96 to either of these.

SIGH(ICHN) : evaluated differential cross section for allowed channel number ICHN, i.e. matrix-element value times parton distributions, for current kinematical setup (in addition, Jacobian factors are included in the numbers, as used to speed up generation). See also comment for ISIG(ICHN,1).

```
COMMON/PYINT4/MWID(500),WIDS(500,5)
```

Purpose: to store character of resonance width treatment and partial and effective decay widths for the different resonances. These variables must not be changed by you.

MWID(KC) : gives the character of particle with compressed code KC, mainly as used in PYWIDT to calculate widths of resonances (not necessarily at the nominal mass).

= 0 : an ordinary particle; not to be treated as resonance.

= 1 : a resonance for which the partial and total widths (and hence branching ratios) are dynamically calculated in PYWIDT calls; i.e. special code has to exist for each such particle. The effects of allowed/unallowed secondary decays are included, both in the relative composition of decays and in the process cross section.

= 2 : The total width is taken to be the one stored in PMAS(KC,2) and the relative branching ratios the ones in BRAT(IDC) for decay channels IDC. There is then no need for any special code in PYWIDT to handle a resonance. During the run, the stored PMAS(KC,2) and BRAT(IDC) values are used to calculate the total and partial widths of the decay channels. Some extra information and assumptions are then used. Firstly, the stored BRAT values are assumed to be the full branching ratios, including all possible channels and all secondary decays. The actual relative branching fractions are modified to take into account that the simulation of some channels may be switched off (even selectively for a particle and an antiparticle), as given by MDME(IDC,1), and that some secondary channels may not be allowed, as expressed by the WIDS factors. This also goes into process cross sections. Secondly, it is assumed that all widths scale like $\sqrt{\hat{s}}/m$, the ratio of the actual to the nominal mass. A further

nontrivial change as a function of the actual mass can be set for each channel by the `MDME(IDC,2)` value, see section 14.4.

`= 3 :` a hybrid version of options 1 and 2 above. At initialization the `PYWIDT` code is used to calculate `PMAS(KC,2)` and `BRAT(IDC)` at the nominal mass of the resonance. Special code must then exist in `PYWIDT` for the particle. The `PMAS(KC,2)` and `BRAT(IDC)` values overwrite the default ones. In the subsequent generation of events, the simpler scheme of option 2 is used, thus saving some execution time.

Note: the Z and Z′ cannot be used with options 2 and 3, since the more complicated interference structure implemented for those particles is only handled correctly for option 1.

`WIDS(KC,J) :` gives the dimensionless suppression factor to cross sections caused by the closing of some secondary decays, as calculated in `PYWIDT`. It is defined as the ratio of the total width of channels switched on to the total width of all possible channels (replace width by squared width for a pair of resonances). The on/off status of channels is set by the `MDME` switches; see section 14.4. The information in `WIDS` is used e.g. in cross-section calculations. Values are built up recursively from the lightest particle to the heaviest one at initialization, with the exception that W and Z are done already from the beginning (since these often are forced off the mass shell). `WIDS` can go wrong in case you have perverse situations where the branching ratios vary rapidly as a function of energy, across the resonance shape. This then influences process cross sections.

`KC :` standard `KC` code for resonance considered.

`J = 1 :` suppression when a pair of resonances of type `KC` are produced together. When an antiparticle exists, the particle–antiparticle pair (such as W^+W^-) is the relevant combination, else the particle–particle one (such as Z^0Z^0).

`J = 2 :` suppression for a particle of type `KF` when produced on its own, or together with a particle of another type.

`J = 3 :` suppression for an antiparticle of type `KF` when produced on its own, or together with a particle of another type.

`J = 4 :` suppression when a pair of two identical particles are produced, for a particle which has a nonidentical antiparticle (e.g. W^+W^+).

`J = 5 :` suppression when a pair of two identical antiparticles are produced, for a particle which has a nonidentical antiparticle (e.g. W^-W^-).

```
COMMON/PYINT5/NGENPD,NGEN(0:500,3),XSEC(0:500,3)
```

Purpose: to store information necessary for cross-section calculation and differential cross-section maximum violation. These variables must not be changed by you.

`NGEN(ISUB,1) :` gives the number of times that the differential cross section (times Jacobian factors) has been evaluated for subprocess `ISUB`, with `NGEN(0,1)` the sum of these.

`NGEN(ISUB,2) :` gives the number of times that a kinematical setup for subprocess `ISUB` is accepted in the generation procedure, with `NGEN(0,2)` the sum of these.

`NGEN(ISUB,3) :` gives the number of times an event of subprocess type `ISUB` is generated, with `NGEN(0,3)` the sum of these. Usually `NGEN(ISUB,3) = NGEN(ISUB,2)`, i.e. an accepted kinematical configuration can normally be used to produce an event.

`XSEC(ISUB,1) :` estimated maximum differential cross section (times the Jacobian factors used to speed up the generation process) for the different subprocesses in use, with `XSEC(0,1)` the sum of these (except low-$p_\perp$, i.e. `ISUB` = 95). For external processes special rules may apply, see subsection 9.9. In particular, negative cross sections and maxima may be allowed. In this case, `XSEC(ISUB,1)` stores the absolute value of the maximum, since this is the number that allows the appropriate mixing of

subprocesses.

`XSEC(ISUB,2)` : gives the sum of differential cross sections (times Jacobian factors) for the `NGEN(ISUB,1)` phase-space points evaluated so far.

`XSEC(ISUB,3)` : gives the estimated integrated cross section for subprocess `ISUB`, based on the statistics accumulated so far, with `XSEC(0,3)` the estimated total cross section for all subprocesses included (all in mb). This is exactly the information obtainable by a `PYSTAT(1)` call.

Warning : For γp and $\gamma\gamma$ events, when several photon components are mixed (see `MSTP(14)`), a master copy of these numbers for each component is stored in the `PYSAVE` routine. What is then visible after each event is only the numbers for the last component considered, not the full statistics. A special `PYSAVE` call, perfomed e.g. in `PYSTAT`, is required to obtain the sum of all the components.

```
COMMON/PYINT6/PROC(0:500)
CHARACTER PROC*28
```

Purpose: to store character strings for the different possible subprocesses; used when printing tables.

`PROC(ISUB)` : name for the different subprocesses, according to `ISUB` code. `PROC(0)` denotes all processes.

```
COMMON/PYINT7/SIGT(0:6,0:6,0:5)
```

Purpose: to store information on total, elastic and diffractive cross sections. These variables should only be set by you for the option `MSTP(31)=0`; else they should not be touched. All numbers are given in mb.

`SIGT(I1,I2,J)` : the cross section, both total and subdivided by class (elastic, diffractive etc.). For a photon to be considered as a VMD meson the cross sections are additionally split into the contributions from the various meson states.

- `I1, I2` : allowed states for the incoming particle on side 1 and 2, respectively.
  - `= 0` : sum of all allowed states. Except for a photon to be considered as a VMD meson this is the only nonvanishing entry.
  - `= 1` : the contribution from the ρ^0 VMD state.
  - `= 2` : the contribution from the ω VMD state.
  - `= 3` : the contribution from the ϕ VMD state.
  - `= 4` : the contribution from the J/ψ VMD state.
  - `= 5, 6` : reserved for future use.
- `J` : the total and partial cross sections.
  - `= 0` : the total cross section.
  - `= 1` : the elastic cross section.
  - `= 2` : the single diffractive cross section $AB \to XB$.
  - `= 3` : the single diffractive cross section $AB \to AX$.
  - `= 4` : the double diffractive cross section.
  - `= 5` : the inelastic, non-diffractive cross section.

Warning: If you set these values yourself, it is important that they are internally consistent, since this is not explicitly checked by the program. Thus the contributions `J=`1–5 should add up to the `J=0` one and, for VMD photons, the contributions `I=`1–4 should add up to the `I=0` one.

```
 COMMON/PYINT8/XPVMD(-6:6),XPANL(-6:6),XPANH(-6:6),XPBEH(-6:6),
&XPDIR(-6:6)
```

Purpose: to store the various components of the photon parton distributions when the `PYGGAM` routine is called.

`XPVMD(KFL)` : gives distributions of the VMD part (ρ^0, ω and ϕ).

`XPANL(KFL)` : gives distributions of the anomalous part of light quarks (d, u and s).

`XPANH(KFL)` : gives distributions of the anomalous part of heavy quarks (c and b).

`XPBEH(KFL)` : gives Bethe-Heitler distributions of heavy quarks (c and b). This provides an alternative to `XPANH`, i.e. both should not be used at the same time.

`XPDIR(KFL)` : gives direct correction to the production of light quarks (d, u and s). This term is nonvanishing only in the $\overline{\mathrm{MS}}$ scheme, and is applicable for F_2^γ rather than for the parton distributions themselves.

```
 COMMON/PYINT9/VXPVMD(-6:6),VXPANL(-6:6),VXPANH(-6:6),VXPDGM(-6:6)
```

Purpose: to give the valence parts of the photon parton distributions (x-weighted, as usual) when the `PYGGAM` routine is called. Companion to `/PYINT8/`, which gives the total parton distributions.

`VXPVMD(KFL)` : valence distributions of the VMD part; matches `XPVMD` in `/PYINT8/`.

`VXPANL(KFL)` : valence distributions of the anomalous part of light quarks; matches `XPANL` in `/PYINT8/`.

`VXPANH(KFL)` : valence distributions of the anomalous part of heavy quarks; matches `XPANH` in `/PYINT8/`.

`VXPDGM(KFL)` : gives the sum of valence distributions parts; matches `XPDFGM` in the `PYGGAM` call.

Note 1: the Bethe-Heitler and direct contributions in `XPBEH(KFL)` and `XPDIR(KFL)` in `/PYINT8/` are pure valence-like, and therefore are not duplicated here.

Note 2: the sea parts of the distributions can be obtained by taking the appropriate differences between the total distributions and the valence distributions.

10 Initial- and Final-State Radiation

Starting from the hard interaction, initial- and final-state radiation corrections may be added. This is normally done by making use of the parton-shower language — only for the $e^+e^- \to q\overline{q}$ process does PYTHIA offer a matrix-element option (described in section 6.1). The algorithms used to generate initial- and final-state showers are rather different, and are therefore described separately below, starting with the conceptually easier final-state one. Before that, some common elements are introduced.

The main references for final-state showers is [Ben87a, Nor01] and for initial-state ones [Sjö85, Miu99].

10.1 Shower Evolution

In the leading-logarithmic picture, a shower may be viewed as a sequence of $1 \to 2$ branchings $a \to bc$. Here a is called the mother and b and c the two daughters. Each daughter is free to branch in its turn, so that a tree-like structure can evolve. We will use the word 'parton' for all the objects a, b and c involved in the branching process, i.e. not only for quarks and gluons but also for leptons and photons. The branchings included in the program are $q \to qg$, $g \to gg$, $g \to q\overline{q}$, $q \to q\gamma$ and $\ell \to \ell\gamma$. Photon branchings, i.e. $\gamma \to q\overline{q}$ and $\gamma \to \ell\overline{\ell}$, have not been included so far, since they are reasonably rare and since no urgent need for them has been perceived.

A word on terminology may be in place. The algorithms described here are customarily referred to as leading-log showers. This is correct insofar as no explicit corrections from higher orders are included, i.e. there are no $O(\alpha_s^2)$ terms in the splitting kernels, neither by new $1 \to 3$ processes nor by corrections to the $1 \to 2$ ones. However, it is grossly misleading if leading-log showers are equated with leading-log analytical calculations. In particular, the latter contain no constraints from energy–momentum conservation: the radiation off a quark is described in the approximation that the quark does not lose any energy when a gluon is radiated, so that the effects of multiple emissions factorize. Therefore energy–momentum conservation is classified as a next-to-leading-log correction. In a Monte Carlo shower, on the other hand, energy–momentum conservation is explicit branching by branching. By including coherence phenomena and optimized choices of α_s scales, further information on higher orders is inserted. While the final product is still not certified fully to comply with a NLO/NLL standard, it is well above the level of an unsophisticated LO/LL analytic calculation.

10.1.1 The evolution equations

In the shower formulation, the kinematics of each branching is given in terms of two variables, Q^2 and z. Slightly different interpretations may be given to these variables, and indeed this is one main area where the various programs on the market differ. Q^2 has dimensions of squared mass, and is related to the mass or transverse momentum scale of the branching. z gives the sharing of the a energy and momentum between the two daughters, with parton b taking a fraction z and parton c a fraction $1-z$. To specify the kinematics, an azimuthal angle φ of the b around the a direction is needed in addition; in the simple discussions φ is chosen to be isotropically distributed, although options for non-isotropic distributions currently are the defaults.

The probability for a parton to branch is given by the evolution equations (also called DGLAP or Altarelli–Parisi [Gri72, Alt77]). It is convenient to introduce

$$t = \ln(Q^2/\Lambda^2) \quad \Rightarrow \quad \mathrm{d}t = \mathrm{d}\ln(Q^2) = \frac{\mathrm{d}Q^2}{Q^2} \,, \tag{147}$$

where Λ is the QCD Λ scale in α_s. Of course, this choice is more directed towards the QCD parts of the shower, but it can be used just as well for the QED ones. In terms of the two

variables t and z, the differential probability $\mathrm{d}\mathcal{P}$ for parton a to branch is now

$$\mathrm{d}\mathcal{P}_a = \sum_{b,c} \frac{\alpha_{abc}}{2\pi} P_{a\to bc}(z)\,\mathrm{d}t\,\mathrm{d}z \ . \tag{148}$$

Here the sum is supposed to run over all allowed branchings, for a quark $\mathrm{q} \to \mathrm{qg}$ and $\mathrm{q} \to \mathrm{q}\gamma$, and so on. The α_{abc} factor is α_{em} for QED branchings and α_{s} for QCD ones (to be evaluated at some suitable scale, see below).

The splitting kernels $P_{a\to bc}(z)$ are

$$\begin{aligned}
P_{\mathrm{q}\to\mathrm{qg}}(z) &= C_F\,\frac{1+z^2}{1-z} \ , \\
P_{\mathrm{g}\to\mathrm{gg}}(z) &= N_C\,\frac{(1-z(1-z))^2}{z(1-z)} \ , \\
P_{\mathrm{g}\to\mathrm{q}\overline{\mathrm{q}}}(z) &= T_R\,(z^2+(1-z)^2) \ , \\
P_{\mathrm{q}\to\mathrm{q}\gamma}(z) &= e_{\mathrm{q}}^2\,\frac{1+z^2}{1-z} \ , \\
P_{\ell\to\ell\gamma}(z) &= e_{\ell}^2\,\frac{1+z^2}{1-z} \ ,
\end{aligned} \tag{149}$$

with $C_F = 4/3$, $N_C = 3$, $T_R = n_f/2$ (i.e. T_R receives a contribution of $1/2$ for each allowed $\mathrm{q}\overline{\mathrm{q}}$ flavour), and e_{q}^2 and e_{ℓ}^2 the squared electric charge ($4/9$ for u-type quarks, $1/9$ for d-type ones, and 1 for leptons).

Persons familiar with analytical calculations may wonder why the '+ prescriptions' and $\delta(1-z)$ terms of the splitting kernels in eq. (149) are missing. These complications fulfil the task of ensuring flavour and energy conservation in the analytical equations. The corresponding problem is solved trivially in Monte Carlo programs, where the shower evolution is traced in detail, and flavour and four-momentum are conserved at each branching. The legacy left is the need to introduce a cut-off on the allowed range of z in splittings, so as to avoid the singular regions corresponding to excessive production of very soft gluons.

Also note that $P_{\mathrm{g}\to\mathrm{gg}}(z)$ is given here with a factor N_C in front, while it is sometimes shown with $2N_C$. The confusion arises because the final state contains two identical partons. With the normalization above, $P_{a\to bc}(z)$ is interpreted as the branching probability for the original parton a. On the other hand, one could also write down the probability that a parton b is produced with a fractional energy z. Almost all the above kernels can be used unchanged also for this purpose, with the obvious symmetry $P_{a\to bc}(z) = P_{a\to cb}(1-z)$. For $\mathrm{g} \to \mathrm{gg}$, however, the total probability to find a gluon with energy fraction z is the sum of the probability to find either the first or the second daughter there, and that gives the factor of 2 enhancement.

10.1.2 The Sudakov form factor

The t variable fills the function of a kind of time for the shower evolution. In final-state showers, t is constrained to be gradually decreasing away from the hard scattering, in initial-state ones to be gradually increasing towards the hard scattering. This does not mean that an individual parton runs through a range of t values: in the end, each parton is associated with a fixed t value, and the evolution procedure is just a way of picking that value. It is only the ensemble of partons in many events that evolves continuously with t, cf. the concept of parton distributions.

For a given t value we define the integral of the branching probability over all allowed z values,

$$I_{a\to bc}(t) = \int_{z_-(t)}^{z_+(t)} \mathrm{d}z\,\frac{\alpha_{abc}}{2\pi} P_{a\to bc}(z) \ . \tag{150}$$

The naïve probability that a branching occurs during a small range of t values, δt, is given by $\sum_{b,c} I_{a\to bc}(t)\,\delta t$, and thus the probability for no emission by $1-\sum_{b,c} I_{a\to bc}(t)\,\delta t$.

If the evolution of parton a starts at a 'time' t_0, the probability that the parton has not yet branched at a 'later time' $t > t_0$ is given by the product of the probabilities that it did not branch in any of the small intervals δt between t_0 and t. In other words, letting $\delta t \to 0$, the no-branching probability exponentiates:

$$\mathcal{P}_{\mathrm{no-branching}}(t_0,t) = \exp\left\{-\int_{t_0}^{t} \mathrm{d}t' \sum_{b,c} I_{a\to bc}(t')\right\} = S_a(t)\ . \tag{151}$$

Thus the actual probability that a branching of a occurs at t is given by

$$\frac{\mathrm{d}\mathcal{P}_a}{\mathrm{d}t} = -\frac{\mathrm{d}\mathcal{P}_{\mathrm{no-branching}}(t_0,t)}{\mathrm{d}t} = \left(\sum_{b,c} I_{a\to bc}(t)\right)\exp\left\{-\int_{t_0}^{t} \mathrm{d}t' \sum_{b,c} I_{a\to bc}(t')\right\}\ . \tag{152}$$

The first factor is the naïve branching probability, the second the suppression due to the conservation of total probability: if a parton has already branched at a 'time' $t' < t$, it can no longer branch at t. This is nothing but the exponential factor that is familiar from radioactive decay. In parton-shower language the exponential factor $S_a(t) = \mathcal{P}_{\mathrm{no-branching}}(t_0,t)$ is referred to as the Sudakov form factor [Sud56].

The ordering in terms of increasing t above is the appropriate one for initial-state showers. In final-state showers the evolution is from an initial t_{max} (set by the hard scattering) and towards smaller t. In that case the integral from t_0 to t in eqs. (151) and (152) is replaced by an integral from t to t_{max}. Since, by convention, the Sudakov factor is still defined from the lower cut-off t_0, i.e. gives the probability that a parton starting at scale t will not have branched by the lower cut-off scale t_0, the no-branching factor is actually $\mathcal{P}_{\mathrm{no-branching}}(t_{\mathrm{max}},t) = S_a(t_{\mathrm{max}})/S_a(t)$.

We note that the above structure is exactly of the kind discussed in section 4.2. The veto algorithm is therefore extensively used in the Monte Carlo simulation of parton showers.

10.1.3 Matching to the hard scattering

The evolution in Q^2 is begun from some maximum scale Q^2_{max} for final-state parton showers, and is terminated at (a possibly different) Q^2_{max} for initial-state showers. In general Q^2_{max} is not known. Indeed, since the parton-shower language does not guarantee agreement with higher-order matrix-element results, neither in absolute shape nor normalization, there is no unique prescription for a 'best' choice. Generically Q_{max} should be of the order of the hard-scattering scale, i.e. the largest virtuality should be associated with the hard scattering, and initial- and final-state parton showers should only involve virtualities smaller than that. This may be viewed just as a matter of sound book-keeping: in a $2\to n$ graph, a $2\to 2$ hard-scattering subgraph could be chosen several different ways, but if all the possibilities were to be generated then the cross section would be double-counted. Therefore one should define the $2\to 2$ 'hard' piece of a $2\to n$ graph as the one that involves the largest virtuality.

Of course, the issue of double-counting depends a bit on what processes are actually generated in the program. If one considers a $\mathrm{q\bar{q}g}$ final state in hadron colliders, this could come either as final-state radiation off a $\mathrm{q\bar{q}}$ pair, or by a gluon splitting in a $\mathrm{q\bar{q}}$ pair, or many other ways, so that the danger of double-counting is very real. On the other hand, consider the production of a low-$p_\perp$, low-mass Drell–Yan pair of leptons, together with two quark jets. Such a process in principle could proceed by having a γ^* emitted off a quark leg, with a quark–quark scattering as hard interaction. However, since this process is not included in the program, there is no actual danger of (this particular) double-counting, and so the scale of evolution could be picked larger than the mass of the Drell–Yan pair, as we shall see.

For most $2\to 2$ scattering processes in PYTHIA, the Q^2 scale of the hard scattering is chosen to be $Q^2_{\mathrm{hard}} = p_\perp^2$ (when the final-state particles are massless, otherwise masses are added). In

final-state showers, where Q is associated with the mass of the branching parton, transverse momenta generated in the shower are constrained by $p_\perp < Q/2$. An ordering that the shower $p_\perp$ should be smaller than the hard-scattering $p_\perp$ therefore corresponds roughly to $Q^2_{\rm max} = 4Q^2_{\rm hard}$, which is the default assumption. The constraints are slightly different for initial-state showers, where the spacelike virtuality Q^2 attaches better to $p^2_\perp$, and therefore $Q^2_{\rm max} = Q^2_{\rm hard}$ is a sensible default. We iterate that these limits, set by `PARP(71)` and `PARP(67)`, respectively, are imagined sensible when there is a danger of doublecounting; if not, large values could well be relevant to cover a wider range of topologies, but always with some caution. (See also `MSTP(68)`.)

The situation is rather better for the final-state showers in the decay of any colour-singlet particles, or coloured but reasonably long-lived ones, such as the Z^0 or the h^0, either as part of a hard $2 \to 1 \to 2$ process, or anywhere else in the final state. Then we know that $Q_{\rm max}$ has to be put equal to the particle mass. It is also possible to match the parton-shower evolution to the first-order matrix-element results.

QCD processes such as $\mathrm{qg} \to \mathrm{qg}$ pose a special problem when the scattering angle is small. Coherence effects (see below) may then restrict the emission further than what is just given by the $Q_{\rm max}$ scale introduced above. This is most easily viewed in the rest frame of the $2 \to 2$ hard scattering subprocess. Some colours flow from the initial to the final state. The radiation associated with such a colour flow should be restricted to a cone with opening angle given by the difference between the original and the final colour directions; there is one such cone around the incoming parton for initial state radiation and one around the outgoing parton for final state radiation. Colours that are annihilated or created in the process effectively correspond to an opening angle of 180° and therefore the emission is not constrained for these. For a gluon, which have two colours and therefore two different cones, a random choice is made between the two for the first branching. Further, coherence effects also imply azimuthal anisotropies of the emission inside the allowed cones.

Finally, we note that there can be some overlap between descriptions of the same process. Section 8.4.2 gives two examples. One is the correspondence between the description of a single W or Z with additional jet production by showering, or the same picture obtained by using explicit matrix elements to generate at least one jet in association with the W/Z. The other is the generation of $\mathrm{Z}^0\mathrm{b}\overline{\mathrm{b}}$ final states either starting from $\mathrm{b}\overline{\mathrm{b}} \to \mathrm{Z}^0$, or from $\mathrm{bg} \to \mathrm{Z}^0\mathrm{b}$ or from $\mathrm{gg} \to \mathrm{b}\overline{\mathrm{b}}\mathrm{Z}^0$. As a rule of thumb, to be used with common sense, one would start from as low an order as possible for an inclusive description, where the low-$p_\perp$ region is likely to generate most of the cross section, whereas higher-order topologies are more relevant for studies of exclusive event samples at high $p_\perp$.

10.2 Final-State Showers

Final-state showers are time-like, i.e. all virtualities $m^2 = E^2 - \mathbf{p}^2 \geq 0$. The maximum allowed virtuality scale $Q^2_{\rm max}$ is set by the hard-scattering process, and thereafter the virtuality is decreased in each subsequent branching, down to the cut-off scale Q^2_0. This cut-off scale is used to regulate both soft and collinear divergences in the emission probabilities.

The main points of the PYTHIA showering algorithm are as follows.

- It is a leading-log algorithm, of the improved, coherent kind, i.e. with angular ordering.
- It can be used for an arbitrary initial pair of partons or, in fact, for any number between one and seven given entities (including hadrons and gauge bosons) although only quarks, gluons, leptons, squarks and gluinos can initiate a shower.
- The pair of showering partons may be given in any frame, but the evolution is carried out in the c.m. frame of the showering partons.
- Energy and momentum are conserved exactly at each step of the showering process.
- If the initial pair comes from the decay of a known resonance, an additional rejection technique is used in the gluon emission off a parton of the pair, so as to reproduce the lowest-order differential 3-jet cross section.

- In subsequent branchings, angular ordering (coherence effects) is imposed.
- Gluon helicity effects, i.e. correlations between the production plane and the decay plane of a gluon, can be included.
- The first-order α_s expression is used, with the Q^2 scale given by (an approximation to) the squared transverse momentum of a branching. The default $\Lambda_{\rm QCD}$, which should not be regarded as a proper $\Lambda_{\overline{\rm MS}}$, is 0.29 GeV.
- The parton shower is by default cut off at a mass scale of 1 GeV.

Let us now proceed with a more detailed description.

10.2.1 The choice of evolution variable

In the PYTHIA shower algorithm, the evolution variable Q^2 is associated with the squared mass of the branching parton, $Q^2 = m_a^2$ for a branching $a \to bc$. As a consequence, $t = \ln(Q^2/\Lambda^2) = \ln(m_a^2/\Lambda^2)$. This Q^2 choice is not unique, and indeed other programs have other definitions: HERWIG uses $Q^2 \approx m^2/(2z(1-z))$ [Mar88] and ARIADNE $Q^2 = p_\perp^2 \approx z(1-z)m^2$ [Pet88]. Below we will also modify the Q^2 choice to give a better account of mass effects, e.g. for b quarks.

With Q a mass scale, the lower cut-off Q_0 is one in mass. To be more precise, in a QCD shower, the Q_0 parameter is used to derive effective masses

$$\begin{aligned} m_{\rm eff,g} &= \frac{1}{2} Q_0 \ , \\ m_{\rm eff,q} &= \sqrt{m_{\rm q}^2 + \frac{1}{4} Q_0^2} \ , \end{aligned} \tag{153}$$

where the $m_{\rm q}$ have been chosen as typical kinematical quark masses, see section 13.2.1. A parton cannot branch unless its mass is at least the sum of the lightest pair of allowed decay products, i.e. the minimum mass scale at which a branching is possible is

$$\begin{aligned} m_{\rm min,g} &= 2\, m_{\rm eff,g} = Q_0 \ , \\ m_{\rm min,q} &= m_{\rm eff,q} + m_{\rm eff,g} \geq Q_0 \ . \end{aligned} \tag{154}$$

The above masses are used to constrain the allowed range of Q^2 and z values. However, once it has been decided that a parton cannot branch any further, that parton is put on the mass shell, i.e. 'final-state' gluons are massless.

When also photon emission is included, a separate Q_0 scale is introduced for the QED part of the shower, and used to calculate cut-off masses by analogy with eqs. (153) and (154) above [Sjö92c]. By default the two Q_0 scales are chosen equal, and have the value 1 GeV. If anything, one would be inclined to allow a cut-off lower for photon emission than for gluon one. In that case the allowed z range of photon emission would be larger than that of gluon emission, and at the end of the shower evolution only photon emission would be allowed.

Photon and gluon emission differ fundamentally in that photons appear as physical particles in the final state, while gluons are confined. For photon emission off quarks, however, the confinement forces acting on the quark may provide an effective photon emission cut-off at larger scales than the bare quark mass. Soft and collinear photons could also be emitted by the final-state charged hadrons [Bar94a]; the matching between emission off quarks and off hadrons is a delicate issue, and we therefore do not attempt to address the soft-photon region.

For photon emission off leptons, there is no need to introduce any collinear emission cut-off beyond what is given by the lepton mass, but we keep the same cut-off approach as for quarks, although at a smaller scale. However, note that, firstly, the program is not aimed at high-precision studies of lepton pairs (where interference terms between initial- and final-state radiation also would have to be included), and, secondly, most experimental procedures would include the energy of collinear photons into the effective energy of a final-state lepton.

10.2.2 The choice of energy splitting variable

The final-state radiation machinery is always applied in the c.m. frame of the hard scattering, from which normally emerges a pair of evolving partons. Occasionally there may be one evolving parton recoiling against a non-evolving one, as in $q\overline{q} \to g\gamma$, where only the gluon evolves in the final state, but where the energy of the photon is modified by the branching activity of the gluon. (With only one evolving parton and nothing else, it would not be possible to conserve energy and momentum when the parton is assigned a mass.) Thus, before the evolution is performed, the parton pair is boosted to their common c.m. frame, and rotated to sit along the z axis. After the evolution, the full parton shower is rotated and boosted back to the original frame of the parton pair.

The interpretation of the energy and momentum splitting variable z is not unique, and in fact the program allows the possibility to switch between four different alternatives [Ben87a], 'local' and 'global' z definition combined with 'constrained' or 'unconstrained' evolution. In all four of them, the z variable is interpreted as an energy fraction, i.e. $E_b = zE_a$ and $E_c = (1-z)E_a$. In the 'local' choice of z definition, energy fractions are defined in the rest frame of the grandmother, i.e. the mother of parton a. The preferred choice is the 'global' one, in which energies are always evaluated in the c.m. frame of the hard scattering. The two definitions agree for the branchings of the partons that emerge directly from the hard scattering, since the hard scattering itself is considered to be the 'mother' of the first generation of partons. For instance, in $Z^0 \to q\overline{q}$ the Z^0 is considered the mother of the q and $\overline{q}$, even though the branching is not handled by the parton-showering machinery. The 'local' and 'global' definitions diverge for subsequent branchings, where the 'global' tends to allow more shower evolution.

In a branching $a \to bc$ the kinematically allowed range of $z = z_a$ values, $z_- < z < z_+$, is given by

$$z_\pm = \frac{1}{2}\left\{1 + \frac{m_b^2 - m_c^2}{m_a^2} \pm \frac{|\mathbf{p}_a|}{E_a}\frac{\sqrt{(m_a^2 - m_b^2 - m_c^2)^2 - 4m_b^2 m_c^2}}{m_a^2}\right\} . \tag{155}$$

With 'constrained' evolution, these bounds are respected in the evolution. The cut-off masses $m_{\mathrm{eff},b}$ and $m_{\mathrm{eff},c}$ are used to define the maximum allowed z range, within which z_a is chosen, together with the m_a value. In the subsequent evolution of b and c, only pairs of m_b and m_c are allowed for which the already selected z_a fulfils the constraints in eq. (155).

For 'unconstrained' evolution, which is the preferred alternative, one may start off by assuming the daughters to be massless, so that the allowed z range is

$$z_\pm = \frac{1}{2}\left\{1 \pm \frac{|\mathbf{p}_a|}{E_a}\theta(m_a - m_{\mathrm{min},a})\right\} , \tag{156}$$

where $\theta(x)$ is the step function, $\theta(x) = 1$ for $x > 0$ and $\theta(x) = 0$ for $x < 0$. The decay kinematics into two massless four-vectors $p_b^{(0)}$ and $p_c^{(0)}$ is now straightforward. Once m_b and m_c have been found from the subsequent evolution, subject only to the constraints $m_b < z_a E_a$, $m_c < (1-z_a)E_a$ and $m_b + m_c < m_a$, the actual massive four-vectors may be defined as

$$p_{b,c} = p_{b,c}^{(0)} \pm (r_c p_c^{(0)} - r_b p_b^{(0)}) , \tag{157}$$

where

$$r_{b,c} = \frac{m_a^2 \pm (m_c^2 - m_b^2) - \sqrt{(m_a^2 - m_b^2 - m_c^2)^2 - 4m_b^2 m_c^2}}{2m_a^2} . \tag{158}$$

In other words, the meaning of z_a is somewhat reinterpreted *post facto*. Needless to say, the 'unconstrained' option allows more branchings to take place than the 'constrained' one. In the following discussion we will only refer to the 'global, unconstrained' z choice.

10.2.3 First branchings and matrix-element matching

The final-state evolution is normally started from some initial parton pair $1+2$, at a Q^2_{max} scale determined by deliberations already discussed. When the evolution of parton 1 is considered, it is assumed that parton 2 is massless, so that the parton 1 energy and momentum are simple functions of its mass (and of the c.m. energy of the pair, which is fixed), and hence also the allowed z_1 range for splittings is a function of this mass, eq. (156). Correspondingly, parton 2 is evolved under the assumption that parton 1 is massless. After both partons have been assigned masses, their correct energies may be found, which are smaller than originally assumed. Therefore the allowed z ranges have shrunk, and it may happen that a branching has been assigned a z value outside this range. If so, the parton is evolved downwards in mass from the rejected mass value; if both z values are rejected, the parton with largest mass is evolved further. It may also happen that the sum of m_1 and m_2 is larger than the c.m. energy, in which case the one with the larger mass is evolved downwards. The checking and evolution steps are iterated until an acceptable set of m_1, m_2, z_1 and z_2 has been found.

The procedure is an extension of the veto algorithm, where an initial overestimation of the allowed z range is compensated by rejection of some branchings. One should note, however, that the veto algorithm is not strictly applicable for the coupled evolution in two variables (m_1 and m_2), and that therefore some arbitrariness is involved. This is manifest in the choice of which parton will be evolved further if both z values are unacceptable, or if the mass sum is too large.

For quark and lepton pairs which come from the decay of a colour-singlet particle, the first branchings are matched to the explicit first-order matrix elements for gauge boson decays.

The matching is based on a mapping of the parton-shower variables on to the 3-jet phase space. To produce a 3-jet event, $\gamma^*/\mathrm{Z}^0 \to \mathrm{q}(p_1)\overline{\mathrm{q}}(p_2)\mathrm{g}(p_3)$, in the shower language, one will pass through an intermediate state, where either the q or the $\overline{\mathrm{q}}$ is off the mass shell. If the former is the case then

$$\begin{aligned} m^2 &= (p_1+p_3)^2 = E^2_{\mathrm{cm}}(1-x_2)\ , \\ z &= \frac{E_1}{E_1+E_3} = \frac{x_1}{x_1+x_3} = \frac{x_1}{2-x_2}\ , \end{aligned} \tag{159}$$

where $x_i = 2E_i/E_{\mathrm{cm}}$. The $\overline{\mathrm{q}}$ emission case is obtained with $1 \leftrightarrow 2$. The parton-shower splitting expression in terms of m^2 and z, eq. (148), can therefore be translated into the following differential 3-jet rate:

$$\begin{aligned} \frac{1}{\sigma}\frac{\mathrm{d}\sigma_{\mathrm{PS}}}{\mathrm{d}x_1\,\mathrm{d}x_2} &= \frac{\alpha_{\mathrm{s}}}{2\pi} C_F \frac{1}{(1-x_1)(1-x_2)} \times \\ &\times \left\{ \frac{1-x_1}{x_3}\left(1+\left(\frac{x_1}{2-x_2}\right)^2\right) + \frac{1-x_2}{x_3}\left(1+\left(\frac{x_2}{2-x_1}\right)^2\right)\right\}\ , \end{aligned} \tag{160}$$

where the first term inside the curly bracket comes from emission off the quark and the second term from emission off the antiquark. The corresponding expression in matrix-element language is

$$\frac{1}{\sigma}\frac{\mathrm{d}\sigma_{\mathrm{ME}}}{\mathrm{d}x_1\,\mathrm{d}x_2} = \frac{\alpha_{\mathrm{s}}}{2\pi} C_F \frac{1}{(1-x_1)(1-x_2)}\left\{x_1^2+x_2^2\right\}\ . \tag{161}$$

With the kinematics choice of PYTHIA, the matrix-element expression is always smaller than the parton-shower one. It is therefore possible to run the shower as usual, but to impose an extra weight factor $\mathrm{d}\sigma_{\mathrm{ME}}/\mathrm{d}\sigma_{\mathrm{PS}}$, which is just the ratio of the expressions in curly brackets. If a branching is rejected, the evolution is continued from the rejected Q^2 value onwards (the veto algorithm). The weighting procedure is applied to the first branching of both the q and the $\overline{\mathrm{q}}$, in each case with the (nominal) assumption that none of the other partons branch (neither the sister nor the daughters), so that the relations of eq. (159) are applicable.

If a photon is emitted instead of a gluon, the emission rate in parton showers is given by

$$\frac{1}{\sigma}\frac{\mathrm{d}\sigma_{\mathrm{PS}}}{\mathrm{d}x_1\,\mathrm{d}x_2} = \frac{\alpha_{\mathrm{em}}}{2\pi}\frac{1}{(1-x_1)(1-x_2)}\times \times \left\{ e_{\mathrm{q}}^2\frac{1-x_1}{x_3}\left(1+\left(\frac{x_1}{2-x_2}\right)^2\right) + e_{\overline{\mathrm{q}}}^2\frac{1-x_2}{x_3}\left(1+\left(\frac{x_2}{2-x_1}\right)^2\right)\right\}\,, \quad (162)$$

and in matrix elements by [Gro81]

$$\frac{1}{\sigma}\frac{\mathrm{d}\sigma_{\mathrm{ME}}}{\mathrm{d}x_1\,\mathrm{d}x_2} = \frac{\alpha_{\mathrm{em}}}{2\pi}\frac{1}{(1-x_1)(1-x_2)}\left\{\left(e_{\mathrm{q}}\frac{1-x_1}{x_3} - e_{\overline{\mathrm{q}}}\frac{1-x_2}{x_3}\right)^2\left(x_1^2+x_2^2\right)\right\}\,. \quad (163)$$

As in the gluon emission case, a weighting factor $\mathrm{d}\sigma_{\mathrm{ME}}/\mathrm{d}\sigma_{\mathrm{PS}}$ can therefore be applied when either the original q (ℓ) or the original $\overline{\mathrm{q}}$ ($\overline{\ell}$) emits a photon. For a neutral resonance, such as Z^0, where $e_{\overline{\mathrm{q}}} = -e_{\mathrm{q}}$, the above expressions simplify and one recovers exactly the same ratio $\mathrm{d}\sigma_{\mathrm{ME}}/\mathrm{d}\sigma_{\mathrm{PS}}$ as for gluon emission.

Compared with the standard matrix-element treatment, a few differences remain. The shower one automatically contains the Sudakov form factor and an α_{s} running as a function of the $p_\perp^2$ scale of the branching. The shower also allows all partons to evolve further, which means that the naïve kinematics assumed for a comparison with matrix elements is modified by subsequent branchings, e.g. that the energy of parton 1 is reduced when parton 2 is assigned a mass. All these effects are formally of higher order, and so do not affect a first-order comparison. This does not mean that the corrections need be small, but experimental results are encouraging: the approach outlined does as good as explicit second-order matrix elements for the description of 4-jet production, better in some respects (like overall rate) and worse in others (like some angular distributions).

10.2.4 Subsequent branches and angular ordering

The shower evolution is (almost) always done on a pair of partons, so that energy and momentum can be conserved. In the first step of the evolution, the two original partons thus undergo branchings $1 \to 3+4$ and $2 \to 5+6$. As described above, the allowed m_1, m_2, z_1 and z_2 ranges are coupled by kinematical constraints. In the second step, the pair $3+4$ is evolved and, separately, the pair $5+6$. Considering only the former (the latter is trivially obtained by symmetry), the partons thus have nominal initial energies $E_3^{(0)} = z_1E_1$ and $E_4^{(0)} = (1-z_1)E_1$, and maximum allowed virtualities $m_{\mathrm{max},3} = \min(m_1, E_3^{(0)})$ and $m_{\mathrm{max},4} = \min(m_1, E_4^{(0)})$. Initially partons 3 and 4 are evolved separately, giving masses m_3 and m_4 and splitting variables z_3 and z_4. If $m_3 + m_4 > m_1$, the parton of 3 and 4 that has the largest ratio of $m_i/m_{\mathrm{max},i}$ is evolved further. Thereafter eq. (157) is used to construct corrected energies E_3 and E_4, and the z values are checked for consistency. If a branching has to be rejected because the change of parton energy puts z outside the allowed range, the parton is evolved further.

This procedure can then be iterated for the evolution of the two daughters of parton 3 and for the two of parton 4, etc., until each parton reaches the cut-off mass m_{min}. Then the parton is put on the mass shell.

The model, as described so far, produces so-called conventional showers, wherein masses are strictly decreasing in the shower evolution. Emission angles are decreasing only in an average sense, however, which means that also fairly 'late' branchings can give partons at large angles. Theoretical studies beyond the leading-log level show that this is not correct [Mue81], but that destructive interference effects are large in the region of non-ordered emission angles. To a good first approximation, these so-called coherence effects can be taken into account in parton shower programs by requiring a strict ordering in terms of decreasing emission angles.

(Actually, the fact that the shower described here is already ordered in mass implies that the additional cut on angle will be a bit too restrictive. While effects from this should be small at current energies, some deviations become visible at very high energies.)

The coherence phenomenon is known already from QED. One manifestation is the Chudakov effect [Chu55], discovered in the study of high-energy cosmic γ rays impinging on a nuclear target. If a γ is converted into a highly collinear e^+e^- pair inside the emulsion, the e^+ and e^- in their travel through the emulsion ionize atoms and thereby produce blackening. However, near the conversion point the blackening is small: the e^+ and e^- then are still close together, so that an atom traversed by the pair does not resolve the individual charges of the e^+ and the e^-, but only feels a net charge close to zero. Only later, when the e^+ and e^- are separated by more than a typical atomic radius, are the two able to ionize independently of each other.

The situation is similar in QCD, but is further extended, since now also gluons carry colour. For example, in a branching $q_0 \to qg$ the q and g share the newly created pair of opposite colour–anticolour charges, and therefore the q and g cannot emit subsequent gluons incoherently. Again the net effect is to reduce the amount of soft gluon emission: since a soft gluon (emitted at large angles) corresponds to a large (transverse) wavelength, the soft gluon is unable to resolve the separate colour charges of the q and the g, and only feels the net charge carried by the q_0. Such a soft gluon g' (in the region $\theta_{q_0 g'} > \theta_{qg}$) could therefore be thought of as being emitted by the q_0 rather than by the q–g system. If one considers only emission that should be associated with the q or the g, to a good approximation, there is a complete destructive interference in the regions of non-decreasing opening angles, while partons radiate independently of each other inside the regions of decreasing opening angles ($\theta_{qg'} < \theta_{qg}$ and $\theta_{gg'} < \theta_{qg}$), once azimuthal angles are averaged over. The details of the colour interference pattern are reflected in non-uniform azimuthal emission probabilities.

The first branchings of the shower are not affected by the angular-ordering requirement — since the evolution is performed in the c.m. frame of the original parton pair, where the original opening angle is 180°, any angle would anyway be smaller than this — but here instead the matrix-element matching procedure is used, where applicable. Subsequently, each opening angle is compared with that of the preceding branching in the shower.

For a branching $a \to bc$ the kinematical approximation

$$\theta_a \approx \frac{p_{\perp b}}{E_b} + \frac{p_{\perp c}}{E_c} \approx \sqrt{z_a(1-z_a)}\, m_a \left(\frac{1}{z_a E_a} + \frac{1}{(1-z_a)E_a} \right) = \frac{1}{\sqrt{z_a(1-z_a)}} \frac{m_a}{E_a} \tag{164}$$

is used to derive the opening angle (this is anyway to the same level of approximation as the one in which angular ordering is derived). With θ_b of the b branching calculated similarly, the requirement $\theta_b < \theta_a$ can be reduced to

$$\frac{z_b(1-z_b)}{m_b^2} > \frac{1-z_a}{z_a m_a^2} \; . \tag{165}$$

Since photons do not obey angular ordering, the check on angular ordering is not performed when a photon is emitted. When a gluon is emitted in the branching after a photon, its emission angle is restricted by that of the preceding QCD branching in the shower, i.e. the photon emission angle does not enter.

10.2.5 Other final-state shower aspects

The electromagnetic coupling constant for the emission of photons on the mass shell is $\alpha_{em} = \alpha_{em}(Q^2 = 0) \approx 1/137$. For the strong coupling constant several alternatives are available, the default being the first-order expression $\alpha_s(p_\perp^2)$, where $p_\perp^2$ is defined by the approximate expression $p_\perp^2 \approx z(1-z)m^2$. Studies of next-to-leading-order corrections favour this choice [Ama80]. The other alternatives are a fixed α_s and an $\alpha_s(m^2)$.

With the default choice of $p_\perp^2$ as scale in α_s, a further cut-off is introduced on the allowed phase space of gluon emission, not present in the options with fixed α_s or with $\alpha_s(m^2)$, nor in the QED shower. A minimum requirement, to ensure a well-defined α_s, is that $p_\perp/\Lambda > 1.1$, but additionally PYTHIA requires that $p_\perp > Q_0/2$. This latter requirement is not a necessity, but it makes sense when $p_\perp$ is taken to be the preferred scale of the branching process, rather than e.g. m. It reduces the allowed z range, compared with the purely kinematical constraints. Since the $p_\perp$ cut is not present for photon emission, the relative ratio of photon to gluon emission off a quark is enhanced at small virtualities compared with naïve expectations; in actual fact this enhancement is largely compensated by the running of α_s, which acts in the opposite direction. The main consequence, however, is that the gluon energy spectrum is peaked at around Q_0 and rapidly vanishes for energies below that, whilst the photon spectrum extends all the way to zero energy.

Previously it was said that azimuthal angles in branchings are chosen isotropically. In fact, it is possible to include some effects of gluon polarization, which correlate the production and the decay planes of a gluon, such that a $g \to gg$ branching tends to take place in the production plane of the gluon, while a decay out of the plane is favoured for $g \to q\overline{q}$. The formulae are given e.g. in ref. [Web86], as simple functions of the z value at the vertex where the gluon is produced and of the z value when it branches. Also coherence phenomena lead to non-isotropic azimuthal distributions [Web86]. In either case the φ azimuthal variable is first chosen isotropically, then the weight factor due to polarization times coherence is evaluated, and the φ value is accepted or rejected. In case of rejection, a new φ is generated, and so on.

While the rule is to have an initial pair of partons, there are a few examples where one or three partons have to be allowed to shower. If only one parton is given, it is not possible to conserve both energy and momentum. The choice has been made to conserve energy and jet direction, but the momentum vector is scaled down when the radiating parton acquires a mass. The 'rest frame of the system', used e.g. in the z definition, is taken to be whatever frame the jet is given in.

In $\Upsilon \to ggg$ decays and other configurations (e.g. from external processes) with three or more primary parton, one is left with the issue how the kinematics from the on-shell matrix elements should be reinterpreted for an off-shell multi-parton configuration. We have made the arbitrary choice of preserving the direction of motion of each parton in the rest frame of the system, which means that all three-momenta are scaled down by the same amount, and that some particles gain energy at the expense of others. Mass multiplets outside the allowed phase space are rejected and the evolution continued.

Finally, it should be noted that two toy shower models are included as options. One is a scalar gluon model, in which the $q \to qg$ branching kernel is replaced by $P_{q\to qg}(z) = \frac{2}{3}(1-z)$. The couplings of the gluon, $g \to gg$ and $g \to q\overline{q}$, have been left as free parameters, since they depend on the colour structure assumed in the model. The spectra are flat in z for a spin 0 gluon. Higher-order couplings of the type $g \to ggg$ could well contribute significantly, but are not included. The second toy model is an Abelian vector one. In this option $g \to gg$ branchings are absent, and $g \to q\overline{q}$ ones enhanced. More precisely, in the splitting kernels, eq. (149), the Casimir factors are changed as follows: $C_F = 4/3 \to 1$, $N_C = 3 \to 0$, $T_R = n_f/2 \to 3n_f$. When using either of these options, one should be aware that also a number of other components in principle should be changed, from the running of α_s to the whole concept of fragmentation. One should therefore not take them too seriously.

10.2.6 Merging with massive matrix elements

The matching to first-order matrix-elements is well-defined for massless quarks, and was originally used unchanged for massive ones. A first attempt to include massive matrix elements did not compensate for mass effects in the shower kinematics, and therefore came to exaggerate the suppression of radiation off heavy quarks [Nor01, Bam00]. Now the shower has been modified to solve this issue, and also improved and extended to cover better a host of different reactions

Table 25: The processes that have been calculated, also with one extra gluon in the final state. Colour is given with 1 for singlet, 3 for triplet and 8 for octet. See the text for an explanation of the γ_5 column and further comments.

colour	spin	γ_5	example	codes
$1 \to 3+\overline{3}$	—	—	(eikonal)	6 – 9
$1 \to 3+\overline{3}$	$1 \to \frac{1}{2}+\frac{1}{2}$	$1, \gamma_5, 1\pm\gamma_5$	$\mathrm{Z}^0 \to \mathrm{q}\overline{\mathrm{q}}$	11 – 14
$3 \to 3+1$	$\frac{1}{2} \to \frac{1}{2}+1$	$1, \gamma_5, 1\pm\gamma_5$	$\mathrm{t} \to \mathrm{bW}^+$	16 – 19
$1 \to 3+\overline{3}$	$0 \to \frac{1}{2}+\frac{1}{2}$	$1, \gamma_5, 1\pm\gamma_5$	$\mathrm{h}^0 \to \mathrm{q}\overline{\mathrm{q}}$	21 – 24
$3 \to 3+1$	$\frac{1}{2} \to \frac{1}{2}+0$	$1, \gamma_5, 1\pm\gamma_5$	$\mathrm{t} \to \mathrm{bH}^+$	26 – 29
$1 \to 3+\overline{3}$	$1 \to 0+0$	1	$\mathrm{Z}^0 \to \tilde{\mathrm{q}}\overline{\tilde{\mathrm{q}}}$	31 – 34
$3 \to 3+1$	$0 \to 0+1$	1	$\tilde{\mathrm{q}} \to \tilde{\mathrm{q}}'\mathrm{W}^+$	36 – 39
$1 \to 3+\overline{3}$	$0 \to 0+0$	1	$\mathrm{h}^0 \to \tilde{\mathrm{q}}\overline{\tilde{\mathrm{q}}}$	41 – 44
$3 \to 3+1$	$0 \to 0+0$	1	$\tilde{\mathrm{q}} \to \tilde{\mathrm{q}}'\mathrm{H}^+$	46 – 49
$1 \to 3+\overline{3}$	$\frac{1}{2} \to \frac{1}{2}+0$	$1, \gamma_5, 1\pm\gamma_5$	$\tilde{\chi} \to \mathrm{q}\overline{\tilde{\mathrm{q}}}$	51 – 54
$3 \to 3+1$	$0 \to \frac{1}{2}+\frac{1}{2}$	$1, \gamma_5, 1\pm\gamma_5$	$\tilde{\mathrm{q}} \to \mathrm{q}\tilde{\chi}$	56 – 59
$3 \to 3+1$	$\frac{1}{2} \to 0+\frac{1}{2}$	$1, \gamma_5, 1\pm\gamma_5$	$\mathrm{t} \to \tilde{\mathrm{t}}\tilde{\chi}$	61 – 64
$8 \to 3+\overline{3}$	$\frac{1}{2} \to \frac{1}{2}+0$	$1, \gamma_5, 1\pm\gamma_5$	$\tilde{\mathrm{g}} \to \mathrm{q}\overline{\tilde{\mathrm{q}}}$	66 – 69
$3 \to 3+8$	$0 \to \frac{1}{2}+\frac{1}{2}$	$1, \gamma_5, 1\pm\gamma_5$	$\tilde{\mathrm{q}} \to \mathrm{q}\tilde{\mathrm{g}}$	71 – 74
$3 \to 3+8$	$\frac{1}{2} \to 0+\frac{1}{2}$	$1, \gamma_5, 1\pm\gamma_5$	$\mathrm{t} \to \tilde{\mathrm{t}}\tilde{\mathrm{g}}$	76 – 79
$1 \to 8+8$	—	—	(eikonal)	81 – 84

[Nor01].

The starting point is the calculation of processes $a \to bc$ and $a \to bc\mathrm{g}$, where the ratio

$$W_{\mathrm{ME}}(x_1, x_2) = \frac{1}{\sigma(a \to bc)} \frac{\mathrm{d}\sigma(a \to bc\mathrm{g})}{\mathrm{d}x_1\,\mathrm{d}x_2} \tag{166}$$

gives the process-dependent differential gluon-emission rate. Here the phase space variables are $x_1 = 2E_b/m_a$ and $x_2 = 2E_c/m_a$, expressed in the rest frame of particle a. Using the standard model and the minimal supersymmetric extension thereof as templates, a wide selection of colour and spin structures have been addressed, as shown in Table 25. When allowed, processes have been calculated for an arbitrary mixture of 'parities', i.e. without or with a γ_5 factor, like in the vector/axial vector structure of γ^*/Z^0. Various combinations of 1 and γ_5 may also arise e.g. from the wave functions of the sfermion partners to the left- and right-handed fermion states. In cases where the correct combination is not provided, an equal mixture of the two is assumed as a reasonable compromize. All the matrix elements are encoded in the new function `PYMAEL(NI,X1,X2,R1,R2,ALPHA)`, where `NI` distinguishes the matrix elements, `ALPHA` is related to the γ_5 admixture and the mass ratios $r_1 = m_b/m_a$ and $r_2 = m_c/m_a$ are free parameters. This routine is called by `PYSHOW`, but might also have an interest on its own.

In order to match to the singularity structure of the massive matrix elements, the evolution variable Q^2 is changed from m^2 to $m^2 - m^2_{\mathrm{on-shell}}$, i.e. $1/Q^2$ is the propagator of a massive

particle [Nor01]. For the shower history $b \to b\mathrm{g}$ this gives a differential probability

$$W_{\mathrm{PS},1}(x_1,x_2) = \frac{\alpha_\mathrm{s}}{2\pi} C_F \frac{\mathrm{d}Q^2}{Q^2} \frac{2\,\mathrm{d}z}{1-z} \frac{1}{\mathrm{d}x_1\,\mathrm{d}x_2} = \frac{\alpha_\mathrm{s}}{2\pi} C_F \frac{2}{x_3\,(1+r_2^2-r_1^2-x_2)}\,, \tag{167}$$

where the numerator $1+z^2$ of the splitting kernel for $\mathrm{q} \to \mathrm{qg}$ has been replaced by a 2 in the shower algorithm. For a process with only one radiating parton in the final state, such as $\mathrm{t} \to \mathrm{bW}^+$, the ratio $W_\mathrm{ME}/W_{\mathrm{PS},1}$ gives the acceptance probability for an emission in the shower. The singularity structure exactly agrees between ME and PS, giving a well-behaved ratio always below unity. If both b and c can radiate, there is a second possible shower history that has to be considered. The matrix element is here split in two parts, one arbitrarily associated with $b \to b\mathrm{g}$ branchings and the other with $c \to c\mathrm{g}$ ones. A convenient choice is $W_{\mathrm{ME},1} = W_\mathrm{ME}(1+r_1^2-r_2^2-x_1)/x_3$ and $W_{\mathrm{ME},2} = W_\mathrm{ME}(1+r_2^2-r_1^2-x_2)/x_3$, which again gives matching singularity structures in $W_{\mathrm{ME},i}/W_{\mathrm{PS},i}$ and thus a well-behaved Monte Carlo procedure.

Also subsequent emissions of gluons off the primary particles are corrected to W_ME. To this end, a reduced-energy system is constructed, which retains the kinematics of the branching under consideration but omits the gluons already emitted, so that an effective three-body shower state can be mapped to an (x_1,x_2,r_1,r_2) set of variables. For light quarks this procedure is almost equivalent with the original one of using the simple universal splitting kernels after the first branching. For heavy quarks it offers an improved modelling of mass effects also in the collinear region.

Some further changes have been introduced, a few minor as default and some more significant ones as non-default options [Nor01]. This includes the description of coherence effects and α_s arguments, in general and more specifically for secondary heavy flavour production by gluon splittings. The problem in the latter area is that data at LEP1 show a larger rate of secondary charm and bottom production than predicted in most shower descriptions [Bam00, Man00], or in analytical studies [Sey95]. This is based on applying the same kind of coherence considerations to $\mathrm{g} \to \mathrm{q}\overline{\mathrm{q}}$ branchings as to $\mathrm{g} \to \mathrm{gg}$, which is not fully motivated by theory. In the lack of an unambiguous answer, it is therefore helpful to allow options that can explore the range of uncertainty.

Further issues remain to be addressed, e.g. radiation off particles with non-negligible width. In general, however, the new shower should allow an improved description of gluon radiation in many different processes.

10.2.7 Matching to four-parton events

The shower routine, as described above, is optimized for two objects forming the showering system, within which energy and momentum should be conserved. However, occasionally more than two initial objects are given, e.g. if one would like to consider the subclass of $\mathrm{e}^+\mathrm{e}^- \to \mathrm{q}\overline{\mathrm{q}}\mathrm{gg}$ events in order to study angular correlations as a test of the coupling structure of QCD. Such events are generated in the showering of normal $\mathrm{e}^+\mathrm{e}^- \to \mathrm{q}\overline{\mathrm{q}}$ events, but not with high efficiency within desired cuts, and not with the full angular structure included in the shower. Therefore four-parton matrix elements may be the required starting point but, in order to 'dress up' these partons, one nevertheless wishes to add shower emission. A possibility to start from three partons has existed since long, but only with [And98a] was an approach for four parton introduced, and with the possibility to generalize to more partons, although this latter work has not yet been done.

The basic idea is to cast the output of matrix element generators in the form of a parton-shower history, that then can be used as input for a complete parton shower. In the shower, that normally would be allowed to develop at random, some branchings are now fixed to their matrix-element values while the others are still allowed to evolve in the normal shower fashion. The preceding history of the event is also in these random branchings then reflected e.g. in terms of kinematical or dynamical (e.g. angular ordering) constraints.

Consider e.g. the $q\bar{q}gg$ case. The matrix-element expression contains contributions from five graphs, and from interferences between them. The five graphs can also be read as five possible parton-shower histories for arriving at the same four-parton state, but here without the possibility of including interferences. The relative probability for each of these possible shower histories can be obtained from the rules of shower branchings. For example, the relative probability for the history where $e^+e^- \to q(1)\bar{q}(2)$, followed by $q(1) \to q(3)g(4)$ and $g(4) \to g(5)g(6)$, is given by:

$$\mathcal{P} = \mathcal{P}_{1\to 34}\mathcal{P}_{4\to 56} = \frac{1}{m_1^2}\frac{4}{3}\frac{1+z_{34}^2}{1-z_{34}} \cdot \frac{1}{m_4^2} 3 \frac{(1-z_{56}(1-z_{56}))^2}{z_{56}(1-z_{56})} \tag{168}$$

where the probability for each branching contains the mass singularity, the colour factor and the momentum splitting kernel. The masses are given by

$$\begin{aligned} m_1^2 = p_1^2 &= (p_3+p_5+p_6)^2 \,, \\ m_4^2 = p_4^2 &= (p_5+p_6)^2 \,, \end{aligned} \tag{169}$$

and the z values by

$$\begin{aligned} z_{bc} = z_{a\to bc} &= \frac{m_a^2}{\lambda}\frac{E_b}{E_a} - \frac{m_a^2-\lambda+m_b^2-m_c^2}{2\lambda} \\ \text{with } \lambda &= \sqrt{(m_a^2-m_b^2-m_c^2)^2 - 4m_b^2 m_c^2} \,. \end{aligned} \tag{170}$$

We here assume that the on-shell mass of quarks can be neglected. The form of the probability then matches the expression used in the parton-shower algorithm.

Variants on the above probabilities are imaginable. For instance, in the spirit of the matrix-element approach we have assumed a common α_s for all graphs, which thus need not be shown, whereas the parton-shower language normally assumes α_s to be a function of the transverse momentum of each branching. One could also include information on azimuthal anisotropies.

The relative probability $\mathcal{P}$ for each of the five possible parton-shower histories can be used to select one of the possibilities at random. (A less appealing alternative would be a 'winner takes all' strategy, i.e. selecting the configuration with the largest $\mathcal{P}$.) The selection fixes the values of the m, z and φ at two vertices. The azimuthal angle φ is defined by the daughter parton orientation around the mother direction. When the conventional parton-shower algorithm is executed, these values are then forced on the otherwise random evolution. This forcing cannot be exact for the z values, since the final partons given by the matrix elements are on the mass shell, while the corresponding partons in the parton shower might be virtual and branch further. The shift between the wanted and the obtained z values are rather small, very seldom above 10^{-6}. More significant are the changes of the opening angle between two daughters: when daughters originally assumed massless are given a mass the angle between them tends to be reduced. This shift has a non-negligible tail even above 0.1 radians. The 'narrowing' of jets by this mechanism is compensated by the broadening caused by the decay of the massive daughters, and thus overall effects are not so dramatic.

All other branchings of the parton shower are selected at random according to the standard evolution scheme. There is an upper limit on the non-forced masses from internal logic, however. For instance, for four-parton matrix elements, the singular regions are typically avoided with a cut $y > 0.01$, where y is the square of the minimal scaled invariant mass between any pair of partons. Larger y values could be used for some purposes, while smaller ones give so large four-jet rates that the need to include Sudakov form factors can no longer be neglected. The $y > 0.01$ cut roughly corresponds to $m > 9$ GeV at LEP 1 energies, so the hybrid approach must allow branchings at least below 9 GeV in order to account for the emission missing from the

matrix-element part. Since no 5-parton emission is generated by the second-order matrix elements, one could also allow a threshold higher than 9 GeV in order to account for this potential emission. However, if any such mass is larger than one of the forced masses, the result would be a different history than the chosen one, and one would risk some doublecounting issues. So, as an alternative, one could set the minimum invariant mass between any of the four original partons as the maximum scale of the subsequent shower evolution.

10.3 Initial-State Showers

The initial-state shower algorithm in PYTHIA is not quite as sophisticated as the final-state one. This is partly because initial-state radiation is less well understood theoretically, partly because the programming task is more complicated and ambiguous. Still, the program at disposal is known to do a reasonably good job of describing existing data, such as Z^0 production properties at hadron colliders [Sjö85]. It can be used both for QCD showers and for photon emission off leptons (e, μ or τ; relative to earlier versions, the description of incoming μ and τ are better geared to represent the differences in lepton mass, and the lepton-inside-lepton parton distributions are properly defined).

10.3.1 The shower structure

A fast hadron may be viewed as a cloud of quasi-real partons. Similarly a fast lepton may be viewed as surrounded by a cloud of photons and partons; in the program the two situations are on an equal footing, but here we choose the hadron as example. At each instant, an individual parton can initiate a virtual cascade, branching into a number of partons. This cascade can be described in terms of a tree-like structure, composed of many subsequent branchings $a \to bc$. Each branching involves some relative transverse momentum between the two daughters. In a language where four-momentum is conserved at each vertex, this implies that at least one of the b and c partons must have a space-like virtuality, $m^2 < 0$. Since the partons are not on the mass shell, the cascade only lives a finite time before reassembling, with those parts of the cascade that are most off the mass shell living the shortest time.

A hard scattering, e.g. in deeply inelastic leptoproduction, will probe the hadron at a given instant. The probe, i.e. the virtual photon in the leptoproduction case, is able to resolve fluctuations in the hadron up to the Q^2 scale of the hard scattering. Thus probes at different Q^2 values will seem to see different parton compositions in the hadron. The change in parton composition with $t = \ln(Q^2/\Lambda^2)$ is given by the evolution equations

$$\frac{\mathrm{d}f_b(x,t)}{\mathrm{d}t} = \sum_{a,c} \int \frac{\mathrm{d}x'}{x'}\, f_a(x',t)\, \frac{\alpha_{abc}}{2\pi}\, P_{a\to bc}\left(\frac{x}{x'}\right) \ . \tag{171}$$

Here the $f_i(x,t)$ are the parton-distribution functions, expressing the probability of finding a parton i carrying a fraction x of the total momentum if the hadron is probed at virtuality Q^2. The $P_{a\to bc}(z)$ are given in eq. (149). As before, α_{abc} is α_s for QCD shower and α_{em} for QED ones.

Eq. (171) is closely related to eq. (148): $\mathrm{d}\mathcal{P}_a$ describes the probability that a given parton a will branch (into partons b and c), $\mathrm{d}f_b$ the influx of partons b from the branchings of partons a. (The expression $\mathrm{d}f_b$ in principle also should contain a loss term for partons b that branch; this term is important for parton-distribution evolution, but does not appear explicitly in what we shall be using eq. (171) for.) The absolute form of hadron parton distributions cannot be predicted in perturbative QCD, but rather have to be parameterized at some Q_0 scale, with the Q^2 dependence thereafter given by eq. (171). Available parameterizations are discussed in section 7.1. The lepton and photon parton distributions inside a lepton can be fully predicted, but here for simplicity are treated on equal footing with hadron parton distributions.

If a hard interaction scatters a parton out of the incoming hadron, the 'coherence' [Gri83] of the cascade is broken: the partons can no longer reassemble completely back to the cascade-initiating parton. In this semiclassical picture, the partons on the 'main chain' of consecutive branchings that lead directly from the initiating parton to the scattered parton can no longer reassemble, whereas fluctuations on the 'side branches' to this chain may still disappear. A convenient description is obtained by assigning a space-like virtuality to the partons on the main chain, in such a way that the partons on the side branches may still be on the mass shell. Since the momentum transfer of the hard process can put the scattered parton on the mass shell (or even give it a time-like virtuality, so that it can initiate a final-state shower), one is then guaranteed that no partons have a space-like virtuality in the final state. (In real life, confinement effects obviously imply that partons need not be quite on the mass shell.) If no hard scattering had taken place, the virtuality of the space-like parton line would still force the complete cascade to reassemble. Since the virtuality of the cascade probed is carried by one single parton, it is possible to equate the space-like virtuality of this parton with the Q^2 scale of the cascade, to be used e.g. in the evolution equations. Coherence effects [Gri83, Bas83] guarantee that the Q^2 values of the partons along the main chain are strictly ordered, with the largest Q^2 values close to the hard scattering.

Further coherence effects have been studied [Cia87], with particular implications for the structure of parton showers at small x. None of these additional complications are implemented in the current algorithm, with the exception of a few rather primitive options that do not address the full complexity of the problem.

Instead of having a tree-like structure, where all legs are treated democratically, the cascade is reduced to a single sequence of branchings $a \to bc$, where the a and b partons are on the main chain of space-like virtuality, $m^2_{a,b} < 0$, while the c partons are on the mass shell and do not branch. (Later we will include the possibility that the c partons may have positive virtualities, $m^2_c > 0$, which leads to the appearance of time-like 'final-state' parton showers on the side branches.) This truncation of the cascade is only possible when it is known which parton actually partakes in the hard scattering: of all the possible cascades that exist virtually in the incoming hadron, the hard scattering will select one.

To obtain the correct Q^2 evolution of parton distributions, e.g., it is essential that all branches of the cascade be treated democratically. In Monte Carlo simulation of space-like showers this is a major problem. If indeed the evolution of the complete cascade is to be followed from some small Q^2_0 up to the Q^2 scale of the hard scattering, it is not possible at the same time to handle kinematics exactly, since the virtuality of the various partons cannot be found until after the hard scattering has been selected. This kind of 'forward evolution' scheme therefore requires a number of extra tricks to be made to work. Further, in this approach it is not known e.g. what the $\hat{s}$ of the hard scattering subsystem will be until the evolution has been carried out, which means that the initial-state evolution and the hard scattering have to be selected jointly, a not so trivial task.

Instead we use the 'backwards evolution' approach [Sjö85], in which the hard scattering is first selected, and the parton shower that preceded it is subsequently reconstructed. This reconstruction is started at the hard interaction, at the $Q^2_{\max}$ scale, and thereafter step by step one moves 'backwards' in 'time', towards smaller Q^2, all the way back to the parton-shower initiator at the cut-off scale Q^2_0. This procedure is possible if evolved parton distributions are used to select the hard scattering, since the $f_i(x, Q^2)$ contain the inclusive summation of all initial-state parton-shower histories that can lead to the appearance of an interacting parton i at the hard scale. What remains is thus to select an exclusive history from the set of inclusive ones.

10.3.2 Longitudinal evolution

The evolution equations, eq. (171), express that, during a small increase dt there is a probability for parton a with momentum fraction x' to become resolved into parton b at $x = zx'$ and another parton c at $x' - x = (1-z)x'$. Correspondingly, in backwards evolution, during a decrease dt a

parton b may be 'unresolved' into parton a. The relative probability $\mathrm{d}\mathcal{P}_b$ for this to happen is given by the ratio $\mathrm{d}f_b/f_b$. Using eq. (171) one obtains

$$\mathrm{d}\mathcal{P}_b = \frac{\mathrm{d}f_b(x,t)}{f_b(x,t)} = |\mathrm{d}t| \sum_{a,c} \int \frac{\mathrm{d}x'}{x'} \frac{f_a(x',t)}{f_b(x,t)} \frac{\alpha_{abc}}{2\pi} P_{a\to bc}\left(\frac{x}{x'}\right) . \tag{172}$$

Summing up the cumulative effect of many small changes $\mathrm{d}t$, the probability for no radiation exponentiates. Therefore one may define a form factor

$$\begin{aligned} S_b(x,t_{\max},t) &= \exp\left\{ -\int_t^{t_{\max}} \mathrm{d}t' \sum_{a,c} \int \frac{\mathrm{d}x'}{x'} \frac{f_a(x',t')}{f_b(x,t')} \frac{\alpha_{abc}(t')}{2\pi} P_{a\to bc}\left(\frac{x}{x'}\right) \right\} \\ &= \exp\left\{ -\int_t^{t_{\max}} \mathrm{d}t' \sum_{a,c} \int \mathrm{d}z \frac{\alpha_{abc}(t')}{2\pi} P_{a\to bc}(z) \frac{x' f_a(x',t')}{x f_b(x,t')} \right\} , \end{aligned} \tag{173}$$

giving the probability that a parton b remains at x from $t_{\max}$ to a $t < t_{\max}$.

It may be useful to compare this with the corresponding expression for forward evolution, i.e. with $S_a(t)$ in eq. (151). The most obvious difference is the appearance of parton distributions in S_b. Parton distributions are absent in S_a: the probability for a given parton a to branch, once it exists, is independent of the density of partons a or b. The parton distributions in S_b, on the other hand, express the fact that the probability for a parton b to come from the branching of a parton a is proportional to the number of partons a there are in the hadron, and inversely proportional to the number of partons b. Thus the numerator f_a in the exponential of S_b ensures that the parton composition of the hadron is properly reflected. As an example, when a gluon is chosen at the hard scattering and evolved backwards, this gluon is more likely to have been emitted by a u than by a d if the incoming hadron is a proton. Similarly, if a heavy flavour is chosen at the hard scattering, the denominator f_b will vanish at the Q^2 threshold of the heavy-flavour production, which means that the integrand diverges and S_b itself vanishes, so that no heavy flavour remain below threshold.

Another difference between S_b and S_a, already touched upon, is that the $P_{\mathrm{g}\to\mathrm{gg}}(z)$ splitting kernel appears with a normalization $2N_C$ in S_b but only with N_C in S_a, since two gluons are produced but only one decays in a branching.

A knowledge of S_b is enough to reconstruct the parton shower backwards. At each branching $a \to bc$, three quantities have to be found: the t value of the branching (which defines the space-like virtuality Q_b^2 of parton b), the parton flavour a and the splitting variable z. This information may be extracted as follows:

1. If parton b partook in the hard scattering or branched into other partons at a scale $t_{\max}$, the probability that b was produced in a branching $a \to bc$ at a lower scale t is

$$\frac{\mathrm{d}\mathcal{P}_b}{\mathrm{d}t} = -\frac{\mathrm{d}S_b(x,t_{\max},t)}{\mathrm{d}t} = \left(\sum_{a,c} \int dz \frac{\alpha_{abc}(t')}{2\pi} P_{a\to bc}(z) \frac{x' f_a(x',t')}{x f_b(x,t')} \right) S_b(x,t_{\max},t) . \tag{174}$$

   If no branching is found above the cut-off scale t_0 the iteration is stopped and parton b is assumed to be massless.

2. Given the t of a branching, the relative probabilities for the different allowed branchings $a \to bc$ are given by the z integrals above, i.e. by

$$\int \mathrm{d}z \frac{\alpha_{abc}(t)}{2\pi} P_{a\to bc}(z) \frac{x' f_a(x',t)}{x f_b(x,t)} . \tag{175}$$

3. Finally, with t and a known, the probability distribution in the splitting variable $z = x/x' = x_b/x_a$ is given by the integrand in eq. (175).

In addition, the azimuthal angle φ of the branching is selected isotropically, i.e. no spin or coherence effects are included in this distribution.

The selection of t, a and z is then a standard task of the kind than can be performed with the help of the veto algorithm. Specifically, upper and lower bounds for parton distributions are used to find simple functions that are everywhere larger than the integrands in eq. (175). Based on these simple expressions, the integration over z may be carried out, and t, a and z values selected. This set is then accepted with a weight given by a ratio of the correct integrand in eq. (175) to the simple approximation used, both evaluated for the given set. Since parton distributions, as a rule, are not in a simple analytical form, it may be tricky to find reasonably good bounds to parton distributions. It is necessary to make different assumptions for valence and sea quarks, and be especially attentive close to a flavour threshold ([Sjö85]). An electron distribution inside an electron behaves differently from parton distributions encountered in hadrons, and has to be considered separately.

A comment on soft gluon emission. Nominally the range of the z integral in S_b is $x \leq z \leq 1$. The lower limit corresponds to $x' = x/z = 1$, and parton distributions vanish in this limit, wherefore no problems are encountered here. At the upper cut-off $z = 1$ the splitting kernels $P_{\mathrm{q}\to\mathrm{qg}}(z)$ and $P_{\mathrm{g}\to\mathrm{gg}}$ diverge. This is the soft gluon singularity: the energy carried by the emitted gluon is vanishing, $x_{\mathrm{g}} = x' - x = (1-z)x' = (1-z)x/z \to 0$ for $z \to 1$. In order to calculate the integral over z in S_b, an upper cut-off $z_{\mathrm{max}} = x/(x + x_{\varepsilon})$ is introduced, i.e. only branchings with $z \leq z_{\mathrm{max}}$ are included in S_b. Here x_{ε} is a small number, typically chosen so that the gluon energy is above 2 GeV when calculated in the rest frame of the hard scattering. That is, the gluon energy $x_{\mathrm{g}}\sqrt{s}/2 \geq x_{\varepsilon}\sqrt{s}/2 = 2\ \mathrm{GeV}/\gamma$, where γ is the boost factor of the hard scattering. The average amount of energy carried away by gluons in the range $x_g < x_{\varepsilon}$, over the given range of t values from t_a to t_b, may be estimated [Sjö85]. The finally selected z value may thus be picked as $z = z_{\mathrm{hard}}\langle z_{\mathrm{soft}}(t_a, t_b)\rangle$, where z_{hard} is the originally selected z value and z_{soft} is the correction factor for soft gluon emission.

In QED showers, the smallness of α_{em} means that one can use rather smaller cut-off values without obtaining large amounts of emission. A fixed small cut-off $x_{\gamma} > 10^{-10}$ is therefore used to avoid the region of very soft photons. As has been discussed in section 7.1.3, the electron distribution inside the electron is cut off at $x_{\mathrm{e}} < 1 - 10^{-10}$, for numerical reasons, so the two cuts are closely matched.

The cut-off scale Q_0 may be chosen separately for QCD and QED showers, just as in final-state radiation. The defaults are 1 GeV and 0.001 GeV, respectively. The former is the typical hadronic mass scale, below which radiation is not expected resolvable; the latter is of the order of the electron mass.

Normally QED and QCD showers do not appear mixed. The most notable exception is resolved photoproduction (in ep) and resolved $\gamma\gamma$ events (in $\mathrm{e^+e^-}$), i.e. shower histories of the type $\mathrm{e} \to \gamma \to \mathrm{q}$. Here the Q^2 scales need not be ordered at the interface, i.e. the last $\mathrm{e} \to \mathrm{e}\gamma$ branching may well have a larger Q^2 than the first $\mathrm{q} \to \mathrm{qg}$ one, and the branching $\gamma \to \mathrm{q}$ does not even have a strict parton-shower interpretation for the vector dominance model part of the photon parton distribution. This kind of configurations is best described by the `'gamma/`*`lepton'`* machinery for having a flux of virtual photons inisde the lepton, see section 7.1.4. In this case, no initial-state radiation has currently been implemented for the electron (or μ or τ). The one inside the virtual-photon system is considered with the normal algorithm, but with the lower cut-off scale modified by the photon virtuality, see `MSTP(66)`.

An older description still lives on, although no longer as the recommended one. There, these issues are currently not addressed in full. Rather, based on the x selected for the parton (quark or gluon) at the hard scattering, the x_{γ} is selected once and for all in the range $x < x_{\gamma} < 1$, according to the distribution implied by eq. (54). The QCD parton shower is then traced backwards from the hard scattering to the QCD shower initiator at t_0. No attempt is made to perform the full QED shower, but rather the beam remnant treatment (see section 11.1) is used to find the $\overline{\mathrm{q}}$ (or g) remnant that matches the q (or g) QCD shower initiator, with the electron itself considered as a second beam remnant.

10.3.3 Transverse evolution

We have above seen that two parton lines may be defined, stretching back from the hard scattering to the initial incoming hadron wavefunctions at small Q^2. Specifically, all parton flavours i, virtualities Q^2 and energy fractions x may be found. The exact kinematical interpretation of the x variable is not unique, however. For partons with small virtualities and transverse momenta, essentially all definitions agree, but differences may appear for branchings close to the hard scattering.

In first-order QED [Ber85] and in some simple QCD toy models [Got86], one may show that the 'correct' choice is the '$\hat{s}$ approach'. Here one requires that $\hat{s} = x_1 x_2 s$, both at the hard scattering scale and at any lower scale, i.e. $\hat{s}(Q^2) = x_1(Q^2)\, x_2(Q^2)\, s$, where x_1 and x_2 are the x values of the two resolved partons (one from each incoming beam particle) at the given Q^2 scale. In practice this means that, at a branching with the splitting variable z, the total $\hat{s}$ has to be increased by a factor $1/z$ in the backwards evolution. It also means that branchings on the two incoming legs have to be interleaved in a single monotonic sequence of Q^2 values of branchings. A problem with this x interpretation is that it is not quite equivalent with an $\overline{\mathrm{MS}}$ definition of parton densities [Col00], or any other standard definition. In practice, effects should not be large from this mismatch.

For a reconstruction of the complete kinematics in this approach, one should start with the hard scattering, for which $\hat{s}$ has been chosen according to the hard scattering matrix element. By backwards evolution, the virtualities $Q_1^2 = -m_1^2$ and $Q_2^2 = -m_2^2$ of the two interacting partons are reconstructed. Initially the two partons are considered in their common c.m. frame, coming in along the $\pm z$ directions. Then the four-momentum vectors have the non-vanishing components

$$\begin{aligned} E_{1,2} &= \frac{\hat{s} \pm (Q_2^2 - Q_1^2)}{2\sqrt{\hat{s}}} \,, \\ p_{z1} = -p_{z2} &= \sqrt{\frac{(\hat{s} + Q_1^2 + Q_2^2)^2 - 4Q_1^2 Q_2^2}{4\hat{s}}} \,, \end{aligned} \tag{176}$$

with $(p_1 + p_2)^2 = \hat{s}$.

If, say, $Q_1^2 > Q_2^2$, then the branching $3 \to 1 + 4$, which produced parton 1, is the one that took place closest to the hard scattering, and the one to be reconstructed first. With the four-momentum p_3 known, $p_4 = p_3 - p_1$ is automatically known, so there are four degrees of freedom. One corresponds to a trivial azimuthal angle around the z axis. The z splitting variable for the $3 \to 1 + 4$ vertex is found as the same time as Q_1^2, and provides the constraint $(p_3 + p_2)^2 = \hat{s}/z$. The virtuality Q_3^2 is given by backwards evolution of parton 3.

One degree of freedom remains to be specified, and this is related to the possibility that parton 4 initiates a time-like parton shower, i.e. may have a non-zero mass. The maximum allowed squared mass $m_{\mathrm{max},4}^2$ is found for a collinear branching $3 \to 1 + 4$. In terms of the combinations

$$\begin{aligned} s_1 &= \hat{s} + Q_2^2 + Q_1^2 \,, \\ s_3 &= \frac{\hat{s}}{z} + Q_2^2 + Q_3^2 \,, \\ r_1 &= \sqrt{s_1^2 - 4Q_2^2 Q_1^2} \,, \\ r_3 &= \sqrt{s_3^2 - 4Q_2^2 Q_3^2} \,, \end{aligned} \tag{177}$$

one obtains

$$m_{\mathrm{max},4}^2 = \frac{s_1 s_3 - r_1 r_3}{2Q_2^2} - Q_1^2 - Q_3^2 \,, \tag{178}$$

which, for the special case of $Q_2^2 = 0$, reduces to

$$m_{\mathrm{max},4}^2 = \left\{ \frac{Q_1^2}{z} - Q_3^2 \right\} \left\{ \frac{\hat{s}}{\hat{s}+Q_1^2} - \frac{\hat{s}}{\hat{s}/z+Q_3^2} \right\} . \tag{179}$$

These constraints on m_4 are only the kinematical ones, in addition coherence phenomena could constrain the $m_{\mathrm{max},4}$ values further. Some options of this kind are available; the default one is to require additionally that $m_4^2 \leq Q_1^2$, i.e. lesser than the space-like virtuality of the sister parton.

With the maximum virtuality given, the final-state showering machinery may be used to give the development of the subsequent cascade, including the actual mass m_4^2, with $0 \leq m_4^2 \leq m_{\mathrm{max},4}^2$. The evolution is performed in the c.m. frame of the two 'resolved' partons, i.e. that of partons 1 and 2 for the branching $3 \to 1+4$, and parton 4 is assumed to have a nominal energy $E_{\mathrm{nom},4} = (1/z-1)\sqrt{\hat{s}}/2$. (Slight modifications appear if parton 4 has a non-vanishing mass m_{q} or m_ℓ.)

Using the relation $m_4^2 = (p_3 - p_1)^2$, the momentum of parton 3 may now be found as

$$\begin{aligned} E_3 &= \frac{1}{2\sqrt{\hat{s}}} \left\{ \frac{\hat{s}}{z} + Q_2^2 - Q_1^2 - m_4^2 \right\} , \\ p_{z3} &= \frac{1}{2p_{z1}} \left\{ s_3 - 2E_2E_3 \right\} , \\ p_{\perp,3}^2 &= \left\{ m_{\mathrm{max},4}^2 - m_4^2 \right\} \frac{(s_1s_3 + r_1r_3)/2 - Q_2^2(Q_1^2+Q_3^2+m_4^2)}{r_1^2} . \end{aligned} \tag{180}$$

The requirement that $m_4^2 \geq 0$ (or $\geq m_f^2$ for heavy flavours) imposes a constraint on allowed z values. This constraint cannot be included in the choice of Q_1^2, where it logically belongs, since it also depends on Q_2^2 and Q_3^2, which are unknown at this point. It is fairly rare (in the order of 10% of all events) that an unallowed z value is generated, and when it happens it is almost always for one of the two branchings closest to the hard interaction: for $Q_2^2 = 0$ eq. (179) may be solved to yield $z \leq \hat{s}/(\hat{s}+Q_1^2-Q_3^2)$, which is a more severe cut for $\hat{s}$ small and Q_1^2 large. Therefore an essentially bias-free way of coping is to redo completely any initial-state cascade for which this problem appears.

This completes the reconstruction of the $3 \to 1+4$ vertex. The subsystem made out of partons 3 and 2 may now be boosted to its rest frame and rotated to bring partons 3 and 2 along the $\pm z$ directions. The partons 1 and 4 now have opposite and compensating transverse momenta with respect to the event axis. When the next vertex is considered, either the one that produces parton 3 or the one that produces parton 2, the 3–2 subsystem will fill the function the 1–2 system did above, e.g. the rôle of $\hat{s} = \hat{s}_{12}$ in the formulae above is now played by $\hat{s}_{32} = \hat{s}_{12}/z$. The internal structure of the 3–2 system, i.e. the branching $3 \to 1+4$, appears nowhere in the continued description, but has become 'unresolved'. It is only reflected in the successive rotations and boosts performed to bring back the new endpoints to their common rest frame. Thereby the hard scattering subsystem 1–2 builds up a net transverse momentum and also an overall rotation of the hard scattering subsystem.

After a number of steps, the two outermost partons have virtualities $Q^2 < Q_0^2$ and then the shower is terminated and the endpoints assigned $Q^2 = 0$. Up to small corrections from primordial $k_\perp$, discussed in section 11.1, a final boost will bring the partons from their c.m. frame to the overall c.m. frame, where the x values of the outermost partons agree also with the light-cone definition.

10.3.4 Other initial-state shower aspects

In the formulae above, Q^2 has been used as argument for α_{s}, and not only as the space-like virtuality of partons. This is one possibility, but in fact loop calculations tend to indicate that

the proper argument for α_s is not Q^2 but $p_\perp^2 = (1-z)Q^2$ [Bas83]. The variable $p_\perp$ does have the interpretation of transverse momentum, although it is only exactly so for a branching $a \to bc$ with a and c massless and $Q^2 = -m_b^2$, and with z interpreted as light-cone fraction of energy and momentum. The use of $\alpha_s((1-z)Q^2)$ is default in the program. Indeed, if one wanted to, the complete shower might be interpreted as an evolution in $p_\perp^2$ rather than in Q^2.

Angular ordering is included in the shower evolution by default. However, as already mentioned, the physics is much more complicated than for timelike showers, and so this option should only be viewed as a first approximation. In the code the quantity ordered is an approximation of $p_\perp/p \approx \sin\theta$. (An alternative would have been $p_\perp/p_L \approx \tan\theta$, but this suffers from instability problems.)

In flavour excitation processes, a c (or b) quark enters the hard scattering and should be reconstructed by the shower as coming from a $\mathrm{g} \to \mathrm{c}\overline{\mathrm{c}}$ (or $\mathrm{g} \to \mathrm{b}\overline{\mathrm{b}}$) branching. Here an x value for the incoming c above $Q_\mathrm{c}^2/(Q_\mathrm{c}^2 + m_\mathrm{c}^2)$, where Q_c^2 is the spacelike virtuality of the c, does not allow a kinematical reconstruction of the gluon branching with an $x_\mathrm{g} < 1$, and is thus outside the allowed phase space. Such events (with some safety margin) are rejected. Currently they will appear in `PYSTAT(1)` listings in the 'Fraction of events that fail fragmentation cuts', which is partly misleading, but has the correct consequence of suppressing the physical cross section. Further, the Q^2 value of the backwards evolution of a c quark is by force kept above m_c^2, so as to ensure that the branching $\mathrm{g} \to \mathrm{c}\overline{\mathrm{c}}$ is not 'forgotten' by evolving Q^2 below Q_0^2. Thereby the possibility of having a c in the beam remnant proper is eliminated [Nor98]. Warning: as a consequence, flavour excitation is not at all possible too close to threshold. If the `KFIN` array in `PYSUBS` is set so as to require a c (or b) on either side, and the phase space is closed for such a c to come from a $\mathrm{g} \to \mathrm{c}\overline{\mathrm{c}}$ branching, the program will enter an infinite loop.

For proton beams, say, any c or b quark entering the hard scattering has to come from a preceding gluon splitting. This is not the case for a photon beam, since a photon has a c and b valence quark content. Therefore the above procedure need not be pursued there, but c and b quarks may indeed appear as beam remnants.

As we see, the initial-state showering algorithm leads to a net boost and rotation of the hard scattering subsystems. The overall final state is made even more complex by the additional final-state radiation. In principle, the complexity is very physical, but it may still have undesirable side effects. One such, discussed further in section 9.2, is that it is very difficult to generate events that fulfil specific kinematics conditions, since kinematics is smeared and even, at times, ambiguous.

A special case is encountered in Deeply Inelastic Scattering in ep collisions. Here the DIS x and Q^2 values are defined in terms of the scattered electron direction and energy, and therefore are unambiguous (except for issues of final-state photon radiation close to the electron direction). Neither initial- nor final-state showers preserve the kinematics of the scattered electron, however, and hence the DIS x and Q^2 are changed. In principle, this is perfectly legitimate, with the caveat that one then also should use different sets of parton distributions than ones derived from DIS, since these are based on the kinematics of the scattered lepton and nothing else. Alternatively, one might consider showering schemes that leave x and Q^2 unchanged. In [Ben88] detailed modifications are presented that make a preservation possible when radiation off the incoming and outgoing electron is neglected, but these are not included in the current version of PYTHIA. Instead the current `'gamma/`*lepton*`'` machinery explicitly separates off the $\mathrm{e} \to \mathrm{e}\gamma$ vertex from the continued fate of the photon.

The only reason for using the older machinery, such as process 10, is that this is still the only place where weak charged and neutral current effects can be considered. What is available there, as an option, is a simple machinery which preserves x and Q^2 from the effects of QCD radiation, and also from those of primordial $k_\perp$ and the beam remnant treatment, as follows. After the showers have been generated, the four-momentum of the scattered lepton is changed to the expected one, based on the nominal x and Q^2 values. The azimuthal angle of the lepton is maintained when the transverse momentum is adjusted. Photon radiation off the lepton leg is not fully accounted for, i.e. it is assumed that the energy of final-state photons is added to

that of the scattered electron for the definition of x and Q^2 (this is the normal procedure for parton-distribution definitions).

The change of three-momentum on the lepton side of the event is balanced by the final state partons on the hadron side, excluding the beam remnant but including all the partons both from initial- and final-state showering. The fraction of three-momentum shift taken by each parton is proportional to its original light-cone momentum in the direction of the incoming lepton, i.e. to $E \mp p_z$ for a hadron moving in the $\pm$ direction. This procedure guarantees momentum but not energy conservation. For the latter, one additional degree of freedom is needed, which is taken to be the longitudinal momentum of the initial state shower initiator. As this momentum is modified, the change is shared by the final state partons on the hadron side, according to the same light-cone fractions as before (based on the original momenta). Energy conservation requires that the total change in final state parton energies plus the change in lepton side energy equals the change in initiator energy. This condition can be turned into an iterative procedure to find the initiator momentum shift.

Sometimes the procedure may break down. For instance, an initiator with $x > 1$ may be reconstructed. If this should happen, the x and Q^2 values of the event are preserved, but new initial and final state showers are generated. After five such failures, the event is completely discarded in favour of a new kinematical setup.

Kindly note that the four-momentum of intermediate partons in the shower history are not being adjusted. In a listing of the complete event history, energy and momentum need then not be conserved in shower branchings. This mismatch could be fixed up, if need be.

The scheme presented above should not be taken too literally, but is rather intended as a contrast to the more sophisticated schemes already on the market, if one would like to understand whether the kind of conservation scheme chosen does affect the observable physics.

10.3.5 Matrix-element matching

In PYTHIA 6.1, matrix-element matching was introduced for the initial-state shower description of initial-state radiation in the production of a single colour-singlet resonance, such as $\gamma^*/\mathrm{Z}^0/\mathrm{W}^\pm$ [Miu99]. The basic idea is to map the kinematics between the PS and ME descriptions, and to find a correction factor that can be applied to hard emissions in the shower so as to bring agreement with the matrix-element expression. The PYTHIA shower kinematics definitions are based on Q^2 as the spacelike virtuality of the parton produced in a branching and z as the factor by which the $\hat{s}$ of the scattering subsystem is reduced by the branching. Some simple algebra then shows that the two $\mathrm{q}\overline{\mathrm{q}}' \to \mathrm{gW}^\pm$ emission rates disagree by a factor

$$R_{\mathrm{q}\overline{\mathrm{q}}'\to\mathrm{gW}}(\hat{s},\hat{t}) = \frac{(\mathrm{d}\hat{\sigma}/\mathrm{d}\hat{t})_{\mathrm{ME}}}{(\mathrm{d}\hat{\sigma}/\mathrm{d}\hat{t})_{\mathrm{PS}}} = \frac{\hat{t}^2+\hat{u}^2+2m_{\mathrm{W}}^2\hat{s}}{\hat{s}^2+m_{\mathrm{W}}^4}\ , \tag{181}$$

which is always between $1/2$ and 1. The shower can therefore be improved in two ways, relative to the old description. Firstly, the maximum virtuality of emissions is raised from $Q^2_{\mathrm{max}} \approx m^2_{\mathrm{W}}$ to $Q^2_{\mathrm{max}} = s$, i.e. the shower is allowed to populate the full phase space. Secondly, the emission rate for the final (which normally also is the hardest) $\mathrm{q} \to \mathrm{qg}$ emission on each side is corrected by the factor $R(\hat{s},\hat{t})$ above, so as to bring agreement with the matrix-element rate in the hard-emission region. In the backwards evolution shower algorithm [Sjö85], this is the first branching considered.

The other possible $O(\alpha_{\mathrm{s}})$ graph is $\mathrm{qg} \to \mathrm{q}'\mathrm{W}^\pm$, where the corresponding correction factor is

$$R_{\mathrm{qg}\to\mathrm{q}'\mathrm{W}}(\hat{s},\hat{t}) = \frac{(\mathrm{d}\hat{\sigma}/\mathrm{d}\hat{t})_{\mathrm{ME}}}{(\mathrm{d}\hat{\sigma}/\mathrm{d}\hat{t})_{\mathrm{PS}}} = \frac{\hat{s}^2+\hat{u}^2+2m_{\mathrm{W}}^2\hat{t}}{(\hat{s}-m_{\mathrm{W}}^2)^2+m_{\mathrm{W}}^4}\ , \tag{182}$$

which lies between 1 and 3. A probable reason for the lower shower rate here is that the shower does not explicitly simulate the s-channel graph $\mathrm{qg} \to \mathrm{q}^* \to \mathrm{q}'\mathrm{W}$. The $\mathrm{g} \to \mathrm{q}\overline{\mathrm{q}}$ branching

therefore has to be preweighted by a factor of 3 in the shower, but otherwise the method works the same as above. Obviously, the shower will mix the two alternative branchings, and the correction factor for a final branching is based on the current type.

The reweighting procedure prompts some other changes in the shower. In particular, $\hat{u} < 0$ translates into a constraint on the phase space of allowed branchings, not previously implemented. Here $\hat{u} = Q^2 - \hat{s}_{\rm old}(1-z)/z = Q^2 - \hat{s}_{\rm new}(1-z)$, where the association with the $\hat{u}$ variable is relevant if the branching is reinterpreted in terms of a $2 \to 2$ scattering. Usually such a requirement comes out of the kinematics, and therefore is imposed eventually anyway. The corner of emissions that do not respect this requirement is that where the Q^2 value of the spacelike emitting parton is little changed and the z value of the branching is close to unity. (That is, such branchings are kinematically allowed, but since the mapping to matrix-element variables would assume the first parton to have $Q^2 = 0$, this mapping gives an unphysical $\hat{u}$, and hence no possibility to impose a matrix-element correction factor.) The correct behaviour in this region is beyond leading-log predictivity. It is mainly important for the hardest emission, i.e. with largest Q^2. The effect of this change is to reduce the total amount of emission by a non-negligible amount when no matrix-element correction is applied. (This can be confirmed by using the special option `MSTP(68)=-1`.) For matrix-element corrections to be applied, this requirement must be used for the hardest branching, and then whether it is used or not for the softer ones is less relevant.

Our published comparisons with data on the $p_{\perp \rm W}$ spectrum show quite a good agreement with this improved simulation [Miu99]. A worry was that an unexpectedly large primordial $k_\perp$, around 4 GeV, was required to match the data in the low-$p_{\perp \rm Z}$ region. However, at that time we had not realized that the data were not fully unsmeared. The required primordial $k_\perp$ therefore drops by about a factor of two [Bál01]. This number is still uncomfortably large, but not too dissimilar from what is required in various resummation descriptions.

The method can also be used for initial-state photon emission, e.g. in the process $\rm e^+e^- \to \gamma^*/Z^0$. There the old default $Q^2_{\rm max} = m^2_{\rm Z}$ allowed no emission at large $p_\perp$, $p_\perp \gtrsim m_{\rm Z}$ at LEP2. This is now corrected by the increased $Q^2_{\rm max} = s$, and using the R of eq. (181) with $m_{\rm W} \to m_{\rm Z}$.

The above method does not address the issue of next-to-leading order corrections to the total W cross section. Rather, the implicit assumption is that such corrections, coming mainly from soft- and virtual-gluon effects, largely factorize from the hard-emission effects. That is, that the $p_\perp$ shape obtained in our approach will be rather unaffected by next-to-leading order corrections (when used both for the total and the high-$p_\perp$ cross section). A rescaling by a common K factor could then be applied by hand at the end of the day. However, the issue is not clear. Alternative approaches have been proposed, where more sophisticated matching procedures are used also to get the next-to-leading order corrections to the cross section integrated into the shower formalism [Mre99].

A matching can also be attempted for other processes than the ones above. Currently a matrix-element correction factor is also used for $\rm g \to gg$ and $\rm q \to gq$ branchings in the $\rm gg \to h^0$ process, in order to match on to the $\rm gg \to gh^0$ and $\rm qg \to qh^0$ matrix elements [Ell88]. The loop integrals of Higgs production are quite complex, however, and therefore only the expressions obtained in the limit of a heavy top quark is used as a starting point to define the ratios of $\rm gg \to gh^0$ and $\rm qg \to qh^0$ to $\rm gg \to h^0$ cross sections. (Whereas the $\rm gg \to h^0$ cross section by itself contains the complete expressions.) In this limit, the compact correction factors

$$R_{\rm gg\to gh^0}(\hat{s},\hat{t}) = \frac{(\mathrm{d}\hat{\sigma}/\mathrm{d}\hat{t})_{\rm ME}}{(\mathrm{d}\hat{\sigma}/\mathrm{d}\hat{t})_{\rm PS}} = \frac{\hat{s}^4 + \hat{t}^4 + \hat{u}^4 + m_{\rm h}^8}{2(\hat{s}^2 - m_{\rm h}^2(\hat{s} - m_{\rm h}^2))^2} \tag{183}$$

and

$$R_{\rm qg\to qh^0}(\hat{s},\hat{t}) = \frac{(\mathrm{d}\hat{\sigma}/\mathrm{d}\hat{t})_{\rm ME}}{(\mathrm{d}\hat{\sigma}/\mathrm{d}\hat{t})_{\rm PS}} = \frac{\hat{s}^2 + \hat{u}^2}{\hat{s}^2 + (\hat{s} - m_{\rm h}^2)^2} \tag{184}$$

can be derived. Even though they are clearly not as reliable as the above expressions for $\rm \gamma^*/Z^0/W^\pm$, they should hopefully represent an improved description relative to having no cor-

rection factor at all. For this reason they are applied not only for the standard model Higgs, but for all the three Higgs states h^0, H^0 and A^0. The Higgs correction factors are always in the comfortable range between $1/2$ and 1.

Note that a third process, $q\overline{q} \to gh^0$ does not fit into the pattern of the other two. The above process cannot be viewed as a showering correction to a lowest-order $q\overline{q} \to h^0$ one: since the q is assumed (essentially) massless there is no pointlike coupling. The graph above instead again involved a top loop, coupled to the initial state by a single s-channel gluon. The final-state gluon is necessary to balance colours in the process, and therefore the cross section is vanishing in the $p_\perp \to 0$ limit.

10.4 Routines and Common Block Variables

In this section we collect information on how to use the initial- and final-state showering routines. Of these PYSHOW for final-state radiation is the more generally interesting, since it can be called to let a user-defined parton configuration shower. PYSSPA, on the other hand, is so intertwined with the general structure of a PYTHIA event that it is of little use as a stand-alone product.

```
CALL PYSHOW(IP1,IP2,QMAX)
```

Purpose: to generate time-like parton showers, conventional or coherent. The performance of the program is regulated by the switches MSTJ(38) - MSTJ(50) and parameters PARJ(80) - PARJ(90). In order to keep track of the colour flow information, the positions K(I,4) and K(I,5) have to be organized properly for showering partons. Inside the PYTHIA programs, this is done automatically, but for external use proper care must be taken.

IP1 > 0, IP2 = 0 : generate a time-like parton shower for the parton in line IP1 in common block PYJETS, with maximum allowed mass QMAX. With only one parton at hand, one cannot simultaneously conserve both energy and momentum: we here choose to conserve energy and jet direction, while longitudinal momentum (along the jet axis) is not conserved.

IP1 > 0, IP2 > 0 : generate time-like parton showers for the two partons in lines IP1 and IP2 in the common block PYJETS, with maximum allowed mass for each parton QMAX. For shower evolution, the two partons are boosted to their c.m. frame. Energy and momentum is conserved for the pair of partons, although not for each individually. One of the two partons may be replaced by a nonradiating particle, such as a photon or a diquark; the energy and momentum of this particle will then be modified to conserve the total energy and momentum.

IP1 > 0, -7 $\leq$ IP2 < 0 : generate time-like parton showers for the -IP2 (at most 7) partons in lines IP1, IP1+1, ... IPI-IP2-1 in the common block PYJETS, with maximum allowed mass for each parton QMAX. The actions for IP2=-1 and IP2=-2 correspond to what is described above, but additionally larger numbers may be used to generate the evolution starting from three or more given partons. Then the partons are boosted to their c.m. frame, the direction of the momentum vector is conserved for each parton individually and energy for the system as a whole. It should be understood that the uncertainty in this option is larger than for two-parton systems, and that a number of the sophisticated features (such as coherence with the incoming colour flow) are not implemented.

IP1 > 0, IP2 = -8 : generate a four-parton system, where a history starting from two partons has already been constructed as discussed in subsection 10.2.7. Including intermediate partons this requires 8 lines, whence the IP2 value. This option is used in PY4JET, whereas you would normally not want to use this option directly yourself.

`QMAX :` the maximum allowed mass of a radiating parton, i.e. the starting value for the subsequent evolution. (In addition, the mass of a single parton may not exceed its energy, the mass of a parton in a system may not exceed the invariant mass of the system.)

```
FUNCTION PYMAEL(NI,X1,X2,R1,R2,ALPHA)
```

Purpose: returns the ratio of the first-order gluon emission rate normalized to the lowest-order event rate, eq. (166). An overall factor $C_F\alpha_s/2\pi$ is omitted, since the running of α_s probably is done better in shower language anyway.

`NI :` code of the matrix element to be used, see Table 25. In each group of four codes in that table, the first is for the 1 case, the second for the γ_5 one, the third for an arbitrary mixture, see `ALPHA` below, and the last for $1 \pm \gamma_5$.

`X1, X2 :` standard energy fractions of the two daughters.

`R1, R2 :` mass of the two daughters normalized to the mother mass.

`ALPHA:` fraction of the no-γ_5 (i.e. vector/scalar/...) part of the cross section; a free parameter for the third matrix element option of each group in Table 25 (13, 18, 23, 28, ...).

```
SUBROUTINE PYADSH(NFIN)
```

Purpose: to administrate a sequence of final-state showers for external processes, where the order normally is that all resonances have decayed before showers are considered, and therefore already existing daughters have to be boosted when their mothers radiate or take the recoil from radiation.

`NFIN :` line in the event record of the last final-state entry to consider.

```
SUBROUTINE PYSSPA(IPU1,IPU2)
```

Purpose: to generate the space-like showers of the initial-state radiation.

`IPU1, IPU2 :` positions of the two partons entering the hard scattering, from which the backwards evolution is initiated.

```
SUBROUTINE PYMEMX(MECOR,WTFF,WTGF,WTFG,WTGG)
```

Purpose: to set the maximum of the ratio of the correct matrix element to the one implied by the spacelike parton shower.

`MECOR :` kind of hard scattering process, 1 for $f + \bar{f} \to \gamma^*/Z^0/W^\pm/\ldots$ vector gauge bosons, 2 for $g + g \to h^0/H^0/A^0$.

`WTFF, WTGF, WTFG, WTGG :` maximum weights for $f \to f\,(+g/\gamma)$, $g/\gamma \to f\,(+\bar{f})$, $f \to g/\gamma\,(+f)$ and $g \to g\,(+g)$, respectively.

```
SUBROUTINE PYMEWT(MECOR,IFLCB,Q2,Z,PHIBR,WTME)
```

Purpose: to calculate the ratio of the correct matrix element to the one implied by the spacelike parton shower.

`MECOR :` kind of hard scattering process, 1 for $f + \bar{f} \to \gamma^*/Z^0/W^\pm/\ldots$ vector gauge bosons, 2 for $g + g \to h^0/H^0/A^0$.

`IFLCB :` kind of branching, 1 for $f \to f\,(+g/\gamma)$, 2 for $g/\gamma \to f\,(+\bar{f})$, 3 for $f \to g/\gamma\,(+f)$ and 4 for $g \to g\,(+g)$.

`Q2, Z` : Q^2 and z values of shower branching under consideration.
`PHIBR` : φ azimuthal angle of the shower branching; may be overwritten inside routine.
`WTME` : calculated matrix element correction weight, used in the acceptance/rejection of the shower branching under consideration.

```
COMMON/PYDAT1/MSTU(200),PARU(200),MSTJ(200),PARJ(200)
```

Purpose: to give access to a number of status codes and parameters which regulate the performance of PYTHIA. Most parameters are described in section 14.3; here only those related to `PYSHOW` are described.

`MSTJ(38)` : (D=0) matrix element code `NI` for `PYMAEL`; as in `MSTJ(47)`. If nonzero, the `MSTJ(38)` value overrides `MSTJ(47)`, but is then set `=0` in the `PYSHOW` call. The usefulness of this switch lies in processes where sequential decays occur and thus there are several showers, each requiring its matrix element. Therefore `MSTJ(38)` can be set in the calling routine when it is known, and when not set one defaults back to the attempted matching procedure of `MSTJ(47)=3` (e.g.).

`MSTJ(40)` : (D=0) possibility to suppress the branching probability for a branching $q \to qg$ (or $q \to q\gamma$) of a quark produced in the decay of an unstable particle with width Γ, where this width has to be specified by you in `PARJ(89)`. The algorithm used is not exact, but still gives some impression of potential effects. This switch ought to have appeared at the end of the current list of shower switches (after `MSTJ(50)`), but because of lack of space it appears immediately before.

- `= 0` : no suppression, i.e. the standard parton-shower machinery.
- `= 1` : suppress radiation by a factor $\chi(\omega) = \Gamma^2/(\Gamma^2+\omega^2)$, where ω is the energy of the gluon (or photon) in the rest frame of the radiating dipole. Essentially this means that hard radiation with $\omega > \Gamma$ is removed.
- `= 2` : suppress radiation by a factor $1-\chi(\omega) = \omega^2/(\Gamma^2+\omega^2)$, where ω is the energy of the gluon (or photon) in the rest frame of the radiating dipole. Essentially this means that soft radiation with $\omega < \Gamma$ is removed.

`MSTJ(41)` : (D=2) type of branchings allowed in shower.

- `= 0` : no branchings at all, i.e. shower is switched off.
- `= 1` : QCD type branchings of quarks and gluons.
- `= 2` : also emission of photons off quarks and leptons; the photons are assumed on the mass shell.
- `= 10` : as `=2`, but enhance photon emission by a factor `PARJ(84)`. This option is unphysical, but for moderate values, `PARJ(84)`≤ 10, it may be used to enhance the prompt photon signal in $q\overline{q}$ events. The normalization of the prompt photon rate should then be scaled down by the same factor. The dangers of an improper use are significant, so do not use this option if you do not know what you are doing.

`MSTJ(42)` : (D=2) branching mode, especially coherence level, for time-like showers.

- `= 1` : conventional branching, i.e. without angular ordering.
- `= 2` : coherent branching, i.e. with angular ordering.
- `= 3` : in a branching $a \to bg$, where m_b is nonvanishing, the decay angle is reduced by a factor $(1+(m_b^2/m_a^2)(1-z)/z)^{-1}$, thereby taking into account mass effects in the decay [Nor01]. Therefore more branchings are acceptable from an angular ordering point of view. In the definition of the angle in a $g \to q\overline{q}$ branchings, the naive massless expression is reduced by a factor $\sqrt{1-4m_q^2/m_g^2}$, which can be motivated by a corresponding actual reduction in the $p_\perp$ by mass effects. The requirement of angular ordering then kills fewer potential $g \to q\overline{q}$ branchings, i.e. the rate of such comes up. The $g \to gg$ branchings are

not changed from =2. This option is fully within the range of uncertainty that exists.

= 4 : as =3 for $a \to b$g and g $\to$ gg branchings, but no angular ordering requirement conditions at all are imposed on g $\to$ q$\overline{\text{q}}$ branchings. This is an unrealistic extreme, and results obtained with it should not be overstressed. However, for some studies it is of interest. For instance, it not only gives a much higher rate of charm and bottom production in showers, but also affects the kinematical distributions of such pairs.

= 5 : new 'intermediate' coherence level [Nor01], where the consecutive gluon emissions off the original pair of branching partons is not constrained by angular ordering at all. The subsequent showering of such a gluon is angular ordered, however, starting from its production angle. At LEP energies, this gives almost no change in the total parton multiplicity, but this multiplicity now increases somewhat faster with energy than before, in better agreement with analytical formulae. (The PYSHOW algorithm overconstrains the shower by ordering emissions in mass and then vetoing increasing angles. This is a first simple attempt to redress the issue.) Other branchings as in =2.

= 6 : 'intermediate' coherence level as =5 for primary partons, unchanged for g $\to$ gg and reduced angle for g $\to$ q$\overline{\text{q}}$ and secondary q $\to$ qg as in =3.

= 7 : 'intermediate' coherence level as =5 for primary partons, unchanged for g $\to$ gg, reduced angle for secondary q $\to$ qg as in =3 and no angular ordering for g $\to$ q$\overline{\text{q}}$ as in =4.

MSTJ(43) : (D=4) choice of z definition in branching.

= 1 : energy fraction in grandmother's rest frame ('local, constrained').

= 2 : energy fraction in grandmother's rest frame assuming massless daughters, with energy and momentum reshuffled for massive ones ('local, unconstrained').

= 3 : energy fraction in c.m. frame of the showering partons ('global, constrained').

= 4 : energy fraction in c.m. frame of the showering partons assuming massless daughters, with energy and momentum reshuffled for massive ones ('global, unconstrained').

MSTJ(44) : (D=2) choice of α_s scale for shower.

= 0 : fixed at PARU(111) value.

= 1 : running with $Q^2 = m^2/4$, m mass of decaying parton, Λ as stored in PARJ(81) (natural choice for conventional showers).

= 2 : running with $Q^2 = z(1-z)m^2$, i.e. roughly $p_\perp^2$ of branching, Λ as stored in PARJ(81) (natural choice for coherent showers).

= 3 : while $p_\perp^2$ is used as α_s argument in q $\to$ qg and g $\to$ gg branchings, as in =2, instead $m^2/4$ is used as argument for g $\to$ q$\overline{\text{q}}$ ones. The argument is that the soft-gluon resummation results suggesting the $p_\perp^2$ scale [Ama80] in the former processes is not valid for the latter one, so that any multiple of the mass of the branching parton is a perfectly valid alternative. The $m^2/4$ ones then gives continuity with $p_\perp^2$ for $z = 1/2$. Furthermore, with this choice, it is no longer necessary to have the requirement of a minimum $p_\perp$ in branchings, else required in order to avoid having α_s blow up. Therefore, in this option, that cut has been removed for g $\to$ gg branchings. Specifically, when combined with MSTJ(42)=4, it is possible to reproduce the simple $1+\cos^2\theta$ angular distribution of g $\to$ gg branchings, which is not possible in any other approach. (However it may give too high a charm and bottom production rate in showers [Nor01].)

= 4 : $p_\perp^2$ as in =2, but scaled down by a factor $(1-m_b^2/m_a^2)^2$ for a branching $a \to b$g with b massive, in an attempt better to take into account the mass effect on kinematics.

`= 5 :` as for `=4` for $q \to qg$, unchanged for $g \to gg$ and as `=3` for $g \to q\bar{q}$.

`MSTJ(45)` : (D=5) maximum flavour that can be produced in shower by $g \to q\bar{q}$; also used to determine the maximum number of active flavours in the α_s factor in parton showers (here with a minimum of 3).

`MSTJ(46)` : (D=3) nonhomogeneous azimuthal distributions in a shower branching.

`= 0 :` azimuthal angle is chosen uniformly.

`= 1 :` nonhomogeneous azimuthal angle in gluon decays due to a kinematics-dependent effective gluon polarization. Not meaningful for scalar model, i.e. then same as `=0`.

`= 2 :` nonhomogeneous azimuthal angle in gluon decay due to interference with nearest neighbour (in colour). Not meaningful for Abelian model, i.e. then same as `=0`.

`= 3 :` nonhomogeneous azimuthal angle in gluon decay due to both polarization (`=1`) and interference (`=2`). Not meaningful for Abelian model, i.e. then same as `=1`. Not meaningful for scalar model, i.e. then same as `=2`.

`MSTJ(47)` : (D=3) matrix-element-motivated corrections to the gluon shower emission rate in generic processes of the type $a \to bc$g. Also, in the massless fermion approximation, with an imagined vector source, to the lowest-order $q\bar{q}\gamma$, $\ell^+\ell^-\gamma$ or $\ell\nu_\ell\gamma$ matrix elements, i.e. more primitive than for QCD radiation.

`= 0 :` no corrections.

`= 1 - 5 :` yes; try to match to the most relevant matrix element and default back to an assumed source (e.g. a vector for a $q\bar{q}$ pair) if the correct mother particle cannot be found.

`= 6 - :` yes, match to the specific matrix element code `NI = MSTJ(47)` of the `PYMAEL` function; see Table 25.

Warning : since a process may contain sequential decays involving several different kinds of matrix elements, it may be dangerous to fix `MSTJ(47)` to a specialized value > 5; see `MSTJ(38)` above.

`MSTJ(48)` : (D=0) possibility to impose maximum angle for the first branching in a shower.

`= 0 :` no explicit maximum angle.

`= 1 :` maximum angle given by `PARJ(85)` for single showering parton, by `PARJ(85)` and `PARJ(86)` for pair of showering partons.

`MSTJ(49)` : (D=0) possibility to change the branching probabilities according to some alternative toy models (note that the Q^2 evolution of α_s may well be different in these models, but that only the `MSTJ(44)` options are at your disposal).

`= 0 :` standard QCD branchings.

`= 1 :` branchings according to a scalar gluon theory, i.e. the splitting kernels in the evolution equations are, with a common factor $\alpha_s/(2\pi)$ omitted, $P_{q\to qg} = (2/3)(1-z)$, $P_{g\to gg} =$ `PARJ(87)`, $P_{g\to q\bar{q}} =$ `PARJ(88)` (for each separate flavour). The couplings of the gluon have been left as free parameters, since they depend on the colour structure assumed. Note that, since a spin 0 object decays isotropically, the gluon splitting kernels contain no z dependence.

`= 2 :` branchings according to an Abelian vector gluon theory, i.e. the colour factors are changed (compared with QCD) according to $C_F = 4/3 \to 1$, $N_C = 3 \to 0$, $T_R = 1/2 \to 3$. Note that an Abelian model is not expected to contain any coherence effects between gluons, so that one should normally use `MSTJ(42)=1` and `MSTJ(46)=` 0 or 1. Also, α_s is expected to increase with increasing Q^2 scale, rather than decrease. No such α_s option is available; the one that comes closest is `MSTJ(44)=0`, i.e. a fix value.

`MSTJ(50)` : (D=3) possibility to introduce colour coherence effects in the first branching of a final state shower; mainly of relevance for QCD parton–parton scattering processes.

= 0 : none.
= 1 : impose an azimuthal anisotropy.
= 2 : restrict the polar angle of a branching to be smaller than the scattering angle of the relevant colour flow.
= 3 : both azimuthal anisotropy and restricted polar angles.
Note: for subsequent branchings the (polar) angular ordering is automatic (MSTP(42)=2) and MSTJ(46)=3).

PARJ(80) : (D=0.5) 'parity' mixing parameter, α value for the PYMAEL routine, to be used when MSTJ(38) is nonvanishing.

PARJ(81) : (D=0.29 GeV) Λ value in running α_s for parton showers (see MSTJ(44)). This is used in all user calls to PYSHOW, in the PYEEVT/PYONIA e^+e^- routines, and in a resonance decay. It is not intended for other timelike showers, however, for which PARP(72) is used.

PARJ(82) : (D=1.0 GeV) invariant mass cut-off $m_{\min}$ of parton showers, below which partons are not assumed to radiate. For $Q^2 = p_\perp^2$ (MSTJ(44)=2) PARJ(82)/2 additionally gives the minimum $p_\perp$ of a branching. To avoid infinite α_s values, one must have PARJ(82)$>$ 2$\times$PARJ(81) for MSTJ(44)$\geq$ 1 (this is automatically checked in the program, with 2.2$\times$PARJ(81) as the lowest value attainable).

PARJ(83) : (D=1.0 GeV) invariant mass cut-off $m_{\min}$ used for photon emission in parton showers, below which quarks are not assumed to radiate. The function of PARJ(83) closely parallels that of PARJ(82) for QCD branchings, but there is a priori no requirement that the two be equal. The cut-off for photon emission off leptons is given by PARJ(90).

PARJ(84) : (D=1.) used for option MSTJ(41)=10 as a multiplicative factor in the prompt photon emission rate in final state parton showers. Unphysical but useful technical trick, so beware!

PARJ(85), PARJ(86) : (D=10.,10.) maximum opening angles allowed in the first branching of parton showers; see MSTJ(48).

PARJ(87) : (D=0.) coupling of $g \to gg$ in scalar gluon shower, see MSTJ(49)=1.

PARJ(88) : (D=0.) coupling of $g \to q\bar{q}$ in scalar gluon shower (per quark species), see MSTJ(49)=1.

PARJ(89) : (D=0. GeV) the width of the unstable particle studied for the MSTJ(40)>0 options; to be set by you (separately for each PYSHOW call, if need be).

PARJ(90) : (D=0.0001 GeV) invariant mass cut-off $m_{\min}$ used for photon emission in parton showers, below which leptons are not assumed to radiate, cf. PARJ(83) for radiation off quarks. By making this separation of cut-off values, photon emission off leptons becomes more realistic, covering a larger part of the phase space. The emission rate is still not well reproduced for lepton-photon invariant masses smaller than roughly twice the lepton mass itself.

```
COMMON/PYPARS/MSTP(200),PARP(200),MSTI(200),PARI(200)
```

Purpose: to give access to status code and parameters which regulate the performance of PYTHIA. Most parameters are described in section 9.3; here only those related to PYSSPA and PYSHOW are described.

MSTP(22) : (D=0) special override of normal Q^2 definition used for maximum of parton-shower evolution. This option only affects processes 10 and 83 (Deeply Inelastic Scattering) and only in lepton–hadron events.
= 0 : use the scale as given in MSTP(32).
= 1 : use the DIS Q^2 scale, i.e. $-\hat{t}$.
= 2 : use the DIS W^2 scale, i.e. $(-\hat{t})(1-x)/x$.
= 3 : use the DIS $Q \times W$ scale, i.e. $(-\hat{t})\sqrt{(1-x)/x}$.

= 4 : use the scale $Q^2(1-x)\max(1,\ln(1/x))$, as motivated by first order matrix elements [Ing80, Alt78].

Note: in all of these alternatives, a multiplicative factor is introduced by PARP(67) and PARP(71), as usual.

MSTP(61) : (D=1) master switch for initial-state QCD and QED radiation.

= 0 : off.

= 1 : on.

MSTP(62) : (D=3) level of coherence imposed on the space-like parton-shower evolution.

= 1 : none, i.e. neither Q^2 values nor angles need be ordered.

= 2 : Q^2 values at branches are strictly ordered, increasing towards the hard interaction.

= 3 : Q^2 values and opening angles of emitted (on-mass-shell or time-like) partons are both strictly ordered, increasing towards the hard interaction.

MSTP(63) : (D=2) structure of associated time-like showers, i.e. showers initiated by emission off the incoming space-like partons.

= 0 : no associated showers are allowed, i.e. emitted partons are put on the mass shell.

= 1 : a shower may evolve, with maximum allowed time-like virtuality set by the phase space only.

= 2 : a shower may evolve, with maximum allowed time-like virtuality set by phase space or by PARP(71) times the Q^2 value of the space-like parton created in the same vertex, whichever is the stronger constraint.

= 2 : a shower may evolve, with maximum allowed time-like virtuality set by phase space, but further constrained to evolve within a cone with opening angle (approximately) set by the opening angle of the branching where the showering parton was produced.

MSTP(64) : (D=2) choice of α_s and Q^2 scale in space-like parton showers.

= 0 : α_s is taken to be fix at the value PARU(111).

= 1 : first-order running α_s with argument PARP(63)Q^2.

= 2 : first-order running α_s with argument PARP(64)$k_\perp^2$ =PARP(64)$(1-z)Q^2$.

MSTP(65) : (D=1) treatment of soft gluon emission in space-like parton-shower evolution.

= 0 : soft gluons are entirely neglected.

= 1 : soft gluon emission is resummed and included together with the hard radiation as an effective z shift.

MSTP(66) : (D=5) choice of lower cut-off for initial-state QCD radiation in VMD or anomalous photoproduction events, and matching to primordial $k_\perp$.

= 0 : the lower Q^2 cutoff is the standard one in PARP(62)^2.

= 1 : for anomalous photons, the lower Q^2 cut-off is the larger of PARP(62)^2 and VINT(283) or VINT(284), where the latter is the virtuality scale for the $\gamma \to q\bar{q}$ vertex on the appropriate side of the event. The VINT values are selected logarithmically even between PARP(15)^2 and the Q^2 scale of the parton distributions of the hard process.

= 2 : extended option of the above, intended for virtual photons. For VMD photons, the lower Q^2 cut-off is the larger of PARP(62)^2 and the $P_{\rm int}^2$ scale of the SaS parton distributions. For anomalous photons, the lower cut-off is chosen as for =1, but the VINT(283) and VINT(284) are here selected logarithmically even between $P_{\rm int}^2$ and the Q^2 scale of the parton distributions of the hard process.

= 3 : the $k_\perp$ of the anomalous/GVMD component is distributed like $1/k_\perp^2$ between k_0 and $p_{\perp\rm min}(W^2)$. Apart from the change of the upper limit, this option works just like =1.

= 4 : a stronger damping at large $k_\perp$, like $\mathrm{d}k_\perp^2/(k_\perp^2+Q^2/4)^2$ with $k_0 < k_\perp <$

$p_{\perp\mathrm{min}}(W^2)$. Apart from this, it works like =1.

= 5 : a $k_\perp$ generated as in =4 is added vectorially with a standard Gaussian $k_\perp$ generated like for VMD states. Ensures that GVMD has typical $k_\perp$'s above those of VMD, in spite of the large primordial $k_\perp$'s implied by hadronic physics. (Probably attributable to a lack of soft QCD radiation in parton showers.)

MSTP(67) : (D=2) possibility to introduce colour coherence effects in the first branching of the backwards evolution of an initial state shower; mainly of relevance for QCD parton–parton scattering processes.

= 0 : none.

= 2 : restrict the polar angle of a branching to be smaller than the scattering angle of the relevant colour flow.

Note 1: azimuthal anisotropies have not yet been included.

Note 2: for subsequent branchings, MSTP(62)=3 is used to restrict the (polar) angular range of branchings.

MSTP(68) : (D=1) choice of maximum virtuality scale and matrix-element matching scheme for initial-state radiation.

= 0 : maximum shower virtuality is the same as the Q^2 choice for the parton distributions, see MSTP(32). (Except that the multiplicative extra factor PARP(34) is absent and instead PARP(67) can be used for this purpose.) No matrix-element correction.

= 1 : as =0 for most processes, but new scheme for processes 1, 2, 141, 142, 144 and 102, i.e. single s-channel colourless gauge boson and Higgs production: γ^*/Z^0, $\mathrm{W}^\pm$, Z'^0, $\mathrm{W}'^\pm$, R and h^0. Here the maximum scale of shower evolution is s, the total squared energy. The nearest branching on either side of the hard scattering is corrected by the ratio of the first-order matrix-element weight to the parton-shower one, so as to obtain an improved description. For gauge boson production, this branching can be of the types $\mathrm{q} \to \mathrm{q} + \mathrm{g}$, $\mathrm{f} \to \mathrm{f} + \gamma$, $\mathrm{g} \to \mathrm{q} + \overline{\mathrm{q}}$ or $\gamma \to \mathrm{f} + \overline{\mathrm{f}}$, while for Higgs production it is $\mathrm{g} \to \mathrm{g} + \mathrm{g}$. See section 10.3.5 for a detailed description. Note that the improvements apply both for incoming hadron and lepton beams.

= 2 : the maximum scale for initial-state shower evolution is always selected to be s, except for the $2 \to 2$ QCD processes 11, 12, 13, 28, 53 and 68. The QCD exception is to avoid the double-counting issues that could easily arise here. Based on the experience in [Miu99], there is reason to assume that this does give an improved qualitative description of the high-$p_\perp$ tail, although the quantitative agreement is currently beyond our control. No matrix-element corrections, even for the processes in =1.

= -1 : as =0, except that there is no requirement on $\hat{u}$ being negative.

MSTP(69) : (D=0) possibility to change Q^2 scale for parton distributions from the MSTP(32) choice, especially for $\mathrm{e}^+\mathrm{e}^-$.

= 0 : use MSTP(32) scale.

= 1 : in lepton-lepton collisions, the QED lepton-inside-lepton parton distributions are evaluated with s, the full squared c.m. energy, as scale.

= 2 : s is used as parton distribution scale also in other processes.

MSTP(71) : (D=1) master switch for final-state QCD and QED radiation.

= 0 : off.

= 1 : on.

PARP(61) : (D=0.25 GeV) Λ value used in space-like parton shower (see MSTP(64)). This value may be overwritten, see MSTP(3).

PARP(62) : (D=1. GeV) effective cut-off Q or $k_\perp$ value (see MSTP(64)), below which space-like parton showers are not evolved.

PARP(63) : (D=0.25) in space-like shower evolution the virtuality Q^2 of a parton is multiplied

by PARP(63) for use as a scale in α_s and parton distributions when MSTP(64)=1.

PARP(64) : (D=1.) in space-like parton-shower evolution the squared transverse momentum evolution scale $k_\perp^2$ is multiplied by PARP(64) for use as a scale in α_s and parton distributions when MSTP(64)=2.

PARP(65) : (D=2. GeV) effective minimum energy (in c.m. frame) of time-like or on-shell parton emitted in space-like shower; see also PARP(66). For a hard subprocess moving in the rest frame of the hard process, this number is reduced roughly by a factor $1/\gamma$ for the boost to the hard scattering rest frame.

PARP(66) : (D=0.001) effective lower cut-off on $1-z$ in space-like showers, in addition to the cut implied by PARP(65).

PARP(67) : (D=1.) the Q^2 scale of the hard scattering (see MSTP(32)) is multiplied by PARP(67) to define the maximum parton virtuality allowed in space-like showers. This does not apply to *s*-channel resonances, where the maximum virtuality is set by m^2. The current default is based on arguments from a matching of scales in heavy-flavour production [Nor98], and other values such as 4 (the previous default) could be imagined from other arguments or in other processes.

PARP(68) : (D=0.001 GeV) lower Q cut-off for QED space-like showers. Comes in addition to a hardcoded cut that the Q^2 is at least $2m_e^2$, $2m_\mu^2$ or $2m_\tau^2$, as the case may be.

PARP(71) : (D=4.) the Q^2 scale of the hard scattering (see MSTP(32)) is multiplied by PARP(71) to define the maximum parton virtuality allowed in time-like showers. This does not apply to *s*-channel resonances, where the maximum virtuality is set by m^2. Like for PARP(67) this number is uncertain.

PARP(72) : (D=0.25 GeV) Λ value used in running α_s for timelike parton showers, except for showers in the decay of a resonance. (Resonance decay, e.g. γ^*/Z^0 decay, is instead set by PARJ(81).)

11 Beam Remnants and Underlying Events

Each incoming beam particle may leave behind a beam remnant, which does not take part in the initial-state radiation or hard scattering process. If nothing else, the remnants need be reconstructed and connected to the rest of the event. In hadron–hadron collisions, the composite nature of the two incoming beam particles implies the additional possibility that several parton pairs undergo separate hard or semi-hard scatterings, 'multiple interactions'. This may give a non-negligible contribution to the 'underlying event' structure, and thus to the total multiplicity. Finally, in high-luminosity colliders, it is possible to have several collisions between beam particles in one and the same beam crossing, i.e. pile-up events, which further act to build up the general particle production activity that is to be observed by detectors. These three aspects are described in turn, with emphasis on the middle one, that of multiple interactions within a single hadron–hadron collision.

The main reference on the multiple interactions model is [Sjö87a].

11.1 Beam Remnants

The initial-state radiation algorithm reconstructs one shower initiator in each beam. (If initial-state radiation is not included, the initiator is nothing but the incoming parton to the hard interaction.) Together the two initiators delineate an interaction subsystem, which contains all the partons that participate in the initial-state showers, in the hard interaction, and in the final-state showers. Left behind are two beam remnants which, to first approximation, just sail through, unaffected by the hard process. (The issue of additional interactions is covered in the next section.)

A description of the beam remnant structure contains a few components. First, given the flavour content of a (colour-singlet) beam particle, and the flavour and colour of the initiator parton, it is possible to reconstruct the flavour and colour of the beam remnant. Sometimes the remnant may be represented by just a single parton or diquark, but often the remnant has to be subdivided into two separate objects. In the latter case it is necessary to share the remnant energy and momentum between the two. Due to Fermi motion inside hadron beams, the initiator parton may have a 'primordial $k_\perp$' transverse momentum motion, which has to be compensated by the beam remnant. If the remnant is subdivided, there may also be a relative transverse momentum. In the end, total energy and momentum has to be conserved. To first approximation, this is ensured within each remnant separately, but some final global adjustments are necessary to compensate for the primordial $k_\perp$ and any effective beam remnant mass.

Consider first a proton (or, with trivial modifications, any other baryon or antibaryon).

- If the initiator parton is a u or d quark, it is assumed to be a valence quark, and therefore leaves behind a diquark beam remnant, i.e. either a ud or a uu diquark, in a colour antitriplet state. Relative probabilities for different diquark spins are derived within the context of the non-relativistic **SU(6)** model, i.e. flavour **SU(3)** times spin **SU(2)**. Thus a ud is $3/4$ ud_0 and $1/4$ ud_1, while a uu is always uu_1.
- An initiator gluon leaves behind a colour octet uud state, which is subdivided into a colour triplet quark and a colour antitriplet diquark. **SU(6)** gives the appropriate subdivision, $1/2$ of the time into $\mathrm{u}+\mathrm{ud}_0$, $1/6$ into $\mathrm{u}+\mathrm{ud}_1$ and $1/3$ into $\mathrm{d}+\mathrm{uu}_1$.
- A sea quark initiator, such as an s, leaves behind a $\mathrm{uud}\bar{\mathrm{s}}$ four-quark state. The PDG flavour coding scheme and the fragmentation routines do not foresee such a state, so therefore it is subdivided into a meson plus a diquark, i.e. $1/2$ into $\mathrm{u}\bar{\mathrm{s}}+\mathrm{ud}_0$, $1/6$ into $\mathrm{u}\bar{\mathrm{s}}+\mathrm{ud}_1$ and $1/3$ into $\mathrm{d}\bar{\mathrm{s}}+\mathrm{uu}_1$. Once the flavours of the meson are determined, the choice of meson multiplet is performed as in the standard fragmentation description.
- Finally, an antiquark initiator, such as an $\bar{\mathrm{s}}$, leaves behind a uuds four-quark state, which is subdivided into a baryon plus a quark. Since, to first approximation, the $\mathrm{s}\bar{\mathrm{s}}$ pair comes from the branching $\mathrm{g}\to\mathrm{s}\bar{\mathrm{s}}$ of a colour octet gluon, the subdivision $\mathrm{uud}+\mathrm{s}$ is not allowed, since it would correspond to a colour-singlet $\mathrm{s}\bar{\mathrm{s}}$. Therefore the subdivision is $1/2$ into

$\mathrm{ud_0s+u}$, $1/6$ into $\mathrm{ud_1s+u}$ and $1/3$ into $\mathrm{uu_1s+d}$. A baryon is formed among the ones possible for the given flavour content and diquark spin, according to the relative probabilities used in the fragmentation. One could argue for an additional weighting to count the number of baryon states available for a given diquark plus quark combination, but this has not been included.

One may note that any u or d quark taken out of the proton is automatically assumed to be a valence quark. Clearly this is unrealistic, but not quite as bad as it might seem. In particular, one should remember that the beam remnant scenario is applied to the initial-state shower initiators at a scale of $Q_0 \approx 1$ GeV and at an x value usually much larger than the x at the hard scattering. The sea quark contribution therefore normally is negligible.

For a meson beam remnant, the rules are in the same spirit, but somewhat easier, since no diquark or baryons need be taken into account. Thus a valence quark (antiquark) initiator leaves behind a valence antiquark (quark), a gluon initiator leaves behind a valence quark plus a valence antiquark, and a sea quark (antiquark) leaves behind a meson (which contains the partner to the sea parton) plus a valence antiquark (quark).

A resolved photon is similar in spirit to a meson. A VMD photon is associated with either ρ^0, ω, ϕ or J/ψ, and so corresponds to a well-defined valence flavour content. Since the ρ^0 and ω are supposed to add coherently, the $\mathrm{u\overline{u}:d\overline{d}}$ mixing is in the ratio $4:1$. Similarly a GVMD state is characterized by its $\mathrm{q\overline{q}}$ classification, in rates according to e_q^2 times a mass suppression for heavier quarks.

In the older photon physics options, where a quark content inside an electron is obtained by a numerical convolution, one does not have to make the distinction between valence and sea flavour. Thus any quark (antiquark) initiator leaves behind the matching antiquark (quark), and a gluon leaves behind a quark + antiquark pair. The relative quark flavour composition in the latter case is assumed proportional to e_q^2 among light flavours, i.e. $2/3$ into $\mathrm{u+\overline{u}}$, $1/6$ into $\mathrm{d+\overline{d}}$, and $1/6$ into $\mathrm{s+\overline{s}}$. If one wanted to, one could also have chosen to represent the remnant by a single gluon.

If no initial-state radiation is assumed, an electron (or, in general, a lepton or a neutrino) leaves behind no beam remnant. Also when radiation is included, one would expect to recover a single electron with the full beam energy when the shower initiator is reconstructed. This does not have to happen, e.g. if the initial-state shower is cut off at a non-vanishing scale, such that some of the emission at low Q^2 values is not simulated. Further, for purely technical reasons, the distribution of an electron inside an electron, $f_\mathrm{e}^\mathrm{e}(x,Q^2)$, is cut off at $x = 1-10^{-10}$. This means that always, when initial-state radiation is included, a fraction of at least 10^{-10} of the beam energy has to be put into one single photon along the beam direction, to represent this not simulated radiation. The physics is here slightly different from the standard beam remnant concept, but it is handled with the same machinery. Beam remnants can also appear when the electron is resolved with the use of parton distributions, but initial-state radiation is switched off. Conceptually, this is a contradiction, since it is the initial-state radiation that builds up the parton distributions, but sometimes the combination is still useful. Finally, since QED radiation has not yet been included in events with resolved photons inside electrons, also in this case effective beam remnants have to be assigned by the program.

The beam remnant assignments inside an electron, in either of the cases above, is as follows.

- An $\mathrm{e^-}$ initiator leaves behind a γ remnant.
- A γ initiator leaves behind an $\mathrm{e^-}$ remnant.
- An $\mathrm{e^+}$ initiator leaves behind an $\mathrm{e^-+e^-}$ remnant.
- A q ($\overline{\mathrm{q}}$) initiator leaves behind a $\mathrm{\overline{q}+e^-}$ ($\mathrm{q+e^-}$) remnant.
- A g initiator leaves behind a $\mathrm{g+e^-}$ remnant. One could argue that, in agreement with the treatment of photon beams above, the remnant should be $\mathrm{q+\overline{q}+e^-}$. The program currently does not allow for three beam remnant objects, however.

It is customary to assign a primordial transverse momentum to the shower initiator, to take into account the motion of quarks inside the original hadron, basically as required by the un-

certainty principle. A number of the order of $\langle k_\perp \rangle \approx m_\mathrm{p}/3 \approx 300$ MeV could therefore be expected. However, in hadronic collisions much higher numbers than that are often required to describe data, typically of the order of or even above 1 GeV [EMC87, Bál01] if a Gaussian parameterization is used. (This number is now the default.) Thus, an interpretation as a purely nonperturbative motion inside a hadron is difficult to maintain.

Instead a likely culprit is the initial-state shower algorithm. This is set up to cover the region of hard emissions, but may miss out on some of the softer activity, which inherently borders on nonperturbative physics. By default, the shower does not evolve down to scales below $Q_0 = 1$ GeV. Any shortfall in shower activity around or below this cutoff then has to be compensated by the primordial $k_\perp$ source, which thereby largely loses its original meaning. One specific reason for such a shortfall is that the current initial-state shower algorithm does not include non-order emissions in Q^2, as is predicted to occur especially at small x and Q^2 within the BFKL/CCFM framework [Lip76, Cia87].

By the hard scattering and initial-state radiation machinery, the shower initiator has been assigned some fraction x of the four-momentum of the beam particle, leaving behind $1-x$ to the remnant. If the remnant consists of two objects, this energy and momentum has to be shared, somehow. For an electron in the old photoproduction machinery, the sharing is given from first principles: if, e.g., the initiator is a q, then that q was produced in the sequence of branchings $\mathrm{e} \to \gamma \to \mathrm{q}$, where x_γ is distributed according to the convolution in eq. (54). Therefore the $\overline{\mathrm{q}}$ remnant takes a fraction $\chi = (x_\gamma - x)/(1-x)$ of the total remnant energy, and the e takes $1-\chi$.

For the other beam remnants, the relative energy-sharing variable χ is not known from first principles, but picked according to some suitable parameterization. Normally several different options are available, that can be set separately for baryon and meson beams, and for hadron + quark and quark + diquark (or antiquark) remnants. In one extreme are shapes in agreement with naïve counting rules, i.e. where energy is shared evenly between 'valence' partons. For instance, $\mathcal{P}(\chi) = 2\,(1-\chi)$ for the energy fraction taken by the q in a $\mathrm{q}+\mathrm{qq}$ remnant. In the other extreme, an uneven distribution could be used, like in parton distributions, where the quark only takes a small fraction and most is retained by the diquark. The default for a $\mathrm{q}+\mathrm{qq}$ remnant is of an intermediate type,

$$\mathcal{P}(\chi) \propto \frac{(1-\chi)^3}{\sqrt[4]{\chi^2 + c_\mathrm{min}^2}} \,, \tag{185}$$

with $c_\mathrm{min} = 2\langle m_\mathrm{q} \rangle / E_\mathrm{cm} = (0.6\,\mathrm{GeV})/E_\mathrm{cm}$ providing a lower cut-off. The default when a hadron is split off to leave a quark or diquark remnant is to use the standard Lund symmetric fragmentation function. In general, the more uneven the sharing of the energy, the less the total multiplicity in the beam remnant fragmentation. If no multiple interactions are allowed, a rather even sharing is needed to come close to the experimental multiplicity (and yet one does not quite make it). With an uneven sharing there is room to generate more of the total multiplicity by multiple interactions [Sjö87a].

In a photon beam, with a remnant $\mathrm{q}+\overline{\mathrm{q}}$, the χ variable is chosen the same way it would have been in a corresponding meson remnant.

Before the χ variable is used to assign remnant momenta, it is also necessary to consider the issue of primordial $k_\perp$. The initiator partons are thus assigned each a $k_\perp$ value, vanishing for an electron or photon inside an electron, distributed either according to a Gaussian or an exponential shape for a hadron, and according to either of these shapes or a power-like shape for a quark or gluon inside a photon (which may in its turn be inside an electron). The interaction subsystem is boosted and rotated to bring it from the frame assumed so far, with each initiator along the $\pm z$ axis, to one where the initiators have the required primordial $k_\perp$ values.

The $p_\perp$ recoil is taken by the remnant. If the remnant is composite, the recoil is evenly split between the two. In addition, however, the two beam remnants may be given a relative $p_\perp$, which is then always chosen as for $\mathrm{q}_i\overline{\mathrm{q}}_i$ pairs in the fragmentation description.

The χ variable is interpreted as a sharing of light-cone energy and momentum, i.e. $E + p_z$ for the beam moving in the $+z$ direction and $E - p_z$ for the other one. When the two transverse

masses $m_{\perp 1}$ and $m_{\perp 2}$ of a composite remnant have been constructed, the total transverse mass can therefore be found as

$$m_{\perp}^2 = \frac{m_{\perp 1}^2}{\chi} + \frac{m_{\perp 2}^2}{1-\chi} \,, \tag{186}$$

if remnant 1 is the one that takes the fraction χ. The choice of a light-cone interpretation to χ means the definition is invariant under longitudinal boosts, and therefore does not depend on the beam energy itself. A χ value close to the naïve borders 0 or 1 can lead to an unreasonably large remnant $m_{\perp}$. Therefore an additional check is introduced, that the remnant $m_{\perp}$ be smaller than the naïve c.m. frame remnant energy, $(1-x)E_{\mathrm{cm}}/2$. If this is not the case, a new χ and a new relative transverse momentum is selected.

Whether there is one remnant parton or two, the transverse mass of the remnant is not likely to agree with $1-x$ times the mass of the beam particle, i.e. it is not going to be possible to preserve the energy and momentum in each remnant separately. One therefore allows a shuffling of energy and momentum between the beam remnants from each of the two incoming beams. This may be achieved by performing a (small) longitudinal boost of each remnant system. Since there are two boost degrees of freedom, one for each remnant, and two constraints, one for energy and one for longitudinal momentum, a solution may be found.

Under some circumstances, one beam remnant may be absent or of very low energy, while the other one is more complicated. One example is Deeply Inelastic Scattering in ep collisions, where the electron leaves no remnant, or maybe only a low-energy photon. It is clearly then not possible to balance the two beam remnants against each other. Therefore, if one beam remnant has an energy below 0.2 of the beam energy, i.e. if the initiator parton has $x > 0.8$, then the two boosts needed to ensure energy and momentum conservation are instead performed on the other remnant and on the interaction subsystem. If there is a low-energy remnant at all then, before that, energy and momentum are assigned to the remnant constituent(s) so that the appropriate light-cone combination $E \pm p_z$ is conserved, but not energy or momentum separately. If both beam remnants have low energy, but both still exist, then the one with lower $m_{\perp}/E$ is the one that will not be boosted.

11.2 Multiple Interactions

In this section we present the model [Sjö87a] used in PYTHIA to describe the possibility that several parton pairs undergo hard interactions in a hadron–hadron collision, and thereby contribute to the overall event activity, in particular at low $p_{\perp}$. The same model is also used to describe the VMD γp events, where the photon interacts like a hadron. It should from the onset be made clear that this is not an easy topic. In fact, in the full event generation process, probably no other area is as poorly understood as this one. The whole concept of multiple interactions has been very controversial, with contradictory experimental conclusions [AFS87], but a recent CDF study [CDF97] has now started to bring more general acceptance.

The multiple interactions scenario presented here [Sjö87a] was the first detailed model for this kind of physics, and is still one of the very few available. We will present two related but separate scenarios, one 'simple' model and one somewhat more sophisticated. In fact, neither of them are all that simple, which may make the models look unattractive. However, the world of hadron physics *is* complicated, and if we err, it is most likely in being too unsophisticated. The experience gained with the model(s), in failures as well as successes, could be used as a guideline in the evolution of yet more detailed models.

Our basic philosophy will be as follows. The total rate of parton–parton interactions, as a function of the transverse momentum scale $p_{\perp}$, is assumed to be given by perturbative QCD. This is certainly true for reasonably large $p_{\perp}$ values, but here we shall also extend the perturbative parton–parton scattering framework into the low-$p_{\perp}$ region. A regularization of the divergence in the cross section for $p_{\perp} \to 0$ has to be introduced, however, which will provide us with the main free parameter of the model. Since each incoming hadron is a composite

object, consisting of many partons, there should exist the possibility of several parton pairs interacting when two hadrons collide. It is not unreasonable to assume that the different pairwise interactions take place essentially independently of each other, and that therefore the number of interactions in an event is given by a Poissonian distribution. This is the strategy of the 'simple' scenario.

Furthermore, hadrons are not only composite but also extended objects, meaning that collisions range from very central to rather peripheral ones. Reasonably, the average number of interactions should be larger in the former than in the latter case. Whereas the assumption of a Poissonian distribution should hold for each impact parameter separately, the distribution in number of interactions should be widened by the spread of impact parameters. The amount of widening depends on the assumed matter distribution inside the colliding hadrons. In the 'complex' scenario, different matter distributions are therefore introduced.

11.2.1 The basic cross sections

The QCD cross section for hard $2 \to 2$ processes, as a function of the $p_\perp^2$ scale, is given by

$$\frac{\mathrm{d}\sigma}{\mathrm{d}p_\perp^2} = \sum_{i,j,k} \int \mathrm{d}x_1 \int \mathrm{d}x_2 \int \mathrm{d}\hat{t}\, f_i(x_1,Q^2)\, f_j(x_2,Q^2)\, \frac{\mathrm{d}\hat{\sigma}_{ij}^k}{\mathrm{d}\hat{t}}\, \delta\left(p_\perp^2 - \frac{\hat{t}\hat{u}}{\hat{s}}\right) , \tag{187}$$

cf. section 7.2. Implicitly, from now on we are assuming that the 'hardness' of processes is given by the $p_\perp$ scale of the scattering. For an application of the formula above to small $p_\perp$ values, a number of caveats could be made. At low $p_\perp$, the integrals receive major contributions from the small-x region, where parton distributions are poorly understood theoretically (Regge limit behaviour, dense packing problems etc. [Lev90]) and not yet measured. Different sets of parton distributions can therefore give numerically rather different results for the phenomenology of interest. One may also worry about higher-order corrections to the jet rates, K factors, beyond what is given by parton-shower corrections — one simple option we allow here is to evaluate α_s of the hard scattering process at an optimized scale, such as $\alpha_s(0.075p_\perp^2)$ [Ell86].

The hard scattering cross section above some given $p_{\perp\mathrm{min}}$ is given by

$$\sigma_{\mathrm{hard}}(p_{\perp\mathrm{min}}) = \int_{p_{\perp\mathrm{min}}^2}^{s/4} \frac{\mathrm{d}\sigma}{\mathrm{d}p_\perp^2}\, \mathrm{d}p_\perp^2 \ . \tag{188}$$

Since the differential cross section diverges roughly like $\mathrm{d}p_\perp^2/p_\perp^4$, σ_{hard} is also divergent for $p_{\perp\mathrm{min}} \to 0$. We may compare this with the total inelastic, non-diffractive cross section $\sigma_{\mathrm{nd}}(s)$ — elastic and diffractive events are not the topic of this section. At current collider energies $\sigma_{\mathrm{hard}}(p_{\perp\mathrm{min}})$ becomes comparable with σ_{nd} for $p_{\perp\mathrm{min}} \approx 1.5$–$2$ GeV. This need not lead to contradictions: σ_{hard} does not give the hadron–hadron cross section but the parton–parton one. Each of the incoming hadrons may be viewed as a beam of partons, with the possibility of having several parton–parton interactions when the hadrons pass through each other. In this language, $\sigma_{\mathrm{hard}}(p_{\perp\mathrm{min}})/\sigma_{\mathrm{nd}}(s)$ is simply the average number of parton–parton scatterings above $p_{\perp\mathrm{min}}$ in an event, and this number may well be larger than unity.

While the introduction of several interactions per event is the natural consequence of allowing small $p_{\perp\mathrm{min}}$ values and hence large σ_{hard} ones, it is not the solution of $\sigma_{\mathrm{hard}}(p_{\perp\mathrm{min}})$ being divergent for $p_{\perp\mathrm{min}} \to 0$: the average $\hat{s}$ of a scattering decreases slower with $p_{\perp\mathrm{min}}$ than the number of interactions increases, so naïvely the total amount of scattered partonic energy becomes infinite. One cut-off is therefore obtained via the need to introduce proper multi-parton correlated parton distributions inside a hadron. This is not a part of the standard perturbative QCD formalism and is therefore not built into eq. (188). In practice, even correlated parton-distribution functions seems to provide too weak a cut, i.e. one is lead to a picture with too little of the incoming energy remaining in the small-angle beam jet region [Sjö87a].

A more credible reason for an effective cut-off is that the incoming hadrons are colour neutral objects. Therefore, when the $p_\perp$ of an exchanged gluon is made small and the transverse wavelength correspondingly large, the gluon can no longer resolve the individual colour charges, and the effective coupling is decreased. This mechanism is not in contradiction to perturbative QCD calculations, which are always performed assuming scattering of free partons (rather than partons inside hadrons), but neither does present knowledge of QCD provide an understanding of how such a decoupling mechanism would work in detail. In the simple model one makes use of a sharp cut-off at some scale $p_{\perp\mathrm{min}}$, while a more smooth dampening is assumed for the complex scenario.

One key question is the energy-dependence of $p_{\perp\mathrm{min}}$; this may be relevant e.g. for comparisons of jet rates at different Tevatron energies, and even more for any extrapolation to LHC energies. The problem actually is more pressing now than at the time of the original study [Sjö87a], since nowadays parton distributions are known to be rising more steeply at small x than the flat $xf(x)$ behaviour normally assumed for small Q^2 before HERA. This translates into a more dramatic energy dependence of the multiple-interactions rate for a fixed $p_{\perp\mathrm{min}}$.

The larger number of partons should also increase the amount of screening, however, as confirmed by toy simulations [Dis01]. As a simple first approximation, $p_{\perp\mathrm{min}}$ is assumed to increase in the same way as the total cross section, i.e. with some power $\varepsilon \approx 0.08$ [Don92] that, via reggeon phenomenology, should relate to the behaviour of parton distributions at small x and Q^2. Thus the default in PYTHIA is

$$p_{\perp\mathrm{min}}(s) = (1.9\ \mathrm{GeV}) \left(\frac{s}{1\ \mathrm{TeV}^2} \right)^{0.08} \tag{189}$$

for the simple model, with the same ansatz for $p_{\perp 0}$ in the impact-parameter-dependent approach, except that then 1.9 GeV $\rightarrow$ 2.1 GeV. At any energy scale, the simplest criterion to fix $p_{\perp\mathrm{min}}$ is to require the average charged multiplicity to agree with the experimentally determined one. In general, there is quite a strong dependence of the multiplicity on $p_{\perp\mathrm{min}}$, with a lower $p_{\perp\mathrm{min}}$ corresponding to more multiple interactions and therefore a higher multiplicity. This is the way the 1.9 GeV and 2.1 GeV numbers are fixed, based on a comparison with UA5 data in the energy range 200–900 GeV [UA584]. The energy dependence inside this range is also consistent with the chosen ansatz. However, clearly, neither the experimental nor the theoretical precision is high enough to make too strong statements. It should also be remembered that the $p_{\perp\mathrm{min}}$ values are determined within the context of a given calculation of the QCD jet cross section, and given model parameters within the multiple interactions scenario. If anything of this is changed, e.g. the parton distributions used, then $p_{\perp\mathrm{min}}$ ought to be retuned accordingly.

11.2.2 The simple model

In an event with several interactions, it is convenient to impose an ordering. The logical choice is to arrange the scatterings in falling sequence of $x_\perp = 2p_\perp/E_{\mathrm{cm}}$. The 'first' scattering is thus the hardest one, with the 'subsequent' ('second', 'third', etc.) successively softer. It is important to remember that this terminology is in no way related to any picture in physical time; we do not know anything about the latter. In principle, all the scatterings that occur in an event must be correlated somehow, naïvely by momentum and flavour conservation for the partons from each incoming hadron, less naïvely by various quantum mechanical effects. When averaging over all configurations of soft partons, however, one should effectively obtain the standard QCD phenomenology for a hard scattering, e.g. in terms of parton distributions. Correlation effects, known or estimated, can be introduced in the choice of subsequent scatterings, given that the 'preceding' (harder) ones are already known.

With a total cross section of hard interactions $\sigma_{\mathrm{hard}}(p_{\perp\mathrm{min}})$ to be distributed among $\sigma_{\mathrm{nd}}(s)$ (non-diffractive, inelastic) events, the average number of interactions per event is just the ratio $\overline{n} = \sigma_{\mathrm{hard}}(p_{\perp\mathrm{min}})/\sigma_{\mathrm{nd}}(s)$. As a starting point we will assume that all hadron collisions are

equivalent (no impact parameter dependence), and that the different parton–parton interactions take place completely independently of each other. The number of scatterings per event is then distributed according to a Poissonian with mean $\overline{n}$. A fit to Sp$\overline{\text{p}}$S collider multiplicity data [UA584] gave $p_{\perp\text{min}} \approx 1.6$ GeV, which corresponds to $\overline{n} \approx 1$. For Monte Carlo generation of these interactions it is useful to define

$$f(x_\perp) = \frac{1}{\sigma_{\text{nd}}(s)} \frac{\mathrm{d}\sigma}{\mathrm{d}x_\perp} \; , \tag{190}$$

with $\mathrm{d}\sigma/\mathrm{d}x_\perp$ obtained by analogy with eq. (187). Then $f(x_\perp)$ is simply the probability to have a parton–parton interaction at $x_\perp$, given that the two hadrons undergo a non-diffractive, inelastic collision.

The probability that the hardest interaction, i.e. the one with highest $x_\perp$, is at $x_{\perp 1}$, is now given by

$$f(x_{\perp 1}) \exp\left\{ -\int_{x_{\perp 1}}^{1} f(x'_\perp)\, \mathrm{d}x'_\perp \right\} \; , \tag{191}$$

i.e. the naïve probability to have a scattering at $x_{\perp 1}$ multiplied by the probability that there was no scattering with $x_\perp$ larger than $x_{\perp 1}$. This is the familiar exponential dampening in radioactive decays, encountered e.g. in parton showers in section 10.1.2. Using the same technique as in the proof of the veto algorithm, section 4.2, the probability to have an i:th scattering at an $x_{\perp i} < x_{\perp i-1} < \cdots < x_{\perp 1} < 1$ is found to be

$$f(x_{\perp i}) \frac{1}{(i-1)!} \left(\int_{x_{\perp i}}^{1} f(x'_\perp)\, \mathrm{d}x'_\perp \right)^{i-1} \exp\left\{ -\int_{x_{\perp i}}^{1} f(x'_\perp)\, \mathrm{d}x'_\perp \right\} \; . \tag{192}$$

The total probability to have a scattering at a given $x_\perp$, irrespectively of it being the first, the second or whatever, obviously adds up to give back $f(x_\perp)$. The multiple interaction formalism thus retains the correct perturbative QCD expression for the scattering probability at any given $x_\perp$.

With the help of the integral

$$F(x_\perp) = \int_{x_\perp}^{1} f(x'_\perp)\, \mathrm{d}x'_\perp = \frac{1}{\sigma_{\text{nd}}(s)} \int_{s x_\perp^2/4}^{s/4} \frac{\mathrm{d}\sigma}{\mathrm{d}p_\perp^2}\, \mathrm{d}p_\perp^2 \tag{193}$$

(where we assume $F(x_\perp) \to \infty$ for $x_\perp \to 0$) and its inverse F^{-1}, the iterative procedure to generate a chain of scatterings $1 > x_{\perp 1} > x_{\perp 2} > \cdots > x_{\perp i}$ is given by

$$x_{\perp i} = F^{-1}(F(x_{\perp i-1}) - \ln R_i) \; . \tag{194}$$

Here the R_i are random numbers evenly distributed between 0 and 1. The iterative chain is started with a fictitious $x_{\perp 0} = 1$ and is terminated when $x_{\perp i}$ is smaller than $x_{\perp\text{min}} = 2p_{\perp\text{min}}/E_{\text{cm}}$. Since F and F^{-1} are not known analytically, the standard veto algorithm is used to generate a much denser set of $x_\perp$ values, whereof only some are retained in the end. In addition to the $p_\perp^2$ of an interaction, it is also necessary to generate the other flavour and kinematics variables according to the relevant matrix elements.

Whereas the ordinary parton distributions should be used for the hardest scattering, in order to reproduce standard QCD phenomenology, the parton distributions to be used for subsequent scatterings must depend on all preceding x values and flavours chosen. We do not know enough about the hadron wave function to write down such joint probability distributions. To take into account the energy 'already' used in harder scatterings, a conservative approach is to evaluate the parton distributions, not at x_i for the i:th scattered parton from hadron, but at the rescaled value

$$x'_i = \frac{x_i}{\sum_{j=1}^{i-1} x_j} \; . \tag{195}$$

This is our standard procedure in the program; we have tried a few alternatives without finding any significantly different behaviour in the final physics.

In a fraction $\exp(-F(x_{\perp\mathrm{min}}))$ of the events studied, there will be no hard scattering above $x_{\perp\mathrm{min}}$ when the iterative procedure in eq. (194) is applied. It is therefore also necessary to have a model for what happens in events with no (semi)hard interactions. The simplest possible way to produce an event is to have an exchange of a very soft gluon between the two colliding hadrons. Without (initially) affecting the momentum distribution of partons, the 'hadrons' become colour octet objects rather than colour singlet ones. If only valence quarks are considered, the colour octet state of a baryon can be decomposed into a colour triplet quark and an antitriplet diquark. In a baryon-baryon collision, one would then obtain a two-string picture, with each string stretched from the quark of one baryon to the diquark of the other. A baryon-antibaryon collision would give one string between a quark and an antiquark and another one between a diquark and an antidiquark.

In a hard interaction, the number of possible string drawings are many more, and the overall situation can become quite complex when several hard scatterings are present in an event. Specifically, the string drawing now depends on the relative colour arrangement, in each hadron individually, of the partons that are about to scatter. This is a subject about which nothing is known. To make matters worse, the standard string fragmentation description would have to be extended, to handle events where two or more valence quarks have been kicked out of an incoming hadron by separate interactions. In particular, the position of the baryon number would be unclear. We therefore here assume that, following the hardest interaction, all subsequent interactions belong to one of three classes.

- Scatterings of the $\mathrm{gg} \to \mathrm{gg}$ type, with the two gluons in a colour-singlet state, such that a double string is stretched directly between the two outgoing gluons, decoupled from the rest of the system.
- Scatterings $\mathrm{gg} \to \mathrm{gg}$, but colour correlations assumed to be such that each of the gluons is connected to one of the strings 'already' present. Among the different possibilities of connecting the colours of the gluons, the one which minimizes the total increase in string length is chosen. This is in contrast to the previous alternative, which roughly corresponds to a maximization (within reason) of the extra string length.
- Scatterings $\mathrm{gg} \to \mathrm{q\overline{q}}$, with the final pair again in a colour-singlet state, such that a single string is stretched between the outgoing q and $\overline{\mathrm{q}}$.

By default, the three possibilities are assumed equally probable. Note that the total jet rate is maintained at its nominal value, i.e. scatterings such as $\mathrm{qg} \to \mathrm{qg}$ are included in the cross section, but are replaced by a mixture of gg and $\mathrm{q\overline{q}}$ events for string drawing issues. Only the hardest interaction is guaranteed to give strings coupled to the beam remnants. One should not take this approach to colour flow too seriously — clearly it is a simplification — but the overall picture does not tend to be very dependent on the particular choice you make.

Since a $\mathrm{gg} \to \mathrm{gg}$ or $\mathrm{q\overline{q}}$ scattering need not remain of this character if initial- and final-state showers were to be included (e.g. it could turn into a qg-initiated process), radiation is only included for the hardest interaction. In practice, this is not a serious problem: except for the hardest interaction, which can be hard because of experimental trigger conditions, it is unlikely for a parton scattering to be so hard that radiation plays a significant rôle.

In events with multiple interactions, the beam remnant treatment is slightly modified. First the hard scattering is generated, with its associated initial- and final-state radiation, and next any additional multiple interactions. Only thereafter are beam remnants attached to the initiator partons of the hardest scattering, using the same machinery as before, except that the energy and momentum already taken away from the beam remnants also include that of the subsequent interactions.

11.2.3 A model with varying impact parameters

Up to this point, it has been assumed that the initial state is the same for all hadron collisions, whereas in fact each collision also is characterized by a varying impact parameter b. Within the classical framework of the model reviewed here, b is to be thought of as a distance of closest approach, not as the Fourier transform of the momentum transfer. A small b value corresponds to a large overlap between the two colliding hadrons, and hence an enhanced probability for multiple interactions. A large b, on the other hand, corresponds to a grazing collision, with a large probability that no parton–parton interactions at all take place.

In order to quantify the concept of hadronic matter overlap, one may assume a spherically symmetric distribution of matter inside the hadron, $\rho(\mathbf{x})\,\mathrm{d}^3x = \rho(r)\,\mathrm{d}^3x$. For simplicity, the same spatial distribution is taken to apply for all parton species and momenta. Several different matter distributions have been tried, and are available. We will here concentrate on the most extreme one, a double Gaussian

$$\rho(r) \propto \frac{1-\beta}{a_1^3}\exp\left\{-\frac{r^2}{a_1^2}\right\} + \frac{\beta}{a_2^3}\exp\left\{-\frac{r^2}{a_2^2}\right\} \; . \tag{196}$$

This corresponds to a distribution with a small core region, of radius a_2 and containing a fraction β of the total hadronic matter, embedded in a larger hadron of radius a_1. While it is mathematically convenient to have the origin of the two Gaussians coinciding, the physics could well correspond to having three disjoint core regions, reflecting the presence of three valence quarks, together carrying the fraction β of the proton momentum. One could alternatively imagine a hard hadronic core surrounded by a pion cloud. Such details would affect e.g. the predictions for the t distribution in elastic scattering, but are not of any consequence for the current topics. To be specific, the values $\beta = 0.5$ and $a_2/a_1 = 0.2$ have been picked as default values. It should be noted that the overall distance scale a_1 never enters in the subsequent calculations, since the inelastic, non-diffractive cross section $\sigma_{\mathrm{nd}}(s)$ is taken from literature rather than calculated from the $\rho(r)$.

Compared to other shapes, like a simple Gaussian, the double Gaussian tends to give larger fluctuations, e.g. in the multiplicity distribution of minimum bias events: a collision in which the two cores overlap tends to have a strongly increased activity, while ones where they do not are rather less active. One also has a biasing effect: hard processes are more likely when the cores overlap, thus hard scatterings are associated with an enhanced multiple interaction rate. This provides one possible explanation for the experimental ‘pedestal effect’.

For a collision with impact parameter b, the time-integrated overlap $O(b)$ between the matter distributions of the colliding hadrons is given by

$$\begin{aligned} O(b) &\propto \int \mathrm{d}t \int \mathrm{d}^3x\, \rho(x,y,z)\, \rho(x+b,y,z+t) \\ &\propto \frac{(1-\beta)^2}{2a_1^2}\exp\left\{-\frac{b^2}{2a_1^2}\right\} + \frac{2\beta(1-\beta)}{a_1^2+a_2^2}\exp\left\{-\frac{b^2}{a_1^2+a_2^2}\right\} + \frac{\beta^2}{2a_2^2}\exp\left\{-\frac{b^2}{2a_2^2}\right\} \; . \end{aligned} \tag{197}$$

The necessity to use boosted $\rho(\mathbf{x})$ distributions has been circumvented by a suitable scale transformation of the z and t coordinates.

The overlap $O(b)$ is obviously strongly related to the eikonal $\Omega(b)$ of optical models. We have kept a separate notation, since the physics context of the two is slightly different: $\Omega(b)$ is based on the quantum mechanical scattering of waves in a potential, and is normally used to describe the elastic scattering of a hadron-as-a-whole, while $O(b)$ comes from a purely classical picture of point-like partons distributed inside the two colliding hadrons. Furthermore, the normalization and energy dependence is differently realized in the two formalisms.

The larger the overlap $O(b)$ is, the more likely it is to have interactions between partons in the two colliding hadrons. In fact, there should be a linear relationship

$$\langle \tilde{n}(b) \rangle = k O(b) \; , \tag{198}$$

where $\tilde{n} = 0, 1, 2, \ldots$ counts the number of interactions when two hadrons pass each other with an impact parameter b. The constant of proportionality, k, is related to the parton–parton cross section and hence increases with c.m. energy.

For each given impact parameter, the number of interactions is assumed to be distributed according to a Poissonian. If the matter distribution has a tail to infinity (as the double Gaussian does), events may be obtained with arbitrarily large b values. In order to obtain finite total cross sections, it is necessary to assume that each event contains at least one semi-hard interaction. The probability that two hadrons, passing each other with an impact parameter b, will actually undergo a collision is then given by

$$\mathcal{P}_{\text{int}}(b) = 1 - \exp(-\langle \tilde{n}(b) \rangle) = 1 - \exp(-kO(b)) \ , \tag{199}$$

according to Poissonian statistics. The average number of interactions per event at impact parameter b is now

$$\langle n(b) \rangle = \frac{\langle \tilde{n}(b) \rangle}{\mathcal{P}_{\text{int}}(b)} = \frac{kO(b)}{1 - \exp(-kO(b))} \ , \tag{200}$$

where the denominator comes from the removal of hadron pairs which pass without colliding, i.e. with $\tilde{n} = 0$.

The relationship $\langle n \rangle = \sigma_{\text{hard}}/\sigma_{\text{nd}}$ was earlier introduced for the average number of interactions per non-diffractive, inelastic event. When averaged over all impact parameters, this relation must still hold true: the introduction of variable impact parameters may give more interactions in some events and less in others, but it does not affect either σ_{hard} or σ_{nd}. For the former this is because the perturbative QCD calculations only depend on the total parton flux, for the latter by construction. Integrating eq. (200) over b, one then obtains

$$\langle n \rangle = \frac{\int \langle n(b) \rangle \, \mathcal{P}_{\text{int}}(b) \, \mathrm{d}^2 b}{\int \mathcal{P}_{\text{int}}(b) \, \mathrm{d}^2 b} = \frac{\int kO(b) \, \mathrm{d}^2 b}{\int (1 - \exp(-kO(b))) \, \mathrm{d}^2 b} = \frac{\sigma_{\text{hard}}}{\sigma_{\text{nd}}} \ . \tag{201}$$

For $O(b)$, σ_{hard} and σ_{nd} given, with $\sigma_{\text{hard}}/\sigma_{\text{nd}} > 1$, k can thus always be found (numerically) by solving the last equality.

The absolute normalization of $O(b)$ is not interesting in itself, but only the relative variation with impact parameter. It is therefore useful to introduce an 'enhancement factor' $e(b)$, which gauges how the interaction probability for a passage with impact parameter b compares with the average, i.e.

$$\langle \tilde{n}(b) \rangle = kO(b) = e(b) \, \langle kO(b) \rangle \ . \tag{202}$$

The definition of the average $\langle kO(b) \rangle$ is a bit delicate, since the average number of interactions per event is pushed up by the requirement that each event contain at least one interaction. However, an exact meaning can be given [Sjö87a].

With the knowledge of $e(b)$, the $f(x_\perp)$ function of the simple model generalizes to

$$f(x_\perp, b) = e(b) \, f(x_\perp) \ . \tag{203}$$

The naïve generation procedure is thus to pick a b according to the phase space $\mathrm{d}^2 b$, find the relevant $e(b)$ and plug in the resulting $f(x_\perp, b)$ in the formalism of the simple model. If at least one hard interaction is generated, the event is retained, else a new b is to be found. This algorithm would work fine for hadronic matter distributions which vanish outside some radius, so that the $\mathrm{d}^2 b$ phase space which needs to be probed is finite. Since this is not true for the distributions under study, it is necessary to do better.

By analogy with eq. (191), it is possible to ask what the probability is to find the hardest scattering of an event at $x_{\perp 1}$. For each impact parameter separately, the probability to have an interaction at $x_{\perp 1}$ is given by $f(x_\perp, b)$, and this should be multiplied by the probability that the

event contains no interactions at a scale $x'_\perp > x_{\perp 1}$, to yield the total probability distribution

$$\begin{aligned} \frac{\mathrm{d}\mathcal{P}_{\mathrm{hardest}}}{\mathrm{d}^2 b\,\mathrm{d}x_{\perp 1}} &= f(x_{\perp 1}, b)\,\exp\left\{-\int_{x_{\perp 1}}^{1} f(x'_\perp, b)\,\mathrm{d}x'_\perp\right\} \\ &= e(b)\,f(x_{\perp 1})\,\exp\left\{-e(b)\int_{x_{\perp 1}}^{1} f(x'_\perp)\,\mathrm{d}x'_\perp\right\}\,. \end{aligned} \tag{204}$$

If the treatment of the exponential is deferred for a moment, the distribution in b and $x_{\perp 1}$ appears in factorized form, so that the two can be chosen independently of each other. In particular, a high-$p_\perp$ QCD scattering or any other hard scattering can be selected with whatever kinematics desired for that process, and thereafter assigned some suitable 'hardness' $x_{\perp 1}$. With the b chosen according to $e(b)\,\mathrm{d}^2 b$, the neglected exponential can now be evaluated, and the event retained with a probability proportional to it. From the $x_{\perp 1}$ scale of the selected interaction, a sequence of softer $x_{\perp i}$ values may again be generated as in the simple model, using the known $f(x_\perp, b)$. This sequence may be empty, i.e. the event need not contain any further interactions.

It is interesting to understand how the algorithm above works. By selecting b according to $e(b)\,\mathrm{d}^2 b$, i.e. $O(b)\,\mathrm{d}^2 b$, the primary b distribution is maximally biased towards small impact parameters. If the first interaction is hard, by choice or by chance, the integral of the cross section above $x_{\perp 1}$ is small, and the exponential close to unity. The rejection procedure is therefore very efficient for all standard hard processes in the program — one may even safely drop the weighting with the exponential completely. The large $e(b)$ value is also likely to lead to the generation of many further, softer interactions. If, on the other hand, the first interaction is not hard, the exponential is no longer close to unity, and many events are rejected. This pulls down the efficiency for 'minimum bias' event generation. Since the exponent is proportional to $e(b)$, a large $e(b)$ leads to an enhanced probability for rejection, whereas the chance of acceptance is larger with a small $e(b)$. Among events where the hardest interaction is soft, the b distribution is therefore biased towards larger values (smaller $e(b)$), and there is a small probability for yet softer interactions.

To evaluate the exponential factor, the program pretabulates the integral of $f(x_\perp)$ at the initialization stage, and further increases the Monte Carlo statistics of this tabulation as the run proceeds. The $x_\perp$ grid is concentrated towards small $x_\perp$, where the integral is large. For a selected $x_{\perp 1}$ value, the $f(x_\perp)$ integral is obtained by interpolation. After multiplication by the known $e(b)$ factor, the exponential factor may be found.

In this section, nothing has yet been assumed about the form of the $\mathrm{d}\sigma/\mathrm{d}p_\perp$ spectrum. Like in the impact parameter independent case, it is possible to use a sharp cut-off at some given $p_{\perp\mathrm{min}}$ value. However, now each event is required to have at least one interaction, whereas before events without interactions were retained and put at $p_\perp = 0$. It is therefore aesthetically more appealing to assume a gradual turn-off, so that a (semi)hard interaction can be rather soft part of the time. The matrix elements roughly diverge like $\alpha_\mathrm{s}(p_\perp^2)\,\mathrm{d}p_\perp^2/p_\perp^4$ for $p_\perp \to 0$. They could therefore be regularized as follows. Firstly, to remove the $1/p_\perp^4$ behaviour, multiply by a factor $p_\perp^4/(p_\perp^2 + p_{\perp 0}^2)^2$. Secondly, replace the $p_\perp^2$ argument in α_s by $p_\perp^2 + p_{\perp 0}^2$. If one has included a K factor by a rescaling of the α_s argument, as mentioned earlier, replace $0.075\,p_\perp^2$ by $0.075\,(p_\perp^2 + p_{\perp 0}^2)$.

With these substitutions, a continuous $p_\perp$ spectrum is obtained, stretching from $p_\perp = 0$ to $E_\mathrm{cm}/2$. For $p_\perp \gg p_{\perp 0}$ the standard perturbative QCD cross section is recovered, while values $p_\perp \ll p_{\perp 0}$ are strongly damped. The $p_{\perp 0}$ scale, which now is the main free parameter of the model, in practice comes out to be of the same order of magnitude as the sharp cut-off $p_{\perp\mathrm{min}}$ did, i.e. 1.5–2 GeV, but typically about 10% higher.

Above we have argued that $p_{\perp\mathrm{min}}$ and $p_{\perp 0}$ should only have a slow energy dependence, and even allowed for the possibility of fixed values. For the impact parameter independent picture this works out fine, with all events being reduced to low-$p_\perp$ two-string ones when the c.m. energy is reduced. In the variable impact parameter picture, the whole formalism only makes

sense if $\sigma_{\rm hard} > \sigma_{\rm nd}$, see e.g. eq. (201). Since $\sigma_{\rm nd}$ does not vanish with decreasing energy, but $\sigma_{\rm hard}$ would do that for a fixed $p_{\perp 0}$, this means that $p_{\perp 0}$ has to be reduced significantly at low energies, even more than implied by our assumed energy dependence. The more 'sophisticated' model of this section therefore makes sense at collider energies, whereas it is not well suited for applications at fixed-target energies. There one should presumably attach to a picture of multiple soft Pomeron exchanges.

11.3 Pile-up Events

In high luminosity colliders, there is a non-negligible probability that one single bunch crossing may produce several separate events, so-called pile-up events. This in particular applies to future pp colliders like LHC, but one could also consider e.g. e^+e^- colliders with high rates of $\gamma\gamma$ collisions. The program therefore contains an option, currently only applicable to hadron–hadron collisions, wherein several events may be generated and put one after the other in the event record, to simulate the full amount of particle production a detector might be facing.

The program needs to know the assumed luminosity per bunch–bunch crossing, expressed in mb^{-1}. Multiplied by the cross section for pile-up processes studied, $\sigma_{\rm pile}$, this gives the average number of collisions per beam crossing, $\overline{n}$. These pile-up events are taken to be of the minimum bias type, with diffractive and elastic events included or not (and a further subdivision of diffractive events into single and double). This means that $\sigma_{\rm pile}$ may be either $\sigma_{\rm tot}$, $\sigma_{\rm tot} - \sigma_{\rm el}$ or $\sigma_{\rm tot} - \sigma_{\rm el} - \sigma_{\rm diffr}$. Which option to choose depends on the detector: most detectors would not be able to observe elastic pp scattering, and therefore it would be superfluous to generate that kind of events. In addition, we allow for the possibility that one interaction may be of a rare kind, selected freely by you. There is no option to generate two 'rare' events in the same crossing; normally the likelihood for that kind of occurrences should be small.

If only minimum bias type events are generated, i.e. if only one cross section is involved in the problem, then the number of events in a crossing is distributed according to a Poissonian with the average number $\overline{n}$ as calculated above. The program actually will simulate only those beam crossings where at least one event occurs, i.e. not consider the fraction $\exp(-\overline{n})$ of zero-event crossings. Therefore the actually generated average number of pile-up events is $\langle n \rangle = \overline{n}/(1-\exp(-\overline{n}))$.

Now instead consider the other extreme, where one event is supposed be rare, with a cross section $\sigma_{\rm rare}$ much smaller than $\sigma_{\rm pile}$, i.e. $f \equiv \sigma_{\rm rare}/\sigma_{\rm pile} \ll 1$. The probability that a bunch crossing will give i events, whereof one of the rare kind, now is

$$\mathcal{P}_i = f\, i \exp(-\overline{n})\, \frac{\overline{n}^i}{i!} = f\, \overline{n} \exp(-\overline{n})\, \frac{\overline{n}^{i-1}}{(i-1)!}\ . \tag{205}$$

The naïve Poissonian is suppressed by a factor f, since one of the events is rare rather than of the normal kind, but enhanced by a factor i, since any one of the i events may be the rare one. As the equality shows, the probability distribution is now a Poissonian in $i-1$: in a beam crossing which produces one rare event, the multiplicity of additional pile-up events is distributed according to a Poissonian with average number $\overline{n}$. The total average number of events thus is $\langle n \rangle = \overline{n} + 1$.

Clearly, for processes with intermediate cross sections, $\overline{n}\sigma_{\rm rare}/\sigma_{\rm pile} \simeq 1$, also the average number of events will be intermediate, and it is not allowed to assume only one event to be of the 'rare' type. We do not consider that kind of situations.

Each pileup event can be assigned a separate collision vertex within the envelope provided by the colliding beams, see `MSTP(151)`. Only simple Gaussian shapes in space and time are implemented internally, however. If this is too restrictive, you would have to assign interaction points yourself, and then shift each event separately by the required amount in space and time.

When the pile-up option is used, one main limitation is that event records may become very large when several events are put one after the other, so that the space limit in the `PYJETS`

common block is reached. It is possible to expand the dimension of the common block, see MSTU(4) and MSTU(5), but only up to about 20 000 entries, which might not always be enough, e.g. for LHC. Simplifications like switching off π^0 decay may help keep down the size, but also has its limits.

For practical reasons, the program will only allow a $\overline{n}$ up to 120. The multiplicity distribution is truncated above 200, or when the probability for a multiplicity has fallen below 10^{-6}, whichever occurs sooner. Also low multiplicities with probabilities below 10^{-6} are truncated.

11.4 Common Block Variables

Of the routines used to generate beam remnants, multiple interactions and pile-up events, none are intended to be called directly by the user. The only way to regulate these aspects is therefore via the variables in the PYPARS common block.

```
COMMON/PYPARS/MSTP(200),PARP(200),MSTI(200),PARI(200)
```

Purpose: to give access to a number of status codes and parameters which regulate the performance of PYTHIA. Most parameters are described in section 9.3; here only those related to beam remnants, multiple interactions and pile-up events are described. If the default values, below denoted by (D=...), are not satisfactory, they must in general be changed before the PYINIT call. Exceptions, i.e. variables which can be changed for each new event, are denoted by (C).

MSTP(81) : (D=1) master switch for multiple interactions.
- = 0 : off.
- = 1 : on.

MSTP(82) : (D=1) structure of multiple interactions. For QCD processes, used down to $p_\perp$ values below $p_{\perp\mathrm{min}}$, it also affects the choice of structure for the one hard/semi-hard interaction.
- = 0 : simple two-string model without any hard interactions. Toy model only!
- = 1 : multiple interactions assuming the same probability in all events, with an abrupt $p_{\perp\mathrm{min}}$ cut-off at PARP(81). (With a slow energy dependence given by PARP(89) and PARP(90).)
- = 2 : multiple interactions assuming the same probability in all events, with a continuous turn-off of the cross section at $p_{\perp 0}$ =PARP(82). (With a slow energy dependence given by PARP(89) and PARP(90).)
- = 3 : multiple interactions assuming a varying impact parameter and a hadronic matter overlap consistent with a Gaussian matter distribution, with a continuous turn-off of the cross section at $p_{\perp 0}$ =PARP(82). (With a slow energy dependence given by PARP(89) and PARP(90).)
- = 4 : multiple interactions assuming a varying impact parameter and a hadronic matter overlap consistent with a double Gaussian matter distribution given by PARP(83) and PARP(84), with a continuous turn-off of the cross section at $p_{\perp 0}$ =PARP(82). (With a slow energy dependence given by PARP(89) and PARP(90).)

Note 1: For MSTP(82)$\geq$ 2 and CKIN(3)>PARP(82) (modulo the slow energy dependence noted above), cross sections given with PYSTAT(1) may be somewhat too large, since (for reasons of efficiency) the probability factor that the hard interaction is indeed the hardest in the event is not included in the cross sections. It is included in the event selection, however, so the events generated are correctly distributed. For CKIN(3) values a couple of times larger than PARP(82) this ceases to be a problem.

Note 2: The PARP(81) and PARP(82) values are sensitive to the choice of parton dis-

tributions, $\Lambda_{\rm QCD}$, etc., in the sense that a change in the latter variables leads to a net change in the multiple interaction rate, which has to be compensated by a retuning of PARP(81) or PARP(82) if one wants to keep the net multiple interaction structure the same. The default PARP(81) and PARP(82) values are consistent with the other default values give, i.e. parton distributions of the proton etc.

MSTP(83) : (D=100) number of Monte Carlo generated phase-space points per bin (whereof there are 20) in the initialization (in PYMULT) of multiple interactions for MSTP(82)≥ 2.

MSTP(86) : (D=2) requirements on multiple interactions based on the hardness scale of the main process.

= 1 : the main collision is harder than all the subsequent ones. This is the old behaviour, preserved for reasons of backwards compatibility, and most of the time quite sensible, but with dangers as follows.
The traditional multiple interactions procedure is to let the main interaction set the upper $p_\perp$ scale for subsequent multiple interactions. For QCD, this is a matter of avoiding double-counting. Other processes normally are hard, so the procedure is then also sensible. However, for a soft main interaction, further softer interactions are hardly possible, i.e. multiple interactions are more or less killed. Such a behaviour could be motivated by the rejected events instead appearing as part of the interactions underneath a normal QCD hard interaction, but in practice the latter mechanism is not implemented. (And would have been very inefficient to work with, had it been.) For MSTP(82)≥ 3 it is even worse, since also the events themselves are likely to be rejected in the impact-parameter selection stage. Thus the spectrum of main events that survive is biased, with the low-$p_\perp$, soft tail suppressed. Furthermore, even when events are rejected by the impact parameter procedure, this is not reflected in the cross section for the process, as it should have been. Results may thus be misleading.

= 2 : when the main process is of the QCD jets type (the same as those in multiple interactions) subsequent jets are requested to be softer, but for other processes no such requirement exists.

= 3 : no requirements at all that multiple interactions have to be softer than the main interactions (of dubious use for QCD processes but intended for cross-checks).

Note: process cross sections are unreliable whenever the main process does restrict subsequent interactions, and the main process can become soft. For QCD jet studies in this region it is then better to put CKIN(3)=0 and get the 'correct' total cross section.

MSTP(91) : (D=1) (C) primordial $k_\perp$ distribution in hadron. See MSTP(93) for photon.

= 0 : no primordial $k_\perp$.

= 1 : Gaussian, width given in PARP(91), upper cut-off in PARP(93).

= 2 : exponential, width given in PARP(92), upper cut-off in PARP(93).

MSTP(92) : (D=3) (C) energy partitioning in hadron or resolved photon remnant, when this remnant is split into two jets. (For a splitting into a hadron plus a jet, see MSTP(94).) The energy fraction χ taken by one of the two objects, with conventions as described for PARP(94) and PARP(96), is chosen according to the different distributions below. Here $c_{\rm min} = 0.6\ {\rm GeV}/E_{\rm cm} \approx 2\langle m_{\rm q}\rangle/E_{\rm cm}$.

= 1 : 1 for meson or resolved photon, $2(1-\chi)$ for baryon, i.e. simple counting rules.

= 2 : $(k+1)(1-\chi)^k$, with k given by PARP(94) or PARP(96).

= 3 : proportional to $(1-\chi)^k/\sqrt[4]{\chi^2+c_{\rm min}^2}$, with k given by PARP(94) or PARP(96).

= 4 : proportional to $(1-\chi)^k/\sqrt{\chi^2+c_{\rm min}^2}$, with k given by PARP(94) or PARP(96).

= 5 : proportional to $(1-\chi)^k/(\chi^2+c_{\rm min}^2)^{b/2}$, with k given by PARP(94) or PARP(96), and b by PARP(98).

MSTP(93) : (D=1) (C) primordial $k_\perp$ distribution in photon, either it is one of the incoming particles or inside an electron.

= 0 : no primordial $k_\perp$.

= 1 : Gaussian, width given in PARP(99), upper cut-off in PARP(100).

= 2 : exponential, width given in PARP(99), upper cut-off in PARP(100).

= 3 : power-like of the type $\mathrm{d}k_\perp^2/(k_{\perp 0}^2+k_\perp^2)^2$, with $k_{\perp 0}$ in PARP(99) and upper $k_\perp$ cut-off in PARP(100).

= 4 : power-like of the type $\mathrm{d}k_\perp^2/(k_{\perp 0}^2+k_\perp^2)$, with $k_{\perp 0}$ in PARP(99) and upper $k_\perp$ cut-off in PARP(100).

= 5 : power-like of the type $\mathrm{d}k_\perp^2/(k_{\perp 0}^2+k_\perp^2)$, with $k_{\perp 0}$ in PARP(99) and upper $k_\perp$ cut-off given by the $p_\perp$ of the hard process or by PARP(100), whichever is smaller.

Note: for options 1 and 2 the PARP(100) value is of minor importance, once PARP(100)$\gg$PARP(99). However, options 3 and 4 correspond to distributions with infinite $\langle k_\perp^2\rangle$ if the $k_\perp$ spectrum is not cut off, and therefore the PARP(100) value is as important for the overall distribution as is PARP(99).

MSTP(94) : (D=3) (C) energy partitioning in hadron or resolved photon remnant, when this remnant is split into a hadron plus a remainder-jet. The energy fraction chi is taken by one of the two objects, with conventions as described below or for PARP(95) and PARP(97).

= 1 : 1 for meson or resolved photon, $2(1-\chi)$ for baryon, i.e. simple counting rules.

= 2 : $(k+1)(1-\chi)^k$, with k given by PARP(95) or PARP(97).

= 3 : the χ of the hadron is selected according to the normal fragmentation function used for the hadron in jet fragmentation, see MSTJ(11). The possibility of a changed fragmentation function shape in diquark fragmentation (see PARJ(45)) is not included.

= 4 : as =3, but the shape is changed as allowed in diquark fragmentation (see PARJ(45)); this change is here also allowed for meson production. (This option is not so natural for mesons, but has been added to provide the same amount of freedom as for baryons).

MSTP(131) : (D=0) master switch for pile-up events, i.e. several independent hadron–hadron interactions generated in the same bunch–bunch crossing, with the events following one after the other in the event record.

= 0 : off, i.e. only one event is generated at a time.

= 1 : on, i.e. several events are allowed in the same event record. Information on the processes generated may be found in MSTI(41) - MSTI(50).

MSTP(132) : (D=4) the processes that are switched on for pile-up events. The first event may be set up completely arbitrarily, using the switches in the PYSUBS common block, while all the subsequent events have to be of one of the 'inclusive' processes which dominate the cross section, according to the options below. It is thus not possible to generate two rare events in the pile-up option.

= 1 : low-$p_\perp$ processes (ISUB = 95) only. The low-$p_\perp$ model actually used, both in the hard event and in the pile-up events, is the one set by MSTP(81) etc. This means that implicitly also high-$p_\perp$ jets can be generated in the pile-up events.

= 2 : low-$p_\perp$ + double diffractive processes (ISUB = 95 and 94).

= 3 : low-$p_\perp$ + double diffractive + single diffractive processes (ISUB = 95, 94, 93 and 92).

= 4 : low-$p_\perp$ + double diffractive + single diffractive + elastic processes, together

corresponding to the full hadron–hadron cross section (ISUB = 95, 94, 93, 92 and 91).

MSTP(133) : (D=0) multiplicity distribution of pile-up events.

= 0 : selected by you, before each PYEVNT call, by giving the MSTP(134) value.

= 1 : a Poissonian multiplicity distribution in the total number of pile-up events. This is the relevant distribution if the switches set for the first event in PYSUBS give the same subprocesses as are implied by MSTP(132). In that case the mean number of events per beam crossing is $\overline{n} = \sigma_{\rm pile} \times$ PARP(131), where $\sigma_{\rm pile}$ is the sum of the cross section for allowed processes. Since bunch crossing which do not give any events at all (probability $\exp(-\overline{n})$) are not simulated, the actual average number per PYEVNT call is $\langle n \rangle = \overline{n}/(1-\exp(-\overline{n}))$.

= 2 : a biased distribution, as is relevant when one of the events to be generated is assumed to belong to an event class with a cross section much smaller than the total hadronic cross section. If $\sigma_{\rm rare}$ is the cross section for this rare process (or the sum of the cross sections of several rare processes) and $\sigma_{\rm pile}$ the cross section for the processes allowed by MSTP(132), then define $\overline{n} = \sigma_{\rm pile} \times$ PARP(131) and $f = \sigma_{\rm rare}/\sigma_{\rm pile}$. The probability that a bunch crossing will give i events is then $\mathcal{P}_i = f\, i \exp(-\overline{n})\, \overline{n}^i/i!$, i.e. the naïve Poissonian is suppressed by a factor f since one of the events will be rare rather than frequent, but enhanced by a factor i since any of the i events may be the rare one. Only beam crossings which give at least one event of the required rare type are simulated, and the distribution above normalized accordingly.

Note: for practical reasons, it is required that $\overline{n} < 120$, i.e. that an average beam crossing does not contain more than 120 pile-up events. The multiplicity distribution is truncated above 200, or when the probability for a multiplicity has fallen below 10^{-6}, whichever occurs sooner. Also low multiplicities with probabilities below 10^{-6} are truncated. See also PARI(91) - PARI(93).

MSTP(134) : (D=1) a user selected multiplicity, i.e. total number of pile-up events, to be generated in the next PYEVNT call when MSTP(133)=0. May be reset for each new event, but must be in the range $1 \leq$ MSTP(134) ≤ 200.

PARP(81) : (D=1.9 GeV) effective minimum transverse momentum $p_{\perp \rm min}$ for multiple interactions with MSTP(82)=1, at the reference energy scale PARP(89), with the degree of energy rescaling given by PARP(90).

PARP(82) : (D=2.1 GeV) regularization scale $p_{\perp 0}$ of the transverse momentum spectrum for multiple interactions with MSTP(82) ≥ 2, at the reference energy scale PARP(89), with the degree of energy rescaling given by PARP(90). (Current default based on the MSTP(82)=4 option, without any change of MSTP(2) or MSTP(33).)

PARP(83), PARP(84) : (D=0.5, 0.2) parameters of an assumed double Gaussian matter distribution inside the colliding hadrons for MSTP(82)=4, of the form given in eq. (196), i.e. with a core of radius PARP(84) of the main radius and containing a fraction PARP(83) of the total hadronic matter.

PARP(85) : (D=0.33) probability that an additional interaction in the multiple interaction formalism gives two gluons, with colour connections to 'nearest neighbours' in momentum space.

PARP(86) : (D=0.66) probability that an additional interaction in the multiple interaction formalism gives two gluons, either as described in PARP(85) or as a closed gluon loop. Remaining fraction is supposed to consist of quark–antiquark pairs.

PARP(87), PARP(88) : (D=0.7, 0.5) in order to account for an assumed dominance of valence quarks at low transverse momentum scales, a probability is introduced that a gg-scattering according to naïve cross section is replaced by a $q\overline{q}$ one; this is used only for MSTP(82) ≥ 2. The probability is parameterized as $\mathcal{P} = a(1-(p_\perp^2/(p_\perp^2 + b^2)^2)$, where $a =$ PARP(87) and $b =$ PARP(88) $\times$ PARP(82) (including the slow en-

ergy rescaling of the $p_{\perp 0}$ parameter).

PARP(89) : (D=1000. GeV) reference energy scale, at which PARP(81) and PARP(82) give the $p_{\perp\mathrm{min}}$ and $p_{\perp 0}$ values directly. Has no physical meaning in itself, but is used for convenience only. (A form $p_{\perp\mathrm{min}} = \mathtt{PARP(81)} E_{\mathrm{cm}}^{\mathtt{PARP(90)}}$ would have been equally possible but then with a less transparent meaning of PARP(81).) For studies of the $p_{\perp\mathrm{min}}$ dependence at some specific energy it may be convenient to choose PARP(89) equal to this energy.

PARP(90) : (D=0.16) power of the energy-rescaling term of the $p_{\perp\mathrm{min}}$ and $p_{\perp 0}$ parameters, which are assumed proportional to $E_{\mathrm{cm}}^{\mathtt{PARP(90)}}$. The default value is inspired by the rise of the total cross section by the pomeron term, $s^{\varepsilon} = E_{\mathrm{cm}}^{2\varepsilon} = E_{\mathrm{cm}}^{2\cdot 0.08}$, which is not inconsistent with the small-x behaviour. It is also reasonably consistent with the energy-dependence implied by a comparison with the UA5 multiplicity distributions at 200 and 900 GeV [UA584]. PARP(90) = 0 is an allowed value, i.e. it is possible to have energy-independent parameters.

PARP(91) : (D=1. GeV/c) (C) width of Gaussian primordial $k_{\perp}$ distribution inside hadron for MSTP(91)=1, i.e. $\exp(-k_{\perp}^2/\sigma^2)\, k_{\perp}\, \mathrm{d}k_{\perp}$ with $\sigma =$PARP(91) and $\langle k_{\perp}^2 \rangle =$PARP(91)^2.

PARP(92) : (D=0.40 GeV/c) (C) width parameter of exponential primordial $k_{\perp}$ distribution inside hadron for MSTP(91)=2, i.e. $\exp(-k_{\perp}/\sigma)\, k_{\perp}\, \mathrm{d}k_{\perp}$ with $\sigma =$PARP(92) and $\langle k_{\perp}^2 \rangle = 6\times$PARP(92)^2. Thus one should put PARP(92)$\approx$PARP(91)$/\sqrt{6}$ to have continuity with the option above.

PARP(93) : (D=5. GeV/c) (C) upper cut-off for primordial $k_{\perp}$ distribution inside hadron.

PARP(94) : (D=1.) (C) for MSTP(92)≥ 2 this gives the value of the parameter k for the case when a meson or resolved photon remnant is split into two fragments (which is which is chosen at random).

PARP(95) : (D=0.) (C) for MSTP(94)=2 this gives the value of the parameter k for the case when a meson or resolved photon remnant is split into a meson and a spectator fragment jet, with χ giving the energy fraction taken by the meson.

PARP(96) : (D=3.) (C) for MSTP(92)≥ 2 this gives the value of the parameter k for the case when a nucleon remnant is split into a diquark and a quark fragment, with χ giving the energy fraction taken by the quark jet.

PARP(97) : (D=1.) (C) for MSTP(94)=2 this gives the value of the parameter k for the case when a nucleon remnant is split into a baryon and a quark jet or a meson and a diquark jet, with χ giving the energy fraction taken by the quark jet or meson, respectively.

PARP(98) : (D=0.75) (C) for MSTP(92)=5 this gives the power of an assumed basic $1/\chi^b$ behaviour in the splitting distribution, with $b =$PARP(98).

PARP(99) : (D=1. GeV/c) (C) width parameter of primordial $k_{\perp}$ distribution inside photon; exact meaning depends on MSTP(93) value chosen (cf. PARP(91) and PARP(92) above).

PARP(100) : (D=5. GeV/c) (C) upper cut-off for primordial $k_{\perp}$ distribution inside photon.

PARP(131) : (D=0.01 mb^{-1}) in the pile-up events scenario, PARP(131) gives the assumed luminosity per bunch–bunch crossing, i.e. if a subprocess has a cross section σ, the average number of events of this type per bunch–bunch crossing is $\overline{n} = \sigma\times$PARP(131). PARP(131) may be obtained by dividing the integrated luminosity over a given time (1 s, say) by the number of bunch–bunch crossings that this corresponds to. Since the program will not generate more than 200 pile-up events, the initialization procedure will crash if $\overline{n}$ is above 120.

12 Fragmentation

The main fragmentation option in PYTHIA is the Lund string scheme, but independent fragmentation options are also available. These latter options should not be taken too seriously, since we know that independent fragmentation does not provide a consistent alternative, but occasionally one may like to compare string fragmentation with something else.

The subsequent four sections give further details; the first one on flavour selection, which is common to the two approaches, the second on string fragmentation, the third on independent fragmentation, while the fourth and final contains information on a few other issues.

The Lund fragmentation model is described in [And83], where all the basic ideas are presented and earlier papers [And79, And80, And82, And82a] summarized. The details given there on how a multiparton jet system is allowed to fragment are out of date, however, and for this one should turn to [Sjö84]. Also the 'popcorn' baryon production mechanism is not covered, see [And85], and [Edé97] for a more sophisticated version. The most recent comprehensive description of the Lund model is found in [And98]. Reviews of fragmentation models in general may be found in [Sjö88, Sjö89].

12.1 Flavour Selection

In either string or independent fragmentation, an iterative approach is used to describe the fragmentation process. Given an initial quark $q = q_0$, it is assumed that a new $q_1\overline{q}_1$ pair may be created, such that a meson $q_0\overline{q}_1$ is formed, and a q_1 is left behind. This q_1 may at a later stage pair off with a $\overline{q}_2$, and so on. What need be given is thus the relative probabilities to produce the various possible $q_i\overline{q}_i$ pairs, $u\overline{u}$, $d\overline{d}$, $s\overline{s}$, etc., and the relative probabilities that a given $q_{i-1}\overline{q}_i$ quark pair combination forms a specific meson, e.g. for $u\overline{d}$ either π^+, ρ^+ or some higher state.

In PYTHIA, it is assumed that the two aspects can be factorized, i.e. that it is possible first to select a $q_i\overline{q}_i$ pair, without any reference to allowed physical meson states, and that, once the $q_{i-1}\overline{q}_i$ flavour combination is given, it can be assigned to a given meson state with total probability unity. Corrections to this factorized ansatz will come in the baryon sector.

12.1.1 Quark flavours and transverse momenta

In order to generate the quark–antiquark pairs $q_i\overline{q}_i$ which lead to string breakups, the Lund model invokes the idea of quantum mechanical tunnelling, as follows. If the q_i and $\overline{q}_i$ have no (common) mass or transverse momentum, the pair can classically be created at one point and then be pulled apart by the field. If the quarks have mass and/or transverse momentum, however, the q_i and $\overline{q}_i$ must classically be produced at a certain distance so that the field energy between them can be transformed into the sum of the two transverse masses $m_\perp$. Quantum mechanically, the quarks may be created in one point (so as to keep the concept of local flavour conservation) and then tunnel out to the classically allowed region. In terms of a common transverse mass $m_\perp$ of the q_i and the $\overline{q}_i$, the tunnelling probability is given by

$$\exp\left(-\frac{\pi m_\perp^2}{\kappa}\right) = \exp\left(-\frac{\pi m^2}{\kappa}\right)\exp\left(-\frac{\pi p_\perp^2}{\kappa}\right) \ . \tag{206}$$

The factorization of the transverse momentum and the mass terms leads to a flavour-independent Gaussian spectrum for the p_x and p_y components of $q_i\overline{q}_i$ pairs. Since the string is assumed to have no transverse excitations, this $p_\perp$ is locally compensated between the quark and the antiquark of the pair. The $p_\perp$ of a meson $q_{i-1}\overline{q}_i$ is given by the vector sum of the $p_\perp$:s of the q_{i-1} and $\overline{q}_i$ constituents, which implies Gaussians in p_x and p_y with a width $\sqrt{2}$ that of the quarks themselves. The assumption of a Gaussian shape may be a good first approximation, but there remains the possibility of non-Gaussian tails, that can be important in some situations.

In a perturbative QCD framework, a hard scattering is associated with gluon radiation, and further contributions to what is naïvely called fragmentation $p_\perp$ comes from unresolved radiation. This is used as an explanation why the experimental $\langle p_\perp \rangle$ is somewhat higher than obtained with the formula above.

The formula also implies a suppression of heavy quark production $u : d : s : c \approx 1 : 1 : 0.3 : 10^{-11}$. Charm and heavier quarks are hence not expected to be produced in the soft fragmentation. Since the predicted flavour suppressions are in terms of quark masses, which are notoriously difficult to assign (should it be current algebra, or constituent, or maybe something in between?), the suppression of $s\bar{s}$ production is left as a free parameter in the program: $u\bar{u} : d\bar{d} : s\bar{s} = 1 : 1 : \gamma_s$, where by default $\gamma_s = 0.3$. At least qualitatively, the experimental value agrees with theoretical prejudice. There is no production at all of heavier flavours in the fragmentation process, but only in the hard process or as part of the shower evolution.

12.1.2 Meson production

Once the flavours q_{i-1} and $\bar{q}_i$ have been selected, a choice is made between the possible multiplets. The relative composition of different multiplets is not given from first principles, but must depend on the details of the fragmentation process. To some approximation one would expect a negligible fraction of states with radial excitations or non-vanishing orbital angular momentum. Spin counting arguments would then suggest a 3:1 mixture between the lowest lying vector and pseudoscalar multiplets. Wave function overlap arguments lead to a relative enhancement of the lighter pseudoscalar states, which is more pronounced the larger the mass splitting is [And82a].

In the program, six meson multiplets are included. If the nonrelativistic classification scheme is used, i.e. mesons are assigned a valence quark spin S and an internal orbital angular momentum L, with the physical spin s denoted J, $\mathbf{J} = \mathbf{L} + \mathbf{S}$, then the multiplets are:

- $L = 0$, $S = 0$, $J = 0$: the ordinary pseudoscalar meson multiplet;
- $L = 0$, $S = 1$, $J = 1$: the ordinary vector meson multiplet;
- $L = 1$, $S = 0$, $J = 1$: an axial vector meson multiplet;
- $L = 1$, $S = 1$, $J = 0$: the scalar meson multiplet;
- $L = 1$, $S = 1$, $J = 1$: another axial vector meson multiplet; and
- $L = 1$, $S = 1$, $J = 2$: the tensor meson multiplet.

Each multiplet has the full five-flavour setup of 5×5 states included in the program. Some simplifications have been made; thus there is no mixing included between the two axial vector multiplets.

In the program, the spin S is first chosen to be either 0 or 1. This is done according to parameterized relative probabilities, where the probability for spin 1 by default is taken to be 0.5 for a meson consisting only of u and d quark, 0.6 for one which contains s as well, and 0.75 for quarks with c or heavier quark, in accordance with the deliberations above.

By default, it is assumed that $L = 0$, such that only pseudoscalar and vector mesons are produced. For inclusion of $L = 1$ production, four parameters can be used, one to give the probability that a $S = 0$ state also has $L = 1$, the other three for the probability that a $S = 1$ state has $L = 1$ and J either 0, 1, or 2.

For the flavour-diagonal meson states $u\bar{u}$, $d\bar{d}$ and $s\bar{s}$, it is also necessary to include mixing into the physical mesons. This is done according to a parameterization, based on the mixing angles given in the Review of Particle Properties [PDG88]. In particular, the default choices correspond to

$$\begin{aligned} \eta &= \frac{1}{2}(u\bar{u} + d\bar{d}) - \frac{1}{\sqrt{2}} s\bar{s} \,; \\ \eta' &= \frac{1}{2}(u\bar{u} + d\bar{d}) + \frac{1}{\sqrt{2}} s\bar{s} \,; \end{aligned}$$

$$\omega = \frac{1}{\sqrt{2}}(u\bar{u} + d\bar{d})$$
$$\phi = s\bar{s}\,. \tag{207}$$

In the $\pi^0 - \eta - \eta'$ system, no account is therefore taken of the difference in masses, an approximation which seems to lead to an overestimate of η' rates [ALE92]. Therefore parameters have been introduced to allow an additional 'brute force' suppression of η and η' states.

12.1.3 Baryon production

The mechanism for meson production follows rather naturally from the simple picture of a meson as a short piece of string between two $q/\bar{q}$ endpoints. There is no unique recipe to generalize this picture to baryons. The program actually contains three different scenarios: diquark, simple popcorn, and advanced popcorn. In the diquark model the baryon and antibaryon are always produced as nearest neighbours along the string, while mesons may (but need not) be produced in between in the popcorn scenarios. The simpler popcorn alternative admits at most one intermediate meson, while the advanced one allows many. Further differences may be found, but several aspects are also common between the three scenarios. Below they are therefore described in order of increasing sophistication. Finally the application of the models to baryon remnant fragmentation, where a diquark originally sits at one endpoint of the string, is discussed.

Diquark picture

Baryon production may, in its simplest form, be obtained by assuming that any flavour q_i given above could represent either a quark or an antidiquark in a colour triplet state. Then the same basic machinery can be run through as above, supplemented with the probability to produce various diquark pairs. In principle, there is one parameter for each diquark, but if tunnelling is still assumed to give an effective description, mass relations can be used to reduce the effective number of parameters. There are three main ones appearing in the program:

- the relative probability to pick a $\overline{qq}$ diquark rather than a q;
- the extra suppression associated with a diquark containing a strange quark (over and above the ordinary s/u suppression factor γ_s); and
- the suppression of spin 1 diquarks relative to spin 0 ones (apart from the factor of 3 enhancement of the former based on counting the number of spin states).

The extra strange diquark suppression factor comes about since what appears in the exponent of the tunnelling formula is m^2 and not m, so that the diquark and the strange quark suppressions do not factorize.

Only two baryon multiplets are included, i.e. there are no $L = 1$ excited states. The two multiplets are:

- $S = J = 1/2$: the 'octet' multiplet of **SU(3)**;
- $S = J = 3/2$: the 'decuplet' multiplet of **SU(3)**.

In contrast to the meson case, different flavour combinations have different numbers of states available: for uuu only Δ^{++}, whereas uds may become either Λ, Σ^0 or Σ^{*0}.

An important constraint is that a baryon is a symmetric state of three quarks, neglecting the colour degree of freedom. When a diquark and a quark are joined to form a baryon, the combination is therefore weighted with the probability that they form a symmetric three-quark state. The program implementation of this principle is to first select a diquark at random, with the strangeness and spin 1 suppression factors above included, but then to accept the selected diquark with a weight proportional to the number of states available for the quark-diquark combination. This means that, were it not for the tunnelling suppression factors, all states in the **SU(6)** (flavour **SU(3)** times spin **SU(2)**) 56-multiplet would become equally populated. Of course also heavier baryons may come from the fragmentation of e.g. c quark jets, but although the particle classification scheme used in the program is **SU(10)**, i.e. with five flavours, all possi-

ble quark-diquark combinations can be related to **SU(6)** by symmetry arguments. As in the case for mesons, one could imagine an explicit further suppression of the heavier spin 3/2 baryons.

In case of rejection, one again chooses between a diquark or a quark. If choosing diquark, a new baryon is selected and tested, etc. (In versions earlier than PYTHIA 6.106, the algorithm was instead to always produce a new diquark if the previous one had been rejected. However, the probability that a quark will produce a baryon and a antidiquark is then flavour independent, which is not in agreement with the model.) Calling the tunnelling factor for diquark D T_D, the number of spin states σ_D and the **SU(6)** factor for D and a quark q $SU_{D,\mathrm{q}}$, the model prediction for the $(\mathrm{q} \to B + D)/(\mathrm{q} \to M + \mathrm{q}')$ ratio is

$$S_{\mathrm{q}} = \frac{P(\mathrm{qq})}{P(\mathrm{q})} \sum_D T_D \sigma_D SU_{D,\mathrm{q}} \; . \tag{208}$$

(Neglecting this flavour dependence e.g. leads to an enhancement of the Ω^- relative to primary proton production with approximately a factor 1.2, using JETSET 7.4 default values.) Since the chosen algorithm implies the normalization $\sum_D T_D \sigma_D = 1$ and $SU_{D,\mathrm{q}} \leq 1$, the final diquark production rate is somewhat reduced from the $P(\mathrm{qq})/P(\mathrm{q})$ input value.

When a diquark has been fitted into a symmetrical three-particle state, it should not suffer any further **SU(6)** suppressions. Thus the accompanying antidiquark should 'survive' with probability 1. When producing a quark to go with a previously produced diquark, this is achieved by testing the configuration against the proper **SU(6)** factor, but in case of rejection keep the diquark and pick a new quark, which then is tested, etc.

There is no obvious corresponding algorithm available when a quark from one side and a diquark from the other are joined to form the last hadron of the string. In this case the quark is a member of a pair, in which the antiquark already has formed a specific hadron. Thus the quark flavour cannot be reselected. One could consider the **SU(6)** rejection as a major joining failure, and restart the fragmentation of the original string, but then the already accepted diquark *does* suffer extra **SU(6)** suppression. In the program the joining of a quark and a diquark is always accepted.

Simple popcorn

A more general framework for baryon production is the 'popcorn' one [And85], in which diquarks as such are never produced, but rather baryons appear from the successive production of several $\mathrm{q}_i\overline{\mathrm{q}}_i$ pairs. The picture is the following. Assume that the original q is red r and the $\overline{\mathrm{q}}$ is $\overline{r}$. Normally a new $\mathrm{q}_1\overline{\mathrm{q}}_1$ pair produced in the field would also be $r\overline{r}$, so that the $\overline{\mathrm{q}}_1$ is pulled towards the q end and vice versa, and two separate colour-singlet systems $\mathrm{q}\overline{\mathrm{q}}_1$ and $\mathrm{q}_1\overline{\mathrm{q}}$ are formed. Occasionally, the $\mathrm{q}_1\overline{\mathrm{q}}_1$ pair may be e.g. $g\overline{g}$ (g = green), in which case there is no net colour charge acting on either q_1 or $\overline{\mathrm{q}}_1$. Therefore, the pair cannot gain energy from the field, and normally would exist only as a fluctuation. If q_1 moves towards q and $\overline{\mathrm{q}}_1$ towards $\overline{\mathrm{q}}$, the net field remaining between q_1 and $\overline{\mathrm{q}}_1$ is $\overline{b}b$ (b = blue; $g + r = \overline{b}$ if only colour triplets are assumed). In this central field, an additional $\mathrm{q}_2\overline{\mathrm{q}}_2$ pair can be created, where q_2 now is pulled towards qq_1 and $\overline{\mathrm{q}}_2$ towards $\overline{\mathrm{q}}\overline{\mathrm{q}}_1$, with no net colour field between q_2 and $\overline{\mathrm{q}}_2$. If this is all that happens, the baryon B will be made up out of q_1, q_2 and some q_4 produced between q and q_1, and $\overline{B}$ of $\overline{\mathrm{q}}_1$, $\overline{\mathrm{q}}_2$ and some $\overline{\mathrm{q}}_5$, i.e. the B and $\overline{B}$ will be nearest neighbours in rank and share two quark pairs. Specifically, q_1 will gain energy from q_2 in order to end up on mass shell, and the tunnelling formula for an effective $\mathrm{q}_1\mathrm{q}_2$ diquark is recovered.

Part of the time, several $b\overline{b}$ colour pair productions may take place between the q_1 and $\overline{\mathrm{q}}_1$, however. With two production vertices $\mathrm{q}_2\overline{\mathrm{q}}_2$ and $\mathrm{q}_3\overline{\mathrm{q}}_3$, a central meson $\overline{\mathrm{q}}_2\mathrm{q}_3$ may be formed, surrounded by a baryon $\mathrm{q}_4\mathrm{q}_1\mathrm{q}_2$ and an antibaryon $\overline{\mathrm{q}}_3\overline{\mathrm{q}}_1\overline{\mathrm{q}}_5$. We call this a $BM\overline{B}$ configuration to distinguish it from the $\mathrm{q}_4\mathrm{q}_1\mathrm{q}_2$ + $\overline{\mathrm{q}}_2\overline{\mathrm{q}}_1\overline{\mathrm{q}}_5$ $B\overline{B}$ configuration above. For $BM\overline{B}$ the B and $\overline{B}$ only share one quark–antiquark pair, as opposed to two for $B\overline{B}$ configurations. The relative probability for a $BM\overline{B}$ configuration is given by the uncertainty relation suppression for having the q_1 and $\overline{\mathrm{q}}_1$ sufficiently far apart that a meson may be formed in between. The suppression of

the $BM\overline{B}$ system is estimated by

$$|\Delta_F|^2 \approx \exp(-2\mu_\perp M_\perp/\kappa) \tag{209}$$

where $\mu_\perp$ and $M_\perp$ is the transverse mass of q_1 and the meson, respectively. Strictly speaking, also configurations like $BMM\overline{B}$, $BMMM\overline{B}$, etc. should be possible, but since the total invariant $M_\perp$ grows rapidly with the number of mesons, the probability for this is small in the simple model. Further, since larger masses corresponds to longer string pieces, the production of pseudoscalar mesons is favoured over that of vector ones. If only $B\overline{B}$ and $BM\overline{B}$ states are included, and if the probability for having a vector meson M is not suppressed extra, two partly compensating errors are made (since a vector meson typically decays into two or more pseudoscalar ones).

In total, the flavour iteration procedure therefore contains the following possible subprocesses (plus, of course, their charge conjugates):

- $q_1 \to q_2 + (q_1\overline{q}_2)$ meson;
- $q_1 \to \overline{q}_2\overline{q}_3 + (q_1q_2q_3)$ baryon;
- $q_1q_2 \to \overline{q}_3 + (q_1q_2q_3)$ baryon;
- $q_1q_2 \to q_1q_3 + (q_2\overline{q}_3)$ meson;

with the constraint that the last process cannot be iterated to obtain several mesons in between the baryon and the antibaryon.

When selecting flavours for $qq \to M + qq'$, the quark coming from the accepted qq is kept, and the other member of qq', as well as the spin of qq', is chosen with weights taking **SU(6)** symmetry into account. Thus the flavour of qq is not influenced by **SU(6)** factors for qq', but the flavour of M is.

Unfortunately, the resulting baryon production model has a fair number of parameters, which would be given by the model only if quark and diquark masses were known unambiguously. We have already mentioned the s/u ratio and the qq/q one; the latter has to be increased from 0.09 to 0.10 for the popcorn model, since the total number of possible baryon production configurations is lower in this case (the particle produced between the B and $\overline{B}$ is constrained to be a meson). With the improved **SU(6)** treatment introduced in PYTHIA 6.106, a rejected $q \to B + qq$ may lead to the splitting $q \to M + q'$ instead. This calls for an increase of the qq/q input ratio by approximately 10%. For the popcorn model, exactly the same parameters as already found in the diquark model are needed to describe the $B\overline{B}$ configurations. For $BM\overline{B}$ configurations, the square root of a suppression factor should be applied if the factor is relevant only for one of the B and $\overline{B}$, e.g. if the B is formed with a spin 1 'diquark' q_1q_2 but the $\overline{B}$ with a spin 0 diquark $\overline{q}_1\overline{q}_3$. Additional parameters include the relative probability for $BM\overline{B}$ configurations, which is assumed to be roughly 0.5 (with the remaining 0.5 being $B\overline{B}$), a suppression factor for having a strange meson M between the B and $\overline{B}$ (as opposed to having a lighter non-strange one) and a suppression factor for having a $s\overline{s}$ pair (rather than a $u\overline{u}$ one) shared between the B and $\overline{B}$ of a $BM\overline{B}$ configuration. The default parameter values are based on a combination of experimental observation and internal model predictions.

In the diquark model, a diquark is expected to have exactly the same transverse momentum distribution as a quark. For $BM\overline{B}$ configurations the situation is somewhat more unclear, but we have checked that various possibilities give very similar results. The option implemented in the program is to assume no transverse momentum at all for the $q_1\overline{q}_1$ pair shared by the B and $\overline{B}$, with all other pairs having the standard Gaussian spectrum with local momentum conservation. This means that the B and $\overline{B}$ $p_\perp$:s are uncorrelated in a $BM\overline{B}$ configuration and (partially) anticorrelated in the $B\overline{B}$ configurations, with the same mean transverse momentum for primary baryons as for primary mesons.

Advanced popcorn

In [Edé97], a revised popcorn model is presented, where the separate production of the quarks in an effective diquark is taken more seriously. The production of a $q\overline{q}$ pair which breaks the string is in this model determined by eq. (206), also when ending up in a diquark.

Furthermore, the popcorn model is re-implemented in such a way that eq (209) could be used explicitly in the Monte Carlo. The two parameters

$$\beta_q \equiv 2\left\langle \mu_{\perp q}\right\rangle / \kappa, \text{ or } \beta_u \text{ and } \Delta\beta \equiv \beta_s - \beta_u, \tag{210}$$

then govern both the diquark and the intermediate meson production. In this algorithm, configurations like $BMM\overline{B}$ etc. are considered in a natural way. The more independent production of the diquark partons implies a moderate suppression of spin 1 diquarks. Instead the direct suppression of spin 3/2 baryons, in correspondence to the suppression of vector mesons relative to pseudo-scalar ones, is assumed to be important. Consequently, a suppression of Σ-states relative to Λ^0 is derived from the spin 3/2 suppression parameter.

Several new routines have been added, and the diquark code has been extended with information about the curtain quark flavour, i.e. the $q\overline{q}$ pair that is shared between the baryon and antibaryon, but this is not visible externally. Some parameters are no longer used, while others have to be given modified values. This is described in section 14.3.1.

Baryon remnant fragmentation

Occasionally, the endpoint of a string is not a single parton, but a diquark or antidiquark, e.g. when a quark has been kicked out of a proton beam particle. One could consider fairly complex schemes for the resulting fragmentation. One such [And81] was available in JETSET version 6 but is no longer found here. Instead the same basic scheme is used as for diquark pair production above. Thus a qq diquark endpoint is let to fragment just as would a qq produced in the field behind a matching $\overline{q}\overline{q}$ flavour, i.e. either the two quarks of the diquark enter into the same leading baryon, or else a meson is first produced, containing one of the quarks, while the other is contained in the baryon produced in the next step.

Similarly, the revised algorithm for baron production can be applied to endpoint diquarks, though this must be made with some care [Edé97]. The suppression factor for popcorn mesons is derived from the assumption of colour fluctuations violating energy conservation and thus being suppressed by the Heisenberg uncertainty principle. When splitting an original diquark into two more independent quarks, the same kind of energy shift does not obviously emerge. One could still expect large separations of the diquark constituents to be suppressed, but the shape of this suppression is beyond the scope of the model. For simplicity, the same kind of exponential suppression as in the "true popcorn" case is implemented in the program. However, there is little reason for the strength of the suppression to be exactly the same in the different situations. Thus the leading rank meson production in a diquark jet is governed by a new β parameter, which is independent of the popcorn parameters β_u and $\Delta\beta$ in eq. (210). Furthermore, in the process (original diquark $\rightarrow$ baryon+$\overline{q}$) the spin 3/2 suppression should not apply at full strength. This suppression factor stems from the normalization of the overlapping q and $\overline{q}$ wavefunctions in a newly produced $q\overline{q}$ pair, but in the process considered here, two out of three valence quarks already exist as an initial condition of the string.

12.2 String Fragmentation

An iterative procedure can also be used for other aspects of the fragmentation. This is possible because, in the string picture, the various points where the string break by the production of $q\overline{q}$ pairs are causally disconnected. Whereas the space–time picture in the c.m. frame is such that slow particles (in the middle of the system) are formed first, this ordering is Lorentz frame dependent and hence irrelevant. One may therefore make the convenient choice of starting an iteration process at the ends of the string and proceeding towards the middle.

The string fragmentation scheme is rather complicated for a generic multiparton state. In order to simplify the discussion, we will therefore start with the simple $q\overline{q}$ process, and only later survey the complications that appear when additional gluons are present. (This distinction is made for pedagogical reasons, in the program there is only one general-purpose algorithm).

12.2.1 Fragmentation functions

Assume a $q\overline{q}$ jet system, in its c.m. frame, with the quark moving out in the $+z$ direction and the antiquark in the $-z$ one. We have discussed how it is possible to start the flavour iteration from the q end, i.e. pick a $q_1\overline{q}_1$ pair, form a hadron $q\overline{q}_1$, etc. It has also been noted that the tunnelling mechanism is assumed to give a transverse momentum $p_\perp$ for each new $q_i\overline{q}_i$ pair created, with the $p_\perp$ locally compensated between the q_i and the $\overline{q}_i$ member of the pair, and with a Gaussian distribution in p_x and p_y separately. In the program, this is regulated by one parameter, which gives the root-mean-square $p_\perp$ of a quark. Hadron transverse momenta are obtained as the sum of $p_\perp$:s of the constituent q_i and $\overline{q}_{i+1}$, where a diquark is considered just as a single quark.

What remains to be determined is the energy and longitudinal momentum of the hadron. In fact, only one variable can be selected independently, since the momentum of the hadron is constrained by the already determined hadron transverse mass $m_\perp$,

$$(E+p_z)(E-p_z) = E^2 - p_z^2 = m_\perp^2 = m^2 + p_x^2 + p_y^2 \; . \tag{211}$$

In an iteration from the quark end, one is led (by the desire for longitudinal boost invariance and other considerations) to select the z variable as the fraction of $E+p_z$ taken by the hadron, out of the available $E+p_z$. As hadrons are split off, the $E+p_z$ (and $E-p_z$) left for subsequent steps is reduced accordingly:

$$\begin{aligned} (E+p_z)_{\rm new} &= (1-z)(E+p_z)_{\rm old} \; , \\ (E-p_z)_{\rm new} &= (E-p_z)_{\rm old} - \frac{m_\perp^2}{z(E+p_z)_{\rm old}} \; . \end{aligned} \tag{212}$$

The fragmentation function $f(z)$, which expresses the probability that a given z is picked, could in principle be arbitrary — indeed, several such choices can be used inside the program, see below.

If one, in addition, requires that the fragmentation process as a whole should look the same, irrespectively of whether the iterative procedure is performed from the q end or the $\overline{q}$ one, 'left–right symmetry', the choice is essentially unique [And83a]: the 'Lund symmetric fragmentation function',

$$f(z) \propto \frac{1}{z} z^{a_\alpha} \left(\frac{1-z}{z}\right)^{a_\beta} \exp\left(-\frac{bm_\perp^2}{z}\right) \; . \tag{213}$$

There is one separate parameter a for each flavour, with the index α corresponding to the 'old' flavour in the iteration process, and β to the 'new' flavour. It is customary to put all $a_{\alpha,\beta}$ the same, and thus arrive at the simplified expression

$$f(z) \propto z^{-1}(1-z)^a \exp(-bm_\perp^2/z) \; . \tag{214}$$

In the program, only two separate a values can be given, that for quark pair production and that for diquark one. In addition, there is the b parameter, which is universal.

The explicit mass dependence in $f(z)$ implies a harder fragmentation function for heavier hadrons. The asymptotic behaviour of the mean z value for heavy hadrons is

$$\langle z \rangle \approx 1 - \frac{1+a}{bm_\perp^2} \; . \tag{215}$$

Unfortunately it seems this predicts a somewhat harder spectrum for B mesons than observed in data. However, Bowler [Bow81] has shown, within the framework of the Artru–Mennessier model [Art74], that a massive endpoint quark with mass m_Q leads to a modification of the symmetric fragmentation function, due to the fact that the string area swept out is reduced for massive endpoint quarks, compared with massless ditto. The Artru–Mennessier model in

principle only applies for clusters with a continuous mass spectrum, and does not allow an a term (i.e. $a \equiv 0$); however, it has been shown [Mor89] that, for a discrete mass spectrum, one may still retain an effective a term. In the program an approximate form with an a term has therefore been used:

$$f(z) \propto \frac{1}{z^{1+r_Q b m_Q^2}} z^{a_\alpha} \left(\frac{1-z}{z}\right)^{a_\beta} \exp\left(-\frac{b m_\perp^2}{z}\right) . \tag{216}$$

In principle the prediction is that $r_Q \equiv 1$, but so as to be able to extrapolate smoothly between this form and the original Lund symmetric one, it is possible to pick r_Q separately for c and b hadrons.

For future reference we note that the derivation of $f(z)$ as a by-product also gives the probability distribution in proper time τ of $\mathrm{q}_i\overline{\mathrm{q}}_i$ breakup vertices. In terms of $\Gamma = (\kappa\tau)^2$, this distribution is

$$\mathcal{P}(\Gamma)\,\mathrm{d}\Gamma \propto \Gamma^a \exp(-b\Gamma)\,\mathrm{d}\Gamma , \tag{217}$$

with the same a and b as above. The exponential decay allows an interpretation in terms of an area law for the colour flux [And98].

Many different other fragmentation functions have been proposed, and a few are available as options in the program.

- The Field-Feynman parameterization [Fie78],

$$f(z) = 1 - a + 3a(1-z)^2 , \tag{218}$$

  with default value $a = 0.77$, is intended to be used only for ordinary hadrons made out of u, d and s quarks.
- Since there are indications that the shape above is too strongly peaked at $z = 0$, instead a shape like

$$f(z) = (1+c)(1-z)^c \tag{219}$$

  may be used.
- Charm and bottom data clearly indicate the need for a harder fragmentation function for heavy flavours. The best known of these is the Peterson/SLAC formula [Pet83]

$$f(z) \propto \frac{1}{z\left(1 - \frac{1}{z} - \frac{\varepsilon_Q}{1-z}\right)^2} , \tag{220}$$

  where ε_Q is a free parameter, expected to scale between flavours like $\varepsilon_Q \propto 1/m_Q^2$.
- As a crude alternative, that is also peaked at $z = 1$, one may use

$$f(z) = (1+c)z^c . \tag{221}$$

- In [Edé97], it is argued that the quarks responsible for the colour fluctuations in stepwise diquark production cannot move along the light-cones. Instead there is an area of possible starting points for the colour fluctuation, which is essentially given by the proper time of the vertex squared. By summing over all possible starting points, one obtains the total weight for the colour fluctuation. The result is a relative suppression of diquark vertices at early times, which is found to be of the form $1 - \exp(-\rho\Gamma)$, where $\Gamma \equiv \kappa^2\tau^2$ and $\rho \approx 0.7\mathrm{GeV}^{-2}$. This result, and especially the value of ρ, is independent of the fragmentation function, $f(z)$, used to reach a specific Γ-value. However, if using a $f(z)$ which implies a small average value $\langle\Gamma\rangle$, the program implementation is such that a large fraction of the $\mathrm{q} \to B + \mathrm{qq}$ attempts will be rejected. This dilutes the interpretation of the input $P(\mathrm{qq})/P(\mathrm{q})$ parameter, which needs to be significantly enhanced to compensate for

the rejections.

A property of the Lund Symmetric Fragmentation Function is that the first vertices produced near the string ends have a lower $\langle\Gamma\rangle$ than central vertices. Thus an effect of the low-Γ suppression is a relative reduction of the leading baryons. The effect is smaller if the baryon is very heavy, as the large mass implies that the first vertex almost reaches the central region. Thus the leading baryon suppression effect is reduced for c- and b jets.

12.2.2 Joining the jets

The $f(z)$ formula above is only valid, for the breakup of a jet system into a hadron plus a remainder-system, when the remainder mass is large. If the fragmentation algorithm were to be used all the way from the q end to the $\overline{\mathrm{q}}$ one, the mass of the last hadron to be formed at the $\overline{\mathrm{q}}$ end would be completely constrained by energy and momentum conservation, and could not be on its mass shell. In theory it is known how to take such effects into account [Edé00], but the resulting formulae are wholly unsuitable for Monte Carlo implementation.

The practical solution to this problem is to carry out the fragmentation both from the q and the $\overline{\mathrm{q}}$ end, such that for each new step in the fragmentation process, a random choice is made as to from what side the step is to be taken. If the step is on the q side, then z is interpreted as fraction of the remaining $E+p_z$ of the system, while z is interpreted as $E-p_z$ fraction for a step from the $\overline{\mathrm{q}}$ end. At some point, when the remaining mass of the system has dropped below a given value, it is decided that the next breakup will produce two final hadrons, rather than a hadron and a remainder-system. Since the momenta of two hadrons are to be selected, rather than that of one only, there are enough degrees of freedom to have both total energy and total momentum completely conserved.

The mass at which the normal fragmentation process is stopped and the final two hadrons formed is not actually a free parameter of the model: it is given by the requirement that the string everywhere looks the same, i.e. that the rapidity spacing of the final two hadrons, internally and with respect to surrounding hadrons, is the same as elsewhere in the fragmentation process. The stopping mass, for a given setup of fragmentation parameters, has therefore been determined in separate runs. If the fragmentation parameters are changed, some retuning should be done but, in practice, reasonable changes can be made without any special arrangements.

Consider a fragmentation process which has already split off a number of hadrons from the q and $\overline{\mathrm{q}}$ sides, leaving behind a a $\mathrm{q}_i\overline{\mathrm{q}}_j$ remainder system. When this system breaks by the production of a $\mathrm{q}_n\overline{\mathrm{q}}_n$ pair, it is decided to make this pair the final one, and produce the last two hadrons $\mathrm{q}_i\overline{\mathrm{q}}_n$ and $\mathrm{q}_n\overline{\mathrm{q}}_j$, if

$$((E+p_z)(E-p_z))_{\mathrm{remaining}} = W_{\mathrm{rem}}^2 < W_{\mathrm{min}}^2 \; . \tag{222}$$

The W_{min} is calculated according to

$$W_{\mathrm{min}} = \left(W_{\mathrm{min0}} + m_{\mathrm{q}i} + m_{\mathrm{q}j} + k\, m_{\mathrm{q}n}\right)(1 \pm \delta) \; . \tag{223}$$

Here W_{min0} is the main free parameter, typically around 1 GeV, determined to give a flat rapidity plateau (separately for each particle species), while the default $k=2$ corresponds to the mass of the final pair being taken fully into account. Smaller values may also be considered, depending on what criteria are used to define the 'best' joining of the q and the $\overline{\mathrm{q}}$ chain. The factor $1\pm\delta$, by default evenly distributed between 0.8 and 1.2, signifies a smearing of the W_{min} value, to avoid an abrupt and unphysical cut-off in the invariant mass distribution of the final two hadrons. Still, this distribution will be somewhat different from that of any two adjacent hadrons elsewhere. Due to the cut there will be no tail up to very high masses; there are also fewer events close to the lower limit, where the two hadrons are formed at rest with respect to each other.

This procedure does not work all that well for heavy flavours, since it does not fully take into account the harder fragmentation function encountered. Therefore, in addition to the check above, one further test is performed for charm and heavier flavours, as follows. If the check

above allows more particle production, a heavy hadron $q_i\overline{q}_n$ is formed, leaving a remainder $q_n\overline{q}_j$. The range of allowed z values, i.e. the fraction of remaining $E+p_z$ that may be taken by the $q_i\overline{q}_n$ hadron, is constrained away from 0 and 1 by the $q_i\overline{q}_n$ mass and minimal mass of the $q_n\overline{q}_j$ system. The limits of the physical z range is obtained when the $q_n\overline{q}_j$ system only consists of one single particle, which then has a well-determined transverse mass $m_\perp^{(0)}$. From the z value obtained with the infinite-energy fragmentation function formulae, a rescaled z' value between these limits is given by

$$z' = \frac{1}{2}\left\{1+\frac{m_{\perp in}^2}{W_{\rm rem}^2}-\frac{m_{\perp nj}^{(0)2}}{W_{\rm rem}^2}+\sqrt{\left(1-\frac{m_{\perp in}^2}{W_{\rm rem}^2}-\frac{m_{\perp nj}^{(0)2}}{W_{\rm rem}^2}\right)^2-4\frac{m_{\perp in}^2}{W_{\rm rem}^2}\frac{m_{\perp nj}^{(0)2}}{W_{\rm rem}^2}(2z-1)}\right\}. \qquad (224)$$

From the z' value, the actual transverse mass $m_{\perp nj} \geq m_{\perp nj}^{(0)}$ of the $q_n\overline{q}_j$ system may be calculated. For more than one particle to be produced out of this system, the requirement

$$m_{\perp nj}^2 = (1-z')\left(W_{\rm rem}^2-\frac{m_{\perp in}^2}{z'}\right) > (m_{qj}+W_{\rm min0})^2+p_\perp^2 \qquad (225)$$

has to be fulfilled. If not, the $q_n\overline{q}_j$ system is assumed to collapse to one single particle.

The consequence of the procedure above is that, the more the infinite energy fragmentation function $f(z)$ is peaked close to $z=1$, the more likely it is that only two particles are produced. The procedure above has been constructed so that the two-particle fraction can be calculated directly from the shape of $f(z)$ and the (approximate) mass spectrum, but it is not unique. For the symmetric Lund fragmentation function, a number of alternatives tried all give essentially the same result, whereas other fragmentation functions may be more sensitive to details.

Assume now that two final hadrons have been picked. If the transverse mass of the remainder-system is smaller than the sum of transverse masses of the final two hadrons, the whole fragmentation chain is rejected, and started over from the q and $\overline{q}$ endpoints. This does not introduce any significant bias, since the decision to reject a fragmentation chain only depends on what happens in the very last step, specifically that the next-to-last step took away too much energy, and not on what happened in the steps before that.

If, on the other hand, the remainder-mass is large enough, there are two kinematically allowed solutions for the final two hadrons: the two mirror images in the rest frame of the remainder-system. Also the choice between these two solutions is given by the consistency requirements, and can be derived from studies of infinite energy jets. The probability for the reverse ordering, i.e. where the rapidity and the flavour orderings disagree, is given by the area law as

$$\mathcal{P}_{\rm reverse} = \frac{1}{1+e^{b\Delta}} \quad \text{where} \quad \Delta = \Gamma_2-\Gamma_1 = \sqrt{(W_{\rm rem}^2-m_{\perp in}^2+m_{\perp nj}^2)^2-4m_{\perp in}^2 m_{\perp nj}^2}\,. \qquad (226)$$

For the Lund symmetric fragmentation function, b is the familiar parameter, whereas for other functions the b value becomes an effective number to be fitted to the behaviour when not in the joining region.

When baryon production is included, some particular problems arise (also see section 12.1.3). First consider $B\overline{B}$ situations. In the naïve iterative scheme, away from the middle of the event, one already has a quark and is to chose a matching diquark flavour or the other way around. In either case the choice of the new flavour can be done taking into account the number of **SU(6)** states available for the quark-diquark combination. For a case where the final $q_n\overline{q}_n$ breakup is an antidiquark-diquark one, the weights for forming $q_i\overline{q}_n$ and $q_n\overline{q}_i$ enter at the same time, however. We do not know how to handle this problem; what is done is to use weights

as usual for the $q_i\overline{q}_n$ baryon to select q_n, but then consider $q_n\overline{q}_i$ as given (or the other way around with equal probability). If $q_n\overline{q}_i$ turns out to be an antidiquark-diquark combination, the whole fragmentation chain is rejected, since we do not know how to form corresponding hadrons. A similar problem arises, and is solved in the same spirit, for a $BM\overline{B}$ configuration in which the B (or $\overline{B}$) was chosen as third-last particle. When only two particles remain to be generated, it is obviously too late to consider having a $BM\overline{B}$ configuration. This is as it should, however, as can be found by looking at all possible ways a hadron of given rank can be a baryon.

While some practical compromises have to be accepted in the joining procedure, the fact that the joining takes place in different parts of the string in different events means that, in the end, essentially no visible effects remain.

12.2.3 String motion and infrared stability

We have now discussed the SF scheme for the fragmentation of a simple $q\overline{q}$ jet system. In order to understand how these results generalize to arbitrary jet systems, it is first necessary to understand the string motion for the case when no fragmentation takes place. In the following we will assume that quarks as well as gluons are massless, but all arguments can be generalized to massive quarks without too much problem.

For a $q\overline{q}$ event viewed in the c.m. frame, with total energy W, the partons start moving out back-to-back, carrying half the energy each. As they move apart, energy and momentum is lost to the string. When the partons are a distance W/κ apart, all the energy is stored in the string. The partons now turn around and come together again with the original momentum vectors reversed. This corresponds to half a period of the full string motion; the second half the process is repeated, mirror-imaged. For further generalizations to multiparton systems, a convenient description of the energy and momentum flow is given in terms of 'genes' [Art83], infinitesimal packets of the four-momentum given up by the partons to the string. Genes with $p_z = E$, emitted from the q end in the initial stages of the string motion above, will move in the $\overline{q}$ direction with the speed of light, whereas genes with $p_z = -E$ given up by the $\overline{q}$ will move in the q direction. Thus, in this simple case, the direction of motion for a gene is just opposite to that of a free particle with the same four-momentum. This is due to the string tension. If the system is not viewed in the c.m. frame, the rules are that any parton gives up genes with four-momentum proportional to its own four-momentum, but the direction of motion of any gene is given by the momentum direction of the genes it meets, i.e. that were emitted by the parton at the other end of that particular string piece. When the q has lost all its energy, the $\overline{q}$ genes, which before could not catch up with q, start impinging on it, and the q is pulled back, accreting $\overline{q}$ genes in the process. When the q and $\overline{q}$ meet in the origin again, they have completely traded genes with respect to the initial situation.

A 3-jet $q\overline{q}g$ event initially corresponds to having a string piece stretched between q and g and another between g and $\overline{q}$. Gluon four-momentum genes are thus flowing towards the q and $\overline{q}$. Correspondingly, q and $\overline{q}$ genes are flowing towards the g. When the gluon has lost all its energy, the g genes continue moving apart, and instead a third string region is formed in the 'middle' of the total string, consisting of overlapping q and $\overline{q}$ genes. The two 'corners' on the string, separating the three string regions, are not of the gluon-kink type: they do not carry any momentum.

If this third region would only appear at a time later than the typical time scale for fragmentation, it could not affect the sharing of energy between different particles. This is true in the limit of high energy, well separated partons. For a small gluon energy, on the other hand, the third string region appears early, and the overall drawing of the string becomes fairly 2-jet-like, since the third string region consists of q and $\overline{q}$ genes and therefore behaves exactly as a sting pulled out directly between the q and $\overline{q}$. In the limit of vanishing gluon energy, the two initial string regions collapse to naught, and the ordinary 2-jet event is recovered [Sjö84]. Also for a collinear gluon, i.e. θ_{qg} (or $\theta_{\overline{q}g}$) small, the stretching becomes 2-jet-like. In particular, the q string endpoint first moves out a distance $\mathbf{p}_q/\kappa$ losing genes to the string, and then a further

distance $\mathbf{p}_g/\kappa$, a first half accreting genes from the g and the second half re-emitting them. (This latter half actually includes yet another string piece; a corresponding piece appears at the $\overline{q}$ end, such that half a period of the system involves five different string regions.) The end result is, approximately, that a string is drawn out as if there had only been a single parton with energy $|\mathbf{p}_q + \mathbf{p}_g|$, such that the simple 2-jet event again is recovered in the limit $\theta_{qg} \to 0$. These properties of the string motion are the reason why the string fragmentation scheme is 'infrared safe' with respect to soft or collinear gluon emission.

The discussions for the 3-jet case can be generalized to the motion of a string with q and $\overline{q}$ endpoints and an arbitrary number of intermediate gluons. For n partons, whereof $n-2$ gluons, the original string contains $n-1$ pieces. Anytime one of the original gluons has lost its energy, a new string region is formed, delineated by a pair of 'corners'. As the extra 'corners' meet each other, old string regions vanish and new are created, so that half a period of the string contains $2n^2 - 6n + 5$ different string regions. Each of these regions can be understood simply as built up from the overlap of (opposite-moving) genes from two of the original partons, according to well specified rules.

12.2.4 Fragmentation of multiparton systems

The full machinery needed for a multiparton system is very complicated, and is described in detail in [Sjö84]. The following outline is far from complete, and is complicated nonetheless. The main message to be conveyed is that a Lorentz covariant algorithm exists for handling an arbitrary parton configuration, but that the necessary machinery is more complex than in either cluster or independent fragmentation.

Assume n partons, with ordering along the string, and related four-momenta, given by $q(p_1)g(p_2)g(p_3)\cdots g(p_{n-1})\overline{q}(p_n)$. The initial string then contains $n-1$ separate pieces. The string piece between the quark and its neighbouring gluon is, in four-momentum space, spanned by one side with four-momentum $p_+^{(1)} = p_1$ and another with $p_-^{(1)} = p_2/2$. The factor of 1/2 in the second expression comes from the fact that the gluon shares its energy between two string pieces. The indices '$+$' and '$-$' denotes direction towards the q and $\overline{q}$ end, respectively. The next string piece, counted from the quark end, is spanned by $p_+^{(2)} = p_2/2$ and $p_-^{(2)} = p_3/2$, and so on, with the last one being $p_+^{(n-1)} = p_{n-1}/2$ and $p_-^{(n-1)} = p_n$.

For the algorithm to work, it is important that all $p_\pm^{(i)}$ be light-cone-like, i.e. $p_\pm^{(i)2} = 0$. Since gluons are massless, it is only the two endpoint quarks which can cause problems. The procedure here is to create new $p_\pm$ vectors for each of the two endpoint regions, defined to be linear combinations of the old $p_\pm$ ones for the same region, with coefficients determined so that the new vectors are light-cone-like. De facto, this corresponds to replacing a massive quark at the end of a string piece with a massless quark at the end of a somewhat longer string piece. With the exception of the added fictitious piece, which anyway ends up entirely within the heavy hadron produced from the heavy quark, the string motion remains unchanged by this.

In the continued string motion, when new string regions appear as time goes by, the first such string regions that appear can be represented as being spanned by one $p_+^{(j)}$ and another $p_-^{(k)}$ four-vector, with j and k not necessarily adjacent. For instance, in the $qg\overline{q}$ case, the 'third' string region is spanned by $p_+^{(1)}$ and $p_-^{(3)}$. Later on in the string evolution history, it is also possible to have regions made up of two p_+ or two p_- momenta. These appear when an endpoint quark has lost all its original momentum, has accreted the momentum of an gluon, and is now re-emitting this momentum. In practice, these regions may be neglected. Therefore only pieces made up by a $(p_+^{(j)}, p_-^{(k)})$ pair of momenta are considered in the program.

The allowed string regions may be ordered in an abstract parameter plane, where the (j,k) indices of the four-momentum pairs define the position of each region along the two (parameter plane) coordinate axes. In this plane the fragmentation procedure can be described as a sequence of steps, starting at the quark end, where $(j,k) = (1,1)$, and ending at the antiquark one, $(j,k) =$

$(n-1, n-1)$. Each step is taken from an 'old' $q_{i-1}\overline{q}_{i-1}$ pair production vertex, to the production vertex of a 'new' $q_i\overline{q}_i$ pair, and the string piece between these two string breaks represent a hadron. Some steps may be taken within one and the same region, while others may have one vertex in one region and the other vertex in another region. Consistency requirements, like energy-momentum conservation, dictates that vertex j and k region values be ordered in a monotonic sequence, and that the vertex positions are monotonically ordered inside each region. The four-momentum of each hadron can be read off, for p_+ (p_-) momenta, by projecting the separation between the old and the new vertex on to the j (k) axis. If the four-momentum fraction of $p_\pm^{(i)}$ taken by a hadron is denoted $x_\pm^{(i)}$, then the total hadron four-momentum is given by

$$p = \sum_{j=j_1}^{j_2} x_+^{(j)} p_+^{(j)} + \sum_{k=k_1}^{k_2} x_-^{(k)} p_-^{(k)} + p_{x1}\hat{e}_x^{(j_1k_1)} + p_{y1}\hat{e}_y^{(j_1k_1)} + p_{x2}\hat{e}_x^{(j_2k_2)} + p_{y2}\hat{e}_y^{(j_2k_2)} \ , \qquad (227)$$

for a step from region (j_1, k_1) to region (j_2, k_2). By necessity, $x_+^{(j)}$ is unity for a $j_1 < j < j_2$, and correspondingly for $x_-^{(k)}$.

The (p_x, p_y) pairs are the transverse momenta produced at the two string breaks, and the $(\hat{e}_x, \hat{e}_y)$ pairs four-vectors transverse to the string directions in the regions of the respective string breaks:

$$\begin{aligned} &\hat{e}_x^{(jk)2} = \hat{e}_y^{(jk)2} = -1 \ , \\ &\hat{e}_x^{(jk)}\hat{e}_y^{(jk)} = \hat{e}_{x,y}^{(jk)} p_+^{(j)} = \hat{e}_{x,y}^{(jk)} p_-^{(k)} = 0 \ . \end{aligned} \qquad (228)$$

The fact that the hadron should be on mass shell, $p^2 = m^2$, puts one constraint on where a new breakup may be, given that the old one is already known, just as eq. (211) did in the simple 2-jet case. The remaining degree of freedom is, as before, to be given by the fragmentation function $f(z)$. The interpretation of the z is only well-defined for a step entirely constrained to one of the initial string regions, however, which is not enough. In the 2-jet case, the z values can be related to the proper times of string breaks, as follows. The variable $\Gamma = (\kappa\tau)^2$, with κ the string tension and τ the proper time between the production vertex of the partons and the breakup point, obeys an iterative relation of the kind

$$\begin{aligned} \Gamma_0 &= 0 \ , \\ \Gamma_i &= (1-z_i)\left(\Gamma_{i-1} + \frac{m_{\perp i}^2}{z_i}\right) \ . \end{aligned} \qquad (229)$$

Here Γ_0 represents the value at the q and $\overline{q}$ endpoints, and Γ_{i-1} and Γ_i the values at the old and new breakup vertices needed to produce a hadron with transverse mass $m_{\perp i}$, and with the z_i of the step chosen according to $f(z_i)$. The proper time can be defined in an unambiguous way, also over boundaries between the different string regions, so for multijet events the z variable may be interpreted just as an auxiliary variable needed to determine the next Γ value. (In the Lund symmetric fragmentation function derivation, the Γ variable actually does appear naturally, so the choice is not as arbitrary as it may seem here.) The mass and Γ constraints together are sufficient to determine where the next string breakup is to be chosen, given the preceding one in the iteration scheme. Actually, several ambiguities remain, but are of no importance for the overall picture.

The algorithm for finding the next breakup then works something like follows. Pick a hadron, $p_\perp$, and z, and calculate the next Γ. If the old breakup is in the region (j, k), and if the new breakup is also assumed to be in the same region, then the m^2 and Γ constraints can be

reformulated in terms of the fractions $x_+^{(j)}$ and $x_-^{(k)}$ the hadron must take of the total four-vectors $p_+^{(j)}$ and $p_-^{(k)}$:

$$\begin{aligned} m^2 &= c_1 + c_2 x_+^{(j)} + c_3 x_-^{(k)} + c_4 x_+^{(j)} x_-^{(k)} \ , \\ \Gamma &= d_1 + d_2 x_+^{(j)} + d_3 x_-^{(k)} + d_4 x_+^{(j)} x_-^{(k)} \ . \end{aligned} \tag{230}$$

Here the coefficients c_n are fairly simple expressions, obtainable by squaring eq. (227), while d_n are slightly more complicated in that they depend on the position of the old string break, but both the c_n and the d_n are explicitly calculable. What remains is an equation system with two unknowns, $x_+^{(j)}$ and $x_-^{(k)}$. The absence of any quadratic terms is due to the fact that all $p_\pm^{(i)2} = 0$, i.e. to the choice of a formulation based on light-cone-like longitudinal vectors. Of the two possible solutions to the equation system (elimination of one variable gives a second degree equation in the other), one is unphysical and can be discarded outright. The other solution is checked for whether the $x_\pm$ values are actually inside the physically allowed region, i.e. whether the $x_\pm$ values of the current step, plus whatever has already been used up in previous steps, are less than unity. If yes, a solution has been found. If no, it is because the breakup could not take place inside the region studied, i.e. because the equation system was solved for the wrong region. One therefore has to change either index j or index k above by one step, i.e. go to the next nearest string region. In this new region, a new equation system of the type in eq. (230) may be written down, with new coefficients. A new solution is found and tested, and so on until a physically acceptable solution is found. The hadron four-momentum is now given by an expression of the type (227). The breakup found forms the starting point for the new step in the fragmentation chain, and so on. The final joining in the middle is done as in the 2-jet case, with minor extensions.

12.3 Independent Fragmentation

The independent fragmentation (IF) approach dates back to the early seventies [Krz72], and gained widespread popularity with the Field-Feynman paper [Fie78]. Subsequently, IF was the basis for two programs widely used in the early PETRA/PEP days, the Hoyer et al. [Hoy79] and the Ali et al. [Ali80] programs. PYTHIA has as (non-default) options a wide selection of independent fragmentation algorithms.

12.3.1 Fragmentation of a single jet

In the IF approach, it is assumed that the fragmentation of any system of partons can be described as an incoherent sum of independent fragmentation procedures for each parton separately. The process is to be carried out in the overall c.m. frame of the jet system, with each jet fragmentation axis given by the direction of motion of the corresponding parton in that frame.

Exactly as in string fragmentation, an iterative ansatz can be used to describe the successive production of one hadron after the next. Assume that a quark is kicked out by some hard interaction, carrying a well-defined amount of energy and momentum. This quark jet q is split into a hadron $\mathrm{q}\overline{\mathrm{q}}_1$ and a remainder-jet q_1, essentially collinear with each other. New quark and hadron flavours are picked as already described. The sharing of energy and momentum is given by some probability distribution $f(z)$, where z is the fraction taken by the hadron, leaving $1-z$ for the remainder-jet. The remainder-jet is assumed to be just a scaled-down version of the original jet, in an average sense. The process of splitting off a hadron can therefore be iterated, to yield a sequence of hadrons. In particular, the function $f(z)$ is assumed to be the same at each step, i.e. independent of remaining energy. If z is interpreted as the fraction of the jet $E + p_\mathrm{L}$, i.e. energy plus longitudinal momentum with respect to the jet axis, this leads to a flat central rapidity plateau dn/dy for a large initial energy.

Fragmentation functions can be chosen among those listed above for string fragmentation, but also here the default is the Lund symmetric fragmentation function.

The normal z interpretation means that a choice of a z value close to 0 corresponds to a particle moving backwards, i.e. with $p_{\mathrm{L}} < 0$. It makes sense to allow only the production of particles with $p_{\mathrm{L}} > 0$, but to explicitly constrain z accordingly would destroy longitudinal invariance. The most straightforward way out is to allow all z values but discard hadrons with $p_{\mathrm{L}} < 0$. Flavour, transverse momentum and $E + p_{\mathrm{L}}$ carried by these hadrons are 'lost' for the forward jet. The average energy of the final jet comes out roughly right this way, with a spread of 1–2 GeV around the mean. The jet longitudinal momentum is decreased, however, since the jet acquires an effective mass during the fragmentation procedure. For a 2-jet event this is as it should be, at least on average, because also the momentum of the compensating opposite-side parton is decreased.

Flavour is conserved locally in each $\mathrm{q}_i\overline{\mathrm{q}}_i$ splitting, but not in the jet as a whole. First of all, there is going to be a last meson $\mathrm{q}_{n-1}\overline{\mathrm{q}}_n$ generated in the jet, and that will leave behind an unpaired quark flavour q_n. Independent fragmetation does not specify the fate of this quark. Secondly, also a meson in the middle of the flavour chain may be selected with such a small z value that it obtains $p_{\mathrm{L}} < 0$ and is rejected. Thus a u quark jet of charge $2/3$ need not only gives jets of charge 0 or 1, but also of -1 or $+2$, or even higher. Like with the jet longitudinal momentum above, one could imagine this compensated by other jets in the event.

It is also assumed that transverse momentum is locally conserved, i.e. the net $p_\perp$ of the $\mathrm{q}_i\overline{\mathrm{q}}_i$ pair as a whole is assumed to be vanishing. The $p_\perp$ of the q_i is taken to be a Gaussian in the two transverse degrees of freedom separately, with the transverse momentum of a hadron obtained by the sum of constituent quark transverse momenta. The total $p_\perp$ of a jet can fluctuate for the same two reasons as discussed above for flavour. Furthermore, in some scenarios one may wish to have the same $p_\perp$ distribution for the first-rank hadron $\mathrm{q}\overline{\mathrm{q}}_1$ as for subsequent ones, in which case also the original q should be assigned an unpaired $p_\perp$ according to a Gaussian.

Within the IF framework, there is no unique recipe for how gluon jet fragmentation should be handled. One possibility is to treat it exactly like a quark jet, with the initial quark flavour chosen at random among u, $\overline{\mathrm{u}}$, d, $\overline{\mathrm{d}}$, s and $\overline{\mathrm{s}}$, including the ordinary s quark suppression factor. Since the gluon is supposed to fragment more softly than a quark jet, the fragmentation function may be chosen independently. Another common option is to split the g jet into a pair of parallel q and $\overline{\mathrm{q}}$ ones, sharing the energy, e.g. as in a perturbative branching $\mathrm{g} \to \mathrm{q}\overline{\mathrm{q}}$, i.e. $f(z) \propto z^2 + (1-z)^2$. The fragmentation function could still be chosen independently, if so desired. Further, in either case the fragmentation $p_\perp$ could be chosen to have a different mean.

12.3.2 Fragmentation of a jet system

In a system of many jets, each jet is fragmented independently. Since each jet by itself does not conserve the flavour, energy and momentum, as we have seen, then neither does a system of jets. At the end of the generation, special algorithms are therefore used to patch this up. The choice of approach has major consequences, e.g. for event shapes and α_{s} determinations [Sjö84a].

Little attention is usually given to flavour conservation, and we only offer one scheme. When the fragmentation of all jets has been performed, independently of each other, the net initial flavour composition, i.e. number of u quarks minus number of $\overline{\mathrm{u}}$ quarks etc., is compared with the net final flavour composition. In case of an imbalance, the flavours of the hadron with lowest three-momentum are removed, and the imbalance is re-evaluated. If the remaining imbalance could be compensated by a suitable choice of new flavours for this hadron, flavours are so chosen, a new mass is found and the new energy can be evaluated, keeping the three-momentum of the original hadron. If the removal of flavours from the hadron with lowest momentum is not enough, flavours are removed from the one with next-lowest momentum, and so on until enough freedom is obtained, whereafter the necessary flavours are recombined at random to form the new hadrons. Occasionally one extra $\mathrm{q}_i\overline{\mathrm{q}}_i$ pair must be created, which is

then done according to the customary probabilities.

Several different schemes for energy and momentum conservation have been devised. One [Hoy79] is to conserve transverse momentum locally within each jet, so that the final momentum vector of a jet is always parallel with that of the corresponding parton. Then longitudinal momenta may be rescaled separately for particles within each jet, such that the ratio of rescaled jet momentum to initial parton momentum is the same in all jets. Since the initial partons had net vanishing three-momentum, so do now the hadrons. The rescaling factors may be chosen such that also energy comes out right. Another common approach [Ali80] is to boost the event to the frame where the total hadronic momentum is vanishing. After that, energy conservation can be obtained by rescaling all particle three-momenta by a common factor.

The number of possible schemes is infinite. Two further options are available in the program. One is to shift all particle three-momenta by a common amount to give net vanishing momentum, and then rescale as before. Another is to shift all particle three-momenta, for each particle by an amount proportional to the longitudinal mass with respect to the imbalance direction, and with overall magnitude selected to give momentum conservation, and then rescale as before. In addition, there is a choice of whether to treat separate colour singlets (like $q\overline{q}'$ and $q'\overline{q}$ in a $q\overline{q}q'\overline{q}'$ event) separately or as one single big system.

A serious conceptual weakness of the IF framework is the issue of Lorentz invariance. The outcome of the fragmentation procedure depends on the coordinate frame chosen, a problem circumvented by requiring fragmentation always to be carried out in the c.m. frame. This is a consistent procedure for 2-jet events, but only a technical trick for multijets.

It should be noted, however, that a Lorentz covariant generalization of the independent fragmentation model exists, in which separate 'gluon-type' and 'quark-type' strings are used, the Montvay scheme [Mon79]. The 'quark string' is characterized by the ordinary string constant κ, whereas a 'gluon string' is taken to have a string constant κ_g. If $\kappa_g > 2\kappa$ it is always energetically favourable to split a gluon string into two quark ones, and the ordinary Lund string model is recovered. Otherwise, for a 3-jet $q\overline{q}g$ event the three different string pieces are joined at a junction. The motion of this junction is given by the composant of string tensions acting on it. In particular, it is always possible to boost an event to a frame where this junction is at rest. In this frame, much of the standard naïve IF picture holds for the fragmentation of the three jets; additionally, a correct treatment would automatically give flavour, momentum and energy conservation. Unfortunately, the simplicity is lost when studying events with several gluon jets. In general, each event will contain a number of different junctions, resulting in a polypod shape with a number of quark and gluons strings sticking out from a skeleton of gluon strings. With the shift of emphasis from three-parton to multi-parton configurations, the simple option existing in JETSET 6.3 therefore is no longer included.

A second conceptual weakness of IF is the issue of collinear divergences. In a parton-shower picture, where a quark or gluon is expected to branch into several reasonably collimated partons, the independent fragmentation of one single parton or of a bunch of collinear ones gives quite different outcomes, e.g. with a much larger hadron multiplicity in the latter case. It is conceivable that a different set of fragmentation functions could be constructed in the shower case in order to circumvent this problem (local parton–hadron duality [Dok89] would correspond to having $f(z) = \delta(z-1)$).

12.4 Other Fragmentation Aspects

Here some further aspects are considered.

12.4.1 Small mass systems

A hadronic event is conventionally subdivided into sets of partons that form separate colour singlets. These sets are represented by strings, that e.g. stretch from a quark end via a number

of intermediate gluons to an antiquark end. Three string mass regions may be distinguished for the hadronization.

1. *Normal string fragmentation*. In the ideal situation, each string has a large invariant mass. Then the standard iterative fragmentation scheme above works well. In practice, this approach can be used for all strings above some cut-off mass of a few GeV.
2. *Cluster decay*. If a string is produced with a small invariant mass, maybe only two-body final states are kinematically accessible. The traditional iterative Lund scheme is then not applicable. We call such a low-mass string a cluster, and consider it separately from above. The modelling is still intended to give a smooth match on to the standard string scheme in the high-cluster-mass limit [Nor98].
3. *Cluster collapse*. This is the extreme case of the above situation, where the string mass is so small that the cluster cannot decay into two hadrons. It is then assumed to collapse directly into a single hadron, which inherits the flavour content of the string endpoints. The original continuum of string/cluster masses is replaced by a discrete set of hadron masses. Energy and momentum then cannot be conserved inside the cluster, but must be exchanged with the rest of the event [Nor98].

String systems below a threshold mass are handled by the cluster machinery. In it, an attempt is first made to produce two hadrons, by having the string break in the middle by the production of a new $q\overline{q}$ pair, with flavours and hadron spins selected according to the normal string rules. If the sum of the hadron masses is larger than the cluster mass, repeated attempts can be made to find allowed hadrons; the default is two tries. If an allowed set is found, the angular distribution of the decay products in the cluster rest framed is picked isotropically near the threshold, but then gradually more elongated along the string direction, to provide a smooth match to the string description at larger masses. This also includes a forward–backward asymmetry, so that each hadron is preferentially in the same hemisphere as the respective original quark it inherits.

If the attempts to find two hadrons fail, one single hadron is formed from the given flavour content. The basic strategy thereafter is to exchange some minimal amount of energy and momentum between the collapsing cluster and other string pieces in the neighbourhood. The momentum transfer can be in either direction, depending on whether the hadron is lighter or heavier than the cluster it comes from. When lighter, the excess momentum is split off and put as an extra 'gluon' on the nearest string piece, where 'nearest' is defined by a space–time history-based distance measure. When the hadron is heavier, momentum is instead borrowed from the endpoints of the nearest string piece.

The free parameters of the model can be tuned to data, especially to the significant asymmetries observed between the production of D and $\overline{D}$ mesons in $\pi^- p$ collisions, with hadrons that share some of the π^- flavour content very much favoured at large x_F in the π^- fragmentation region [Ada93]. These spectra and asymmetries are closely related to the cluster collapse mechanism, and also to other effects of the colour topology of the event ('beam drag') [Nor98]. The most direct parameters are the choice of compensation scheme (`MSTJ(16)`), the number of attempts to find a kinematically valid two-body decay (`MSTJ(16)`) and the border between cluster and string descriptions (`PARJ(32)`). Also many other parameters enter the description, however, such as the effective charm mass (`PMAS(4,1)`), the quark constituent masses (`PARF(101) - PARF(105)`), the beam remnant structure (`MSTP(91) - MSTP(94)` and `PARP(91) - PARP(100)`) and the standard string fragmentation parameters.

The cluster collapse is supposed to be a part of multiparticle production. It is not intended for exclusive production channels, and may there give quite misleading results. For instance, a $c\overline{c}$ quark pair produced in a $\gamma\gamma$ collision could well be collapsed to a single J/ψ if the invariant mass is small enough, even though the process $\gamma\gamma \to J/\psi$ in theory is forbidden by spin–parity–charge considerations. Furthermore, properties such as strong isospin are not considered in the string fragmentation picture (only its third component, i.e. flavour conservation), neither when one nor when many particles are produced. For multiparticle states this should matter little, since the isospin then will be duly randomized, but properly it would forbid the production of several one- or two-body states that currently are generated.

12.4.2 Interconnection Effects

The widths of the W, Z and t are all of the order of 2 GeV. A standard model Higgs with a mass above 200 GeV, as well as many supersymmetric and other beyond the standard model particles would also have widths in the multi-GeV range. Not far from threshold, the typical decay times $\tau = 1/\Gamma \approx 0.1\,\mathrm{fm} \ll \tau_{\mathrm{had}} \approx 1\,\mathrm{fm}$. Thus hadronic decay systems overlap, between a resonance and the underlying event, or between pairs of resonances, so that the final state may not contain independent resonance decays.

So far, studies have mainly been performed in the context of W pair production at LEP2. Pragmatically, one may here distinguish three main eras for such interconnection:

1. Perturbative: this is suppressed for gluon energies $\omega > \Gamma$ by propagator/timescale effects; thus only soft gluons may contribute appreciably.
2. Nonperturbative in the hadroformation process: normally modelled by a colour rearrangement between the partons produced in the two resonance decays and in the subsequent parton showers.
3. Nonperturbative in the purely hadronic phase: best exemplified by Bose–Einstein effects; see next section.

The above topics are deeply related to the unsolved problems of strong interactions: confinement dynamics, $1/N_{\mathrm{C}}^2$ effects, quantum mechanical interferences, etc. Thus they offer an opportunity to study the dynamics of unstable particles, and new ways to probe confinement dynamics in space and time [Gus88, Sjö94], *but* they also risk to limit or even spoil precision measurements.

The reconnection scenarios outlined in [Sjö94a] are now available, plus also an option along the lines suggested in [Gus94]. Currently they can only be invoked in process 25, $\mathrm{e^+e^-} \to \mathrm{W^+W^-} \to \mathrm{q_1\overline{q}_2q_3\overline{q}_4}$, which is the most interesting one for the foreseeable future. (Process 22, $\mathrm{e^+e^-} \to \gamma^*/\mathrm{Z}^0\,\gamma^*/\mathrm{Z}^0 \to \mathrm{q_1\overline{q}_1q_2\overline{q}_2}$ can also be used, but the travel distance is calculated based only on the Z^0 propagator part. Thus, the description in scenarios I, II and II′ below would not be sensible e.g. for a light-mass $\gamma^*\gamma^*$ pair.) If normally the event is considered as consisting of two separate colour singlets, $\mathrm{q_1\overline{q}_2}$ from the $\mathrm{W^+}$ and $\mathrm{q_3\overline{q}_4}$ from the $\mathrm{W^-}$, a colour rearrangement can give two new colour singlets $\mathrm{q_1\overline{q}_4}$ and $\mathrm{q_3\overline{q}_2}$. It therefore leads to a different hadronic final state, although differences usually turn out to be subtle and difficult to isolate [Nor97]. When also gluon emission is considered, the number of potential reconnection topologies increases.

Since hadronization is not understood from first principles, it is important to remember that we deal with model building rather than exact calculations. We will use the standard Lund string fragmentation model [And83] as a starting point, but have to extend it considerably. The string is here to be viewed as a Lorentz covariant representation of a linear confinement field.

The string description is entirely probabilistic, i.e. any negative-sign interference effects are absent. This means that the original colour singlets $\mathrm{q_1\overline{q}_2}$ and $\mathrm{q_3\overline{q}_4}$ may transmute to new singlets $\mathrm{q_1\overline{q}_4}$ and $\mathrm{q_3\overline{q}_2}$, but that any effects e.g. of $\mathrm{q_1q_3}$ or $\mathrm{\overline{q}_2\overline{q}_4}$ dipoles are absent. In this respect, the nonperturbative discussion is more limited in outlook than a corresponding perturbative one. However, note that dipoles such as $\mathrm{q_1q_3}$ do not correspond to colour singlets, and can therefore not survive in the long-distance limit of the theory, i.e. they have to disappear in the hadronization phase.

The imagined time sequence is the following. The $\mathrm{W^+}$ and $\mathrm{W^-}$ fly apart from their common production vertex and decay at some distance. Around each of these decay vertices, a perturbative parton shower evolves from an original $\mathrm{q\overline{q}}$ pair. The typical distance that a virtual parton (of mass $m \sim 10$ GeV, say, so that it can produce separate jets in the hadronic final state) travels before branching is comparable with the average $\mathrm{W^+W^-}$ separation, but shorter than the fragmentation time. Each W can therefore effectively be viewed as instantaneously decaying into a string spanned between the partons, from a quark end via a number of intermediate gluons to the antiquark end. The strings expand, both transversely and longitudinally, at a speed limited by that of light. They eventually fragment into hadrons and disappear. Before that time, however, the string from the $\mathrm{W^+}$ and the one from the $\mathrm{W^-}$ may overlap. If so, there is some

probability for a colour reconnection to occur in the overlap region. The fragmentation process is then modified.

The Lund string model does not constrain the nature of the string fully. At one extreme, the string may be viewed as an elongated bag, i.e. as a flux tube without any pronounced internal structure. At the other extreme, the string contains a very thin core, a vortex line, which carries all the topological information, while the energy is distributed over a larger surrounding region. The latter alternative is the chromoelectric analogue to the magnetic flux lines in a type II superconductor, whereas the former one is more akin to the structure of a type I superconductor. We use them as starting points for two contrasting approaches, with nomenclature inspired by the superconductor analogy.

In scenario I, the reconnection probability is proportional to the space–time volume over which the W^+ and W^- strings overlap, with saturation at unit probability. This probability is calculated as follows. In the rest frame of a string piece expanding along the $\pm z$ direction, the colour field strength is assumed to be given by

$$\Omega(\mathbf{x},t) = \exp\left\{-(x^2+y^2)/2r_{\mathrm{had}}^2\right\}\ \theta(t-|\mathbf{x}|)\ \exp\left\{-(t^2-z^2)/\tau_{\mathrm{frag}}^2\right\}\ . \tag{231}$$

The first factor gives a Gaussian fall-off in the transverse directions, with a string width $r_{\mathrm{had}} \approx 0.5$ fm of typical hadronic dimensions. The time retardation factor $\theta(t-|\mathbf{x}|)$ ensures that information on the decay of the W spreads outwards with the speed of light. The last factor gives the probability that the string has not yet fragmented at a given proper time along the string axis, with $\tau_{\mathrm{frag}} \approx 1.5$ fm. For a string piece e.g. from the W^+ decay, this field strength has to be appropriately rotated, boosted and displaced to the W^+ decay vertex. In addition, since the W^+ string can be made up of many pieces, the string field strength $\Omega^+_{\mathrm{max}}(\mathbf{x},t)$ is defined as the maximum of all the contributing Ω^+'s in the given point. The probability for a reconnection to occur is now given by

$$\mathcal{P}_{\mathrm{recon}} = 1 - \exp\left(-k_{\mathrm{I}} \int \mathrm{d}^3\mathbf{x}\,\mathrm{d}t\ \Omega^+_{\mathrm{max}}(\mathbf{x},t)\,\Omega^-_{\mathrm{max}}(\mathbf{x},t)\right)\ , \tag{232}$$

where k_{I} is a free parameter. If a reconnection occurs, the space–time point for this reconnection is selected according to the differential probability $\Omega^+_{\mathrm{max}}(\mathbf{x},t)\,\Omega^-_{\mathrm{max}}(\mathbf{x},t)$. This defines the string pieces involved and the new colour singlets.

In scenario II it is assumed that reconnections can only take place when the core regions of two string pieces cross each other. This means that the transverse extent of strings can be neglected, which leads to considerable simplifications compared with the previous scenario. The position of a string piece at time t is described by a one-parameter set $\mathbf{x}(t,\alpha)$, where $0 \le \alpha \le 1$ is used to denote the position along the string. To find whether two string pieces i and j from the W^+ and W^- decays cross, it is sufficient to solve the equation system $\mathbf{x}_i^+(t,\alpha^+) = \mathbf{x}_j^-(t,\alpha^-)$ and to check that this (unique) solution is in the physically allowed domain. Further, it is required that neither string piece has had time to fragment, which gives two extra suppression factors of the form $\exp\{-\tau^2/\tau_{\mathrm{frag}}^2\}$, with τ the proper lifetime of each string piece at the point of crossing, i.e. as in scenario I. If there are several string crossings, only the one that occurs first is retained. The II′ scenario is a variant of scenario II, with the requirement that a reconnection is allowed only if it leads to a reduction of the string length.

Other models include a simplified implementation of the 'GH' model [Gus94], where the reconnection is selected solely based on the criterion of a reduced string length. The 'instantaneous' and 'intermediate' scenarios are two toy models. In the former (which is equivalent to that in [Gus88]) the two reconnected systems $q_1\overline{q}_4$ and $q_3\overline{q}_2$ are immediately formed and then subsequently shower and fragment independently of each other. In the latter, a reconnection occurs between the shower and fragmentation stages. One has to bear in mind that the last two 'optimistic' (from the connectometry point of view) toy approaches are oversimplified

extremes and are not supposed to correspond to the true nature. These scenarios may be useful for reference purposes, but are essentially already excluded by data.

While interconnection effects are primarily viewed as hadronization physics, their implementation is tightly coupled to the event generation of a few specific processes, and not to the generic handronization machinery. Therefore the relevant main switch `MSTP(115)` and parameters `PARP(115) - PARP(120)` are described in subsection 9.3.

12.4.3 Bose–Einstein effects

A crude option for the simulation of Bose–Einstein effects is included since long, but is turned off by default. In view of its shortcomings, alternative descriptions have been introduced that try to overcome at least some of them [Lön95].

The detailed BE physics is not that well understood, see e.g. [Lör89]. What is offered is an algorithm, more than just a parameterization (since very specific assumptions and choices have been made), and yet less than a true model (since the underlying physics picture is rather fuzzy). In this scheme, the fragmentation is allowed to proceed as usual, and so is the decay of short-lived particles like ρ. Then pairs of identical particles, π^+ say, are considered one by one. The Q_{ij} value of a pair i and j is evaluated,

$$Q_{ij} = \sqrt{(p_i + p_j)^2 - 4m^2} \; , \tag{233}$$

where m is the common particle mass. A shifted (smaller) Q'_{ij} is then to be found such that the (infinite statistics) ratio $f_2(Q)$ of shifted to unshifted Q distributions is given by the requested parameterization. The shape may be chosen either exponential or Gaussian,

$$f_2(Q) = 1 + \lambda \exp\left(-(Q/d)^r\right) , \quad r = 1 \text{ or } 2 \; . \tag{234}$$

(In fact, the distribution has to dip slightly below unity at Q values outside the Bose enhancement region, from conservation of total multiplicity.) If the inclusive distribution of Q_{ij} values is assumed given just by phase space, at least at small relative momentum then, with $\mathrm{d}^3p/E \propto Q^2\,\mathrm{d}Q/\sqrt{Q^2+4m^2}$, then Q'_{ij} is found as the solution to the equation

$$\int_0^{Q_{ij}} \frac{Q^2\,\mathrm{d}Q}{\sqrt{Q^2+4m^2}} = \int_0^{Q'_{ij}} f_2(Q)\,\frac{Q^2\,\mathrm{d}Q}{\sqrt{Q^2+4m^2}} \; . \tag{235}$$

The change of Q_{ij} can be translated into an effective shift of the three-momenta of the two particles, if one uses as extra constraint that the total three-momentum of each pair be conserved in the c.m. frame of the event. Only after all pairwise momentum shifts have been evaluated, with respect to the original momenta, are these momenta actually shifted, for each particle by the (three-momentum) sum of evaluated shifts. The total energy of the event is slightly reduced in the process, which is compensated by an overall rescaling of all c.m. frame momentum vectors. It can be discussed which are the particles to involve in this rescaling. Currently the only exceptions to using everything are leptons and neutrinos coming from resonance decays (such as W's) and photons radiated by leptons (also in initial state radiation). Finally, the decay chain is resumed with more long-lived particles like π^0.

Two comments can be made. The Bose–Einstein effect is here interpreted almost as a classical force acting on the 'final state', rather than as a quantum mechanical phenomenon on the production amplitude. This is not a credo, but just an ansatz to make things manageable. Also, since only pairwise interactions are considered, the effects associated with three or more nearby particles tend to get overestimated. (More exact, but also more time-consuming methods may be found in [Zaj87].) Thus the input λ may have to be chosen smaller than what one wants to get out. (On the other hand, many of the pairs of an event contains at least one particle produced

in some secondary vertex, like a D decay. This reduces the fraction of pairs which may contribute to the Bose–Einstein effects, and thus reduces the potential signal.) This option should therefore be used with caution, and only as a first approximation to what Bose–Einstein effects can mean.

Probably the largest weakness of the above approach is the issue how to conserve the total four-momentum. It preserves three-momentum locally, but at the expense of not conserving energy. The subsequent rescaling of all momenta by a common factor (in the rest frame of the event) to restore energy conservation is purely *ad hoc*. For studies of a single Z^0 decay, it can plausibly be argued that such a rescaling does minimal harm. The same need not hold for a pair of resonances. Indeed, studies [Lön95] show that this global rescaling scheme, which we will denote BE_0, introduces an artificial negative shift in the reconstructed W mass, making it difficult (although doable) to study the true BE effects in this case. This is one reason to consider alternatives.

The global rescaling is also running counter to our philosophy that BE effects should be local. To be more specific, we assume that the energy density of the string is a fixed quantity. To the extent that a pair of particles have their four-momenta slightly shifted, the string should act as a 'commuting vessel', providing the difference to other particles produced in the same local region of the string. What this means in reality is still not completely specified, so further assumptions are necessary. In the following we discuss four possible algorithms, whereof the last two are based strictly on the local conservation aspect above, while the first two are attempting a slightly different twist to the locality concept. All are based on calculating an additional shift $\delta\mathbf{r}_k^l$ for some pairs of particles, where particles k and l need not be identical bosons. In the end each particle momentum will then be shifted to $\mathbf{p}_i' = \mathbf{p}_i + \sum_{j\neq i}\delta\mathbf{p}_i^j + \alpha\sum_{k\neq i}\delta\mathbf{r}_i^k$, with the parameter α adjusted separately for each event so that the total energy is conserved.

In the first approach we emulate the criticism of the global event weight methods with weights always above unity, as being intrinsically unstable. It appears more plausible that weights fluctuate above and below unity. For instance, the simple pair symmetrization weight is $1+\cos(\Delta x\cdot\Delta p)$, with the $1+\lambda\exp(-Q^2R^2)$ form only obtained after integration over a Gaussian source. Non-Gaussian sources give oscillatory behaviours

If weights above unity correspond to a shift of pairs towards smaller relative Q values, the below-unity weights instead give a shift towards larger Q. One therefore is lead to a picture where very nearby identical particles are shifted closer, those somewhat further are shifted apart, those even further yet again shifted closer, and so on. Probably the oscillations dampen out rather quickly, as indicated both by data and by the global model studies. We therefore simplify by simulating only the first peak and dip. Furthermore, to include the desired damping and to make contact with our normal generation algorithm (for simplicity), we retain the Gaussian form, but the standard $f_2(Q) = 1+\lambda\exp(-Q^2R^2)$ is multiplied by a further factor $1+\alpha\lambda\exp(-Q^2R^2/9)$. The factor $1/9$ in the exponential, i.e. a factor 3 difference in the Q variable, is consistent with data and also with what one might expect from a dampened cos form, but should be viewed more as a simple ansatz than having any deep meaning.

In the algorithm, which we denote BE_3, $\delta\mathbf{r}_i^j$ is then non-zero only for pairs of identical bosons, and is calculated in the same way as $\delta\mathbf{p}_i^j$, with the additional factor $1/9$ in the exponential. As explained above, the $\delta\mathbf{r}_i^j$ shifts are then scaled by a common factor α that ensures total energy conservation. It turns out that the average α needed is ≈ -0.2. The negative sign is exactly what we want to ensure that $\delta\mathbf{r}_i^j$ corresponds to shifting a pair apart, while the order of α is consistent with the expected increase in the number of affected pairs when a smaller effective radius $R/3$ is used. One shortcoming of the method, as implemented here, is that the input $f_2(0)$ is not quite 2 for $\lambda = 1$ but rather $(1+\lambda)(1+\alpha\lambda) \approx 1.6$. This could be solved by starting off with an input λ somewhat above unity.

The second algorithm, denoted BE_{32}, is a modification of the BE_3 form intended to give

$f_2(0) = 1+\lambda$. The ansatz is

$$f_2(Q) = \left\{1+\lambda\exp(-Q^2R^2)\right\}\left\{1+\alpha\lambda\exp(-Q^2R^2/9)\left(1-\exp(-Q^2R^2/4)\right)\right\} \ , \qquad (236)$$

applied to identical pairs. The combination $\exp(-Q^2R^2/9)\left(1-\exp(-Q^2R^2/4)\right)$ can be viewed as a Gaussian, smeared-out representation of the first dip of the cos function. As a technical trick, the $\delta\mathbf{r}_i^j$ are found as in the BE_3 algorithm and thereafter scaled down by the $1-\exp(-Q^2R^2/4)$ factor. (This procedure does not quite reproduce the formalism of eq. (235), but comes sufficiently close for our purpose, given that the ansatz form in itself is somewhat arbitrary.) One should note that, even with the above improvement relative to the BE_3 scheme, the observable two-particle correlation is lower at small Q than in the BE_0 algorithm, so some further tuning of λ could be required. In this scheme, $\langle\alpha\rangle \approx -0.25$.

In the other two schemes, the original form of $f_2(Q)$ is retained, and the energy is instead conserved by picking another pair of particles that are shifted apart appropriately. That is, for each pair of identical particles i and j, a pair of non-identical particles, k and l, neither identical to i or j, is found in the neighbourhood of i and j. For each shift $\delta\mathbf{p}_i^j$, a corresponding $\delta\mathbf{r}_k^l$ is found so that the total energy and momentum in the i,j,k,l system is conserved. However, the actual momentum shift of a particle is formed as the composant of many contributions, so the above pair compensation mechanism is not perfect. The mismatch is reflected in a non-unit value α used to rescale the $\delta\mathbf{r}_k^l$ terms.

The k,l pair should be the particles 'closest' to the pair affected by the BE shift, in the spirit of local energy conservation. One option would here have been to 'look behind the scenes' and use information on the order of production along the string. However, once decays of short-lived particles are included, such an approach would still need arbitrary further rules. We therefore stay with the simplifying principle of only using the produced particles.

Looking at $\mathrm{W}^+\mathrm{W}^-$ (or $\mathrm{Z}^0\mathrm{Z}^0$) events and a pair i,j with both particles from the same W, it is not obvious whether the pair k,l should also be selected only from this W or if all possible pairs should be considered. Below we have chosen the latter as default behaviour, but the former alternative is also studied below.

One obvious measure of closeness is small invariant mass. A first choice would then be to pick the combination that minimizes the invariant mass m_{ijkl} of all four particles. However, such a procedure does not reproduce the input $f_2(Q)$ shape very well: both the peak height and peak width are significantly reduced, compared with what happens in the BE_0 algorithm. The main reason is that either of k or l may have particles identical to itself in its local neighbourhood. The momentum compensation shift of k is at random, more or less, and therefore tends to smear the BE signal that could be introduced relative to k's identical partner. Note that, if k and its partner are very close in Q to start with, the relative change δQ required to produce a significant BE effect is very small, approximately $\delta Q \propto Q$. The momentum compensation shift on k can therefore easily become larger than the BE shift proper.

It is therefore necessary to disfavour momentum compensation shifts that break up close identical pairs. One alternative would have been to share the momentum conservation shifts suitably inside such pairs. We have taken a simpler course, by introducing a suppression factor $1-\exp(-Q_k^2R^2)$ for particle k, where Q_k is the Q value between k and its nearest identical partner. The form is fixed such that a $Q_k = 0$ is forbidden and then the rise matches the shape of the BE distribution itself. Specifically, in the third algorithm, BE_m, the pair k,l is chosen so that the measure

$$W_{ijkl} = \frac{(1-\exp(-Q_k^2R^2))(1-\exp(-Q_l^2R^2))}{m_{ijkl}^2} \qquad (237)$$

is maximized. The average α value required to rescale for the effect of multiple shifts is 0.73, i.e. somewhat below unity.

The BE_λ algorithm is inspired by the so-called λ measure [And89] (not the be confused with the λ parameter of $f_2(Q)$). It corresponds to a string length in the Lund string fragmentation

framework. It can be shown that partons in a string are colour-connected in a way that tends to minimize this measure. The same is true for the ordering of the produced hadrons, although with large fluctuations. As above, having identical particles nearby to k, l gives undesirable side effects. Therefore the selection is made so that

$$W_{ijkl} = \frac{(1-\exp(-Q_k^2 R^2))(1-\exp(-Q_l^2 R^2))}{\min_{(12\text{ permutations})}(m_{ij}m_{jk}m_{kl}, m_{ij}m_{jl}m_{lk}, \ldots)} \tag{238}$$

is maximized. The denominator is intended to correspond to $\exp(\lambda)$. For cases where particles i and j comes from the same string, this would favour compensating the energy using particles that are close by and in the same string. We find $\langle\alpha\rangle \approx 0.73$, as above.

The main switches and parameters affecting the Bose–Einstein algorithm are `MSTJ(51) - MSTJ(57)` and `PARJ(91) - PARJ(96)`.

13 Particles and Their Decays

Particles are the building blocks from which events are constructed. We here use the word 'particle' in its broadest sense, i.e. including partons, resonances, hadrons, and so on, subgroups we will describe in the following. Each particle is characterized by some quantities, such as charge and mass. In addition, many of the particles are unstable and subsequently decay. This section contains a survey of the particle content of the programs, and the particle properties assumed. In particular, the decay treatment is discussed. Some particle and decay properties form part already of the hard subprocess description, and are therefore described in sections 6, 7 and 8.

13.1 The Particle Content

In order to describe both current and potential future physics, a number of different particles are needed. A list of some particles, along with their codes, is given in section 5.1. Here we therefore emphasize the generality rather than the details.

Four full generations of quarks and leptons are included in the program, although indications from LEP strongly suggest that only three exist in Nature. The PDG naming convention for the fourth generation is to repeat the third one but with a prime: b', t' τ' and ν'_τ. Quarks may appear either singly or in pairs; the latter are called diquarks and are characterized by their flavour content and their spin. A diquark is always assumed to be in a colour antitriplet state.

The colour neutral hadrons may be build up from the five lighter coloured quarks (and diquarks). Six full meson multiplets are included and two baryon ones, see section 12.1. In addition, K^0_S and K^0_L are considered as separate particles coming from the 'decay' of K^0 and $\overline{K}^0$ (or, occasionally, produced directly).

Other particles from the Standard Model include the gluon g, the photon γ, the intermediate gauge bosons Z^0 and $W^\pm$, and the standard Higgs h^0. Non-standard particles include additional gauge bosons, Z'^0 and $W'^\pm$, additional Higgs bosons H^0, A^0 and $H^\pm$, a leptoquark L_Q a horizontal gauge boson R^0, technicolor and supersymmetric particles, and more.

From the point of view of usage inside the programs, particles may be subdivided into three classes, partly overlapping.

1. A parton is generically any object which may be found in the wave function of the incoming beams, and may participate in initial- or final-state showers. This includes what is normally meant by partons, i.e. quarks and gluons, but here also leptons and photons. In a few cases other particles may be classified as partons in this sense.
2. A resonance is an unstable particle produced as part of the hard process, and where the decay treatment normally is also part of the hard process. Resonance partial widths are perturbatively calculable, and therefore it is possible to dynamically recalculate branching ratios as a function of the mass assigned to a resonance. Resonances includes particles like the Z^0 and other massive gauge bosons and Higgs particles, in fact everything with a mass above the b quark and additionally also a lighter γ^*.
3. Hadrons and their decay products, i.e. mesons and baryons produced either in the fragmentation process, in secondary decays or as part of the beam remnant treatment, but not directly as part of the hard process (except in a few special cases). Hadrons may be stable or unstable. Branching ratios are not assumed perturbatively calculable, and can therefore be set freely. Also leptons and photons produced in decays belong to this class. In practice, this includes everything up to and including b quarks in mass (except a light γ^*, see above).

Usually the subdivision above is easy to understand and gives you the control you would expect. Thus the restriction on the allowed decay modes of a resonance will directly affect the cross section of a process, while this is not the case for an ordinary hadron, since in the latter case there is no precise theory knowledge on the set of decay modes and branching ratios.

13.2 Masses, Widths and Lifetimes

13.2.1 Masses

Quark masses are not particularly well defined. In the program it is necessary to make use of three kinds of masses, kinematical, running current algebra ones and constituent ones. The first ones are relevant for the kinematics in hard processes, e.g. in $gg \to c\bar{c}$, and are partly fixed by such considerations [Nor98]. The second define couplings to Higgs particles, and also other mass-related couplings in models for physics beyond the standard model. Both these kinds directly affect cross sections in processes. Constituent masses, finally, are used to derive the masses of hadrons, for some not yet found ones, and e.g. to gauge the remainder mass below which the final two hadrons are to be produced in string fragmentation.

The first set of values are the ones stored in the standard mass array `PMAS`. The starting values of the running masses are stored in `PARF(91) - PARF(96)`, with the running calculated in the `PYMRUN` function. Constituent masses are also stored in the `PARF` array, above position 101. We maintain this distinction for the five first flavours, and partly for top, using the following values by default:

quark	kinematical	current algebra mass	constituent mass
d	0.33 GeV	0.0099 GeV	0.325 GeV
u	0.33 GeV	0.0056 GeV	0.325 GeV
s	0.5 GeV	0.199 GeV	0.5 GeV
c	1.5 GeV	1.35 GeV	1.6 GeV
b	4.8 GeV	4.5 GeV	5.0 GeV
t	175 GeV	165 GeV	—

For top no constituent mass is defined, since it does not form hadrons. For hypothetical fourth generation quarks only one set of mass values is used, namely the one in `PMAS`. Constituent masses for diquarks are defined as the sum of the respective quark masses. The gluon is always assumed massless.

Particle masses, when known, are taken from ref. [PDG96]. Hypothesized particles, such as fourth generation fermions and Higgs bosons, are assigned some not unreasonable set of default values, in the sense of where you want to search for them in the not too distant future. Here it is understood that you will go in and change the default values according to your own opinions at the beginning of a run.

The total number of hadrons in the program is very large, whereof some are not yet discovered (like charm and bottom baryons). There the masses are built up, when needed, from the constituent masses. For this purpose one uses formulae of the type [DeR75]

$$m = m_0 + \sum_i m_i + k\, m_{\mathrm{d}}^2 \sum_{i<j} \frac{\langle \sigma_i \cdot \sigma_j \rangle}{m_i\, m_j} \,, \tag{239}$$

i.e. one constant term, a sum over constituent masses and a spin-spin interaction term for each quark pair in the hadron. The constants m_0 and k are fitted from known masses, treating mesons and baryons separately. For mesons with orbital angular momentum $L = 1$ the spin-spin coupling is assumed vanishing, and only m_0 is fitted. One may also define 'constituent diquarks masses' using the formula above, with a k value $2/3$ that of baryons. The default values are:

multiplet	m_0	k
pseudoscalars and vectors	0.	0.16 GeV
axial vectors ($S = 0$)	0.50 GeV	0.
scalars	0.45 GeV	0.
axial vectors ($S = 1$)	0.55 GeV	0.
tensors	0.60 GeV	0.
baryons	0.11 GeV	0.048 GeV
diquarks	0.077 GeV	0.048 GeV.

Unlike earlier versions of the program, the actual hadron values are hardcoded, i.e. are unaffected by any change of the charm or bottom quark masses.

13.2.2 Widths

A width is calculated perturbatively for those resonances which appear in the PYTHIA hard process generation machinery. The width is used to select masses in hard processes according to a relativistic Breit–Wigner shape. In many processes the width is allowed to be $\hat{s}$-dependent, see section 7.3.

Other particle masses, as discussed so far, have been fixed at their nominal value. We now have to consider the mass broadening for short-lived particles such as ρ, K^* or Δ. Compared to the Z^0, it is much more difficult to describe the ρ resonance shape, since nonperturbative and threshold effects act to distort the naïve shape. Thus the ρ mass is limited from below by its decay $\rho \to \pi\pi$, but also from above, e.g. in the decay $\phi \to \rho\pi$. Normally thus the allowed mass range is set by the most constraining decay chains. Some rare decay modes, specifically $\rho^0 \to \eta\gamma$ and $a_2 \to \eta'\pi$, are not allowed to have full impact, however. Instead one accepts an imperfect rendering of the branching ratio, as some low-mass ρ^0/a_2 decays of the above kind are rejected in favour of other decay channels. In some decay chains, several mass choices are coupled, like in $a_2 \to \rho\pi$, where also the a_2 has a non-negligible width. Finally, there are some extreme cases, like the f_0, which has a nominal mass below the KK threshold, but a tail extending beyond that threshold, and therefore a non-negligible branching ratio to the KK channel.

In view of examples like these, no attempt is made to provide a full description. Instead a simplified description is used, which should be enough to give the general smearing of events due to mass broadening, but maybe not sufficient for detailed studies of a specific resonance. By default, hadrons are therefore given a mass distribution according to a non-relativistic Breit–Wigner

$$\mathcal{P}(m)\,\mathrm{d}m \propto \frac{1}{(m-m_0)^2+\Gamma^2/4}\,\mathrm{d}m\ . \tag{240}$$

Leptons and resonances not taken care of by the hard process machinery are distributed according to a relativistic Breit–Wigner

$$\mathcal{P}(m^2)\,\mathrm{d}m^2 \propto \frac{1}{(m^2-m_0^2)^2+m_0^2\Gamma^2}\,\mathrm{d}m^2\ . \tag{241}$$

Here m_0 and Γ are the nominal mass and width of the particle. The Breit–Wigner shape is truncated symmetrically, $|m-m_0| < \delta$, with δ arbitrarily chosen for each particle so that no problems are encountered in the decay chains. It is possible to switch off the mass broadening, or to use either a non-relativistic or a relativistic Breit–Wigners everywhere.

The f_0 problem has been 'solved' by shifting the f_0 mass to be slightly above the KK threshold and have vanishing width. Then kinematics in decays $f_0 \to KK$ is reasonably well modelled. The f_0 mass is too large in the $f_0 \to \pi\pi$ channel, but this does not really matter, since one anyway is far above threshold here.

13.2.3 Lifetimes

Clearly the lifetime and the width of a particle are inversely related. For practical applications, however, any particle with a non-negligible width decays too close to its production vertex for the lifetime to be of any interest. In the program, the two aspects are therefore considered separately. Particles with a non-vanishing nominal proper lifetime $\tau_0 = \langle\tau\rangle$ are assigned an actual lifetime according to

$$\mathcal{P}(\tau)\,\mathrm{d}\tau \propto \exp(-\tau/\tau_0)\,\mathrm{d}\tau\ , \tag{242}$$

i.e. a simple exponential decay is assumed. Since the program uses dimensions where the speed of light $c \equiv 1$, and space dimensions are in mm, then actually the unit of $c\tau_0$ is mm and of τ_0 itself $\mathrm{mm}/c \approx 3.33 \times 10^{-12}$ s.

If a particle is produced at a vertex $v = (\mathbf{x}, t)$ with a momentum $p = (\mathbf{p}, E)$ and a lifetime τ, the decay vertex position is assumed to be

$$v' = v + \tau\frac{p}{m}\ , \tag{243}$$

where m is the mass of the particle. With the primary interaction (normally) in the origin, it is therefore possible to construct all secondary vertices in parallel with the ordinary decay treatment.

The formula above does not take into account any detector effects, such as a magnetic field. It is therefore possible to stop the decay chains at some suitable point, and leave any subsequent decay treatment to the detector simulation program. One may select that particles are only allowed to decay if they have a nominal lifetime τ_0 shorter than some given value or, alternatively, if their decay vertices $\mathbf{x}'$ are inside some spherical or cylindrical volume around the origin.

13.3 Decays

Several different kinds of decay treatment are used in the program, depending on the nature of the decay. Not discussed here are the decays of resonances which are handled as part of the hard process.

13.3.1 Strong and electromagnetic decays

The decays of hadrons containing the 'ordinary' u, d and s quarks into two or three particles are known, and branching ratios may be found in [PDG96]. (At least for the lowest-lying states; the four $L = 1$ meson multiplets are considerably less well known.) We normally assume that the momentum distributions are given by phase space. There are a few exceptions, where the phase space is weighted by a matrix-element expression, as follows.

In ω and ϕ decays to $\pi^+\pi^-\pi^0$, a matrix element of the form

$$|\mathcal{M}|^2 \propto |\mathbf{p}_{\pi^+} \times \mathbf{p}_{\pi^-}|^2 \tag{244}$$

is used, with the $\mathbf{p}_\pi$ the pion momenta in the rest frame of the decay. (Actually, what is coded is the somewhat more lengthy Lorentz invariant form of the expression above.)

Consider the decay chain $P_0 \to P_1 + V \to P_1 + P_2 + P_3$, with P representing pseudoscalar mesons and V a vector one. Here the decay angular distribution of the V in its rest frame is

$$|\mathcal{M}|^2 \propto \cos^2\theta_{02}\ , \tag{245}$$

where θ_{02} is the angle between P_0 and P_2. The classical example is $\mathrm{D} \to \mathrm{K}^*\pi \to \mathrm{K}\pi\pi$. If the P_1 is replaced by a γ, the angular distribution in the V decay is instead $\propto \sin^2\theta_{02}$.

In Dalitz decays, π^0 or $\eta \to e^+e^-\gamma$, the mass m^* of the e^+e^- pair is selected according to

$$\mathcal{P}(m^{*2})\,\mathrm{d}m^{*2} \propto \frac{\mathrm{d}m^{*2}}{m^{*2}} \left(1+\frac{2m_e^2}{m^{*2}}\right) \sqrt{1-\frac{4m_e^2}{m^{*2}}} \left(1-\frac{m^{*2}}{m_{\pi,\eta}^2}\right)^3 \frac{1}{(m_\rho^2-m^{*2})^2+m_\rho^2\Gamma_\rho^2} \,. \tag{246}$$

The last factor, the VMD-inspired ρ^0 propagator, is negligible for π^0 decay. Once the m^* has been selected, the angular distribution of the e^+e^- pair is given by

$$|\mathcal{M}|^2 \propto (m^{*2}-2m_e^2)\left\{(p_\gamma p_{e^+})^2+(p_\gamma p_{e^-})^2\right\}+4m_e^2\left\{(p_\gamma p_{e^+})(p_\gamma p_{e^-})+(p_\gamma p_{e^+})^2+(p_\gamma p_{e^-})^2\right\} \,. \tag{247}$$

Also a number of simple decays involving resonances of heavier hadrons, e.g. $\Sigma_c^0 \to \Lambda_c^+\pi^-$ or $B^{*-} \to B^-\gamma$ are treated in the same way as the other two-particle decays.

13.3.2 Weak decays of the τ lepton

For the τ lepton, an explicit list of decay channels has been put together, which includes channels with up to five final-state particles, some of which may be unstable and subsequently decay to produce even larger total multiplicities.

The leptonic decays $\tau^- \to \nu_\tau \ell^- \overline{\nu}_\ell$, where ℓ is e or μ, are distributed according to the standard $V-A$ matrix element

$$|\mathcal{M}|^2 = (p_\tau p_{\overline{\nu}_\ell})(p_\ell p_{\nu_\tau}) \,. \tag{248}$$

(The corresponding matrix element is also used in μ decays, but normally the μ is assumed stable.)

In τ decays to hadrons, the hadrons and the ν_τ are distributed according to phase space times the factor $x_\nu\,(3-x_\nu)$, where $x_\nu = 2E_\nu/m_\tau$ in the rest frame of the τ. The latter factor is the ν_τ spectrum predicted by the parton level $V-A$ matrix element, and therefore represents an attempt to take into account that the ν_τ should take a larger momentum fraction than given by phase space alone.

The probably largest shortcoming of the τ decay treatment is that no polarization effects are included, i.e. the τ is always assumed to decay isotropically. Usually this is not correct, since a τ is produced polarized in Z^0 and $W^\pm$ decays. The `PYTAUD` routine provides a generic interface to an external τ decay library, such as TAUOLA [Jad91], where such effects could be handled (see also `MSTJ(28)`).

13.3.3 Weak decays of charm hadrons

The charm hadrons have a mass in an intermediate range, where the effects of the naïve $V-A$ weak decay matrix element is partly but not fully reflected in the kinematics of final-state particles. Therefore different decay strategies are combined. We start with hadronic decays, and subsequently consider semileptonic ones.

For the four 'main' charm hadrons, D^+, D^0, D_s^+ and Λ_c^+, a number of branching ratios are already known. The known branching ratios have been combined with reasonable guesses, to construct more or less complete tables of all channels. For hadronic decays of D^0 and D^+, where rather much is known, all channels have an explicitly listed particle content. However, only for the two-body decays and some three-body decays is resonance production properly taken into account. It means that the experimentally measured branching ratio for a $K\pi\pi$ decay channel, say, is represented by contributions from a direct $K\pi\pi$ channel as well as from indirect ones, such as $K^*\pi$ and $K\rho$. For a channel like $K\pi\pi\pi\pi$, on the other hand, not all possible combinations of resonances (many of which would have to be off mass shell to have kinematics work out) are included. This is more or less in agreement with the philosophy adopted in the PDG tables [PDG92]. For D_s^+ and Λ_c^+ knowledge is rather incomplete, and only two-body decay channels are listed. Final states with three or more hadron are only listed in terms of a flavour content.

The way the program works, it is important to include all the allowed decay channels up to a given multiplicity. Channels with multiplicity higher than this may then be generated according to a simple flavour combination scheme. For instance, in a D_s^+ decay, the normal quark content is $s\bar{s}u\bar{d}$, where one $\bar{s}$ is the spectator quark and the others come from the weak decay of the c quark. The spectator quark may also be annihilated, like in $D_s^+ \to u\bar{d}$. The flavour content to make up one or two hadrons is therefore present from the onset. If one decides to generate more hadrons, this means new flavour-antiflavour pairs have to be generated and combined with the existing flavours. This is done using the same flavour approach as in fragmentation, section 12.1.

In more detail, the following scheme is used.

1. The multiplicity is first selected. The D_s^+ and Λ_c^+ multiplicity is selected according to a distribution described further below. The program can also be asked to generate decays of a predetermined multiplicity.
2. One of the non-spectator flavours is selected at random. This flavour is allowed to 'fragment' into a hadron plus a new remaining flavour, using exactly the same flavour generation algorithm as in the standard jet fragmentation, section 12.1.
3. Step 2 is iterated until only one or two hadrons remain to be generated, depending on whether the original number of flavours is two or four. In each step one 'unpaired' flavour is replaced by another one as a hadron is 'peeled off', so the number of unpaired flavours is preserved.
4. If there are two flavours, these are combined to form the last hadron. If there are four, then one of the two possible pairings into two final hadrons is selected at random. To find the hadron species, the same flavour rules are used as when final flavours are combined in the joining of two jets.
5. If the sum of decay product masses is larger than the mass of the decaying particle, the flavour selection is rejected and the process is started over at step 1. Normally a new multiplicity is picked, but for D^0 and D^+ the old multiplicity is retained.
6. Once an acceptable set of hadrons has been found, these are distributed according to phase space.

The picture then is one of a number of partons moving apart, fragmenting almost like jets, but with momenta so low that phase-space considerations are enough to give the average behaviour of the momentum distribution. Like in jet fragmentation, endpoint flavours are not likely to recombine with each other. Instead new flavour pairs are created in between them. One should also note that, while vector and pseudoscalar mesons are produced at their ordinary relative rates, events with many vectors are likely to fail in step 5. Effectively, there is therefore a shift towards lighter particles, especially at large multiplicities.

When a multiplicity is to be picked, this is done according to a Gaussian distribution, centered at $c + n_q/4$ and with a width $\sqrt{c}$, with the final number rounded off to the nearest integer. The value for the number of quarks n_q is 2 or 4, as described above, and

$$c = c_1 \ln\left(\frac{m - \sum m_q}{c_2}\right) , \tag{249}$$

where m is the hadron mass and c_1 and c_2 have been tuned to give a reasonable description of multiplicities. There is always some lower limit for the allowed multiplicity; if a number smaller than this is picked the choice is repeated. Since two-body decays are explicitly enumerated for D_s^+ and Λ_c^+, there the minimum multiplicity is three.

Semileptonic branching ratios are explicitly given in the program for all the four particles discussed here, i.e. it is never necessary to generate the flavour content using the fragmentation description. This does not mean that all branching ratios are known; a fair amount of guesswork is involved for the channels with higher multiplicities, based on a knowledge of the inclusive semileptonic branching ratio and the exclusive branching ratios for low multiplicities.

In semileptonic decays it is not appropriate to distribute the lepton and neutrino momenta according to phase space. Instead the simple $V-A$ matrix element is used, in the limit that decay product masses may be neglected and that quark momenta can be replaced by hadron momenta. Specifically, in the decay $H \to \ell^+ \nu_\ell h$, where H is a charm hadron and h and ordinary hadron, the matrix element

$$|\mathcal{M}|^2 = (p_H p_\ell)(p_\nu p_h) \tag{250}$$

is used to distribute the products. It is not clear how to generalize this formula when several hadrons are present in the final state. In the program, the same matrix element is used as above, with p_h replaced by the total four-momentum of all the hadrons. This tends to favour a low invariant mass for the hadronic system compared with naïve phase space.

There are a few charm hadrons, such as Ξ_c and Ω_c, which decay weakly but are so rare that little is known about them. For these a simplified generic charm decay treatment is used. For hadronic decays only the quark content is given, and then a multiplicity and a flavour composition is picked at random, as already described. Semileptonic decays are assumed to produce only one hadron, so that $V-A$ matrix element can be simply applied.

13.3.4 Weak decays of bottom hadrons

Some exclusive branching ratios now are known for B decays. In this version, the B^0, B^+, B_s^0 and Λ_b^0 therefore appear in a similar vein to the one outlined above for D_s^+ and Λ_c^+ above. That is, all leptonic channels and all hadronic two-body decay channels are explicitly listed, while hadronic channels with three or more particles are only given in terms of a quark content. The B_c is exceptional, in that either the bottom or the charm quark may decay first, and in that annihilation graphs may be non-negligible. Leptonic and semileptonic channels are here given in full, while hadronic channels are only listed in terms of a quark content, with a relative composition as given in [Lus91]. No separate branching ratios are set for any of the other weakly decaying bottom hadrons, but instead a pure 'spectator quark' model is assumed, where the decay of the b quark is the same in all hadrons and the only difference in final flavour content comes from the spectator quark. Compared to the charm decays, the weak decay matrix elements are given somewhat larger importance in the hadronic decay channels.

In semileptonic decays $\mathrm{b} \to \mathrm{c}\ell^-\overline{\nu}_\ell$ the c quark is combined with the spectator antiquark or diquark to form one single hadron. This hadron may be either a pseudoscalar, a vector or a higher resonance (tensor etc.). The relative fraction of the higher resonances has been picked to be about 30%, in order to give a leptonic spectrum in reasonable experiment with data. (This only applies to the main particles B^0, B^+, B_s^0 and Λ_b^0; for the rest the choice is according to the standard composition in the fragmentation.) The overall process is therefore $H \to h\ell^-\overline{\nu}_\ell$, where H is a bottom antimeson or a bottom baryon (remember that $\overline{\mathrm{B}}$ is the one that contains a b quark), and the matrix element used to distribute momenta is

$$|\mathcal{M}|^2 = (p_H p_\nu)(p_\ell p_h) \ . \tag{251}$$

Again decay product masses have been neglected in the matrix element, but in the branching ratios the $\tau^-\overline{\nu}_\tau$ channel has been reduced in rate, compared with $\mathrm{e}^-\overline{\nu}_\mathrm{e}$ and $\mu^-\overline{\nu}_\mu$ ones, according to the expected mass effects. No CKM-suppressed decays $\mathrm{b} \to \mathrm{u}\ell^-\overline{\nu}_\ell$ are currently included.

In most multibody hadronic decays, e.g. $\mathrm{b} \to \mathrm{cd}\overline{\mathrm{u}}$, the c quark is again combined with the spectator flavour to form one single hadron, and thereafter the hadron and the two quark momenta are distributed according to the same matrix element as above, with $\ell^- \leftrightarrow \mathrm{d}$ and $\overline{\nu}_\ell \leftrightarrow \overline{\mathrm{u}}$. The invariant mass of the two quarks is calculated next. If this mass is so low that two hadrons cannot be formed from the system, the two quarks are combined into one single hadron. Else the same kind of approach as in hadronic charm decays is adopted, wherein a multiplicity is selected, a number of hadrons are formed and thereafter momenta are distributed according to phase space. The difference is that here the charm decay product is distributed according to the

$V - A$ matrix element, and only the rest of the system is assumed isotropic in its rest frame, while in charm decays all hadrons are distributed isotropically.

Note that the c quark and the spectator are assumed to form one colour singlet and the $d\overline{u}$ another, separate one. It is thus assumed that the original colour assignments of the basic hard process are better retained than in charm decays. However, sometimes this will not be true, and with about 20% probability the colour assignment is flipped around so that $c\overline{u}$ forms one singlet. (In the program, this is achieved by changing the order in which decay products are given.) In particular, the decay $b \to cs\overline{c}$ is allowed to give a $c\overline{c}$ colour-singlet state part of the time, and this state may collapse to a single J/ψ. Two-body decays of this type are explicitly listed for B^0, B^+, B_s^0 and Λ_b^0; while other J/ψ production channels appear from the flavour content specification.

The B^0–$\overline{B}^0$ and B_s^0–$\overline{B}_s^0$ systems mix before decay. This is optionally included. With a probability

$$\mathcal{P}_{\text{flip}} = \sin^2\left(\frac{x\tau}{2\langle\tau\rangle}\right) \tag{252}$$

a B is therefore allowed to decay like a $\overline{B}$, and vice versa. The mixing parameters are by default $x_d = 0.7$ in the B^0–$\overline{B}^0$ system and $x_s = 10$ in the B_s^0–$\overline{B}_s^0$ one.

In the past, the generic B meson and baryon decay properties were stored for 'particle' 85, now obsolete but not yet removed. This particle contains a description of the free b quark decay, with an instruction to find the spectator flavour according to the particle code of the actual decaying hadron.

13.3.5 Other decays

For onia spin 1 resonances, decay channels into a pair of leptons are explicitly given. Hadronic decays of the J/ψ are simulated using the flavour generation model introduced for charm. For Υ a fraction of the hadronic decays is into $q\overline{q}$ pairs, while the rest is into ggg or $gg\gamma$, using the matrix elements of eq. (43). The η_c and η_b are both allowed to decay into a gg pair, which then subsequently fragments. In Υ and η_b decays the partons are allowed to shower before fragmentation, but energies are too low for showering to have any impact.

Default branching ratios are given for resonances like the Z^0, the $W^\pm$, the t or the h^0. When PYTHIA is initialized, these numbers are replaced by branching ratios evaluated from the given masses. For Z^0 and $W^\pm$ the branching ratios depend only marginally on the masses assumed, while effects are large e.g. for the h^0. In fact, branching ratios may vary over the Breit–Wigner resonance shape, something which is also taken into account in the `PYRESD` description. Therefore the simpler resonance treatment of `PYDECY` is normally not so useful, and should be avoided. When it is used, a channel is selected according to the given fixed branching ratios. If the decay is into a $q\overline{q}$ pair, the quarks are allowed to shower and subsequently the parton system is fragmented.

14 The Fragmentation and Decay Program Elements

In this section we collect information on most of the routines and common block variables found in the fragmentation and decay descriptions of PYTHIA, plus a number of related low-level tasks.

14.1 Definition of Initial Configuration or Variables

With the use of the conventions described for the event record, it is possible to specify any initial jet/particle configuration. This task is simplified for a number of often occurring situations by the existence of the filling routines below. It should be noted that many users do not come in direct contact with these routines, since that is taken care of by higher-level routines for specific processes, particularly PYEVNT and PYEEVT.

Several calls to the routines can be combined in the specification. In case one call is enough, the complete fragmentation/decay chain may be simulated at the same time. At each call, the value of N is updated to the last line used for information in the call, so if several calls are used, they should be made with increasing IP number, or else N should be redefined by hand afterwards.

The routine PYJOIN is very useful to define the colour flow in more complicated parton configurations; thereby one can bypass the not so trivial rules for how to set the K(I,4) and K(I,5) colour-flow information.

The routine PYGIVE contains a facility to set various commonblock variables in a controlled and documented fashion.

```
CALL PY1ENT(IP,KF,PE,THE,PHI)
```

Purpose: to add one entry to the event record, i.e. either a parton or a particle.

IP : normally line number for the parton/particle. There are two exceptions.
If IP=0, line number 1 is used and PYEXEC is called.
If IP<0, line -IP is used, with status code K(-IP,2)=2 rather than 1; thus a parton system may be built up by filling all but the last parton of the system with IP<0.

KF : parton/particle flavour code.

PE : parton/particle energy. If PE is smaller than the mass, the parton/particle is taken to be at rest.

THE, PHI : polar and azimuthal angle for the momentum vector of the parton/particle.

```
CALL PY2ENT(IP,KF1,KF2,PECM)
```

Purpose: to add two entries to the event record, i.e. either a 2-parton system or two separate particles.

IP : normally line number for the first parton/particle, with the second in line IP+1. There are two exceptions.
If IP=0, lines 1 and 2 are used and PYEXEC is called.
If IP<0, lines -IP and -IP+1 are used, with status code K(I,1)=3, i.e. with special colour connection information, so that a parton shower can be generated by a PYSHOW call, followed by a PYEXEC call, if so desired (only relevant for partons).

KF1, KF2 : flavour codes for the two partons/particles.

PECM : ($= E_{\mathrm{cm}}$) the total energy of the system.

Remark: the system is given in the c.m. frame, with the first parton/particle going out in the $+z$ direction.

```
CALL PY3ENT(IP,KF1,KF2,KF3,PECM,X1,X3)
```

Purpose: to add three entries to the event record, i.e. either a 3-parton system or three separate particles.

`IP` : normally line number for the first parton/particle, with the other two in `IP+1` and `IP+2`. There are two exceptions.
If `IP=0`, lines 1, 2 and 3 are used and `PYEXEC` is called.
If `IP<0`, lines `-IP` through `-IP+2` are used, with status code `K(I,1)=3`, i.e. with special colour connection information, so that a parton shower can be generated by a `PYSHOW` call, followed by a `PYEXEC` call, if so desired (only relevant for partons).

`KF1, KF2, KF3`: flavour codes for the three partons/particles.

`PECM` : ($E_{\rm cm}$) the total energy of the system.

`X1, X3` : $x_i = 2E_i/E_{\rm cm}$, i.e. twice the energy fraction taken by the i'th parton. Thus $x_2 = 2 - x_1 - x_3$, and need not be given. Note that not all combinations of x_i are inside the physically allowed region.

Remarks : the system is given in the c.m. frame, in the xz-plane, with the first parton going out in the $+z$ direction and the third one having $p_x > 0$. A system must be given in the order of colour flow, so $\mathrm{qg\overline{q}}$ and $\mathrm{\overline{q}gq}$ are allowed but $\mathrm{q\overline{q}g}$ not. Thus $x1$ and x_3 come to correspond to what is normally called x_1 and x_2, i.e. the scaled q and $\overline{\mathrm{q}}$ energies.

```
CALL PY4ENT(IP,KF1,KF2,KF3,KF4,PECM,X1,X2,X4,X12,X14)
```

Purpose: to add four entries to the event record, i.e. either a 4-parton system or four separate particles (or, for $\mathrm{q\overline{q}q'\overline{q}'}$ events, two 2-parton systems).

`IP` : normally line number for the first parton/particle, with the other three in lines `IP+1`, `IP+2` and `IP+3`. There are two exceptions.
If `IP=0`, lines 1, 2, 3 and 4 are used and `PYEXEC` is called.
If `IP<0`, lines `-IP` through `-IP+3` are used, with status code `K(I,1)=3`, i.e. with special colour connection information, so that a parton shower can be generated by a `PYSHOW` call, followed by a `PYEXEC` call, if so desired (only relevant for partons).

`KF1,KF2,KF3,KF4` : flavour codes for the four partons/particles.

`PECM` : ($= E_{\rm cm}$) the total energy of the system.

`X1,X2,X4` : $x_i = 2E_i/E_{\rm cm}$, i.e. twice the energy fraction taken by the i'th parton. Thus $x_3 = 2 - x_1 - x_2 - x_4$, and need not be given.

`X12,X14` : $x_{ij} = 2p_ip_j/E_{\rm cm}^2$, i.e. twice the four-vector product of the momenta for partons i and j, properly normalized. With the masses known, other x_{ij} may be constructed from the x_i and x_{ij} given. Note that not all combinations of x_i and x_{ij} are inside the physically allowed region.

Remarks: the system is given in the c.m. frame, with the first parton going out in the $+z$ direction and the fourth parton lying in the xz-plane with $p_x > 0$. The second parton will have $p_y > 0$ and $p_y < 0$ with equal probability, with the third parton balancing this p_y (this corresponds to a random choice between the two possible stereoisomers). A system must be given in the order of colour flow, e.g. $\mathrm{qgg\overline{q}}$ and $\mathrm{q\overline{q}'q'\overline{q}}$.

```
CALL PYJOIN(NJOIN,IJOIN)
```

Purpose: to connect a number of previously defined partons into a string configuration. Initially the partons must be given with status codes `K(I,1)=` 1, 2 or 3. Afterwards the partons all have status code 3, i.e. are given with full colour-flow information. Compared to the normal way of defining a parton system, the partons need therefore not appear in the same sequence in the event record as they are assumed to do along

the string. It is also possible to call PYSHOW for all or some of the entries making up the string formed by PYJOIN.

NJOIN: the number of entries that are to be joined by one string.

IJOIN: an one-dimensional array, of size at least NJOIN. The NJOIN first numbers are the positions of the partons that are to be joined, given in the order the partons are assumed to appear along the string. If the system consists entirely of gluons, the string is closed by connecting back the last to the first entry.

Remarks: only one string (i.e. one colour singlet) may be defined per call, but one is at liberty to use any number of PYJOIN calls for a given event. The program will check that the parton configuration specified makes sense, and not take any action unless it does. Note, however, that an initially sensible parton configuration may become nonsensical, if only some of the partons are reconnected, while the others are left unchanged.

```
CALL PYGIVE(CHIN)
```

Purpose: to set the value of any variable residing in the commmonblocks PYJETS, PYDAT1, PYDAT2, PYDAT3, PYDAT4, PYDATR, PYSUBS, PYPARS, PYINT1, PYINT2, PYINT3, PYINT4, PYINT5, PYINT6, PYINT7, PYINT8, PYMSSM or PYMSRV. This is done in a more controlled fashion than by directly including the common blocks in your program, in that array bounds are checked and the old and new values for the variable changed are written to the output for reference. An example how PYGIVE can be used to parse input from a file is given in subsection 3.6.

CHIN : character expression of length at most 100 characters, with requests for variables to be changed, stored in the form
variable1=value1;variable2=value2;variable3=value3....
Note that an arbitrary number of instructions can be stored in one call if separated by semicolons, and that blanks may be included anyplace. An exclamation mark is recognized as the beginning of a comment, which is not to be processed. Normal parsing is resumed at the next semicolon (if any remain). An example would be
CALL PYGIVE('MSEL=16!Higgs production;PMAS(25,1)=115.!h0 mass')
The variable$_i$ may be any single variable in the PYTHIA common blocks, and the value$_i$ must be of the correct integer, real or character (without extra quotes) type. Array indices and values must be given explicitly, i.e. cannot be variables in their own right. The exception is that the first index can be preceded by a C, signifying that the index should be translated from normal KF to compressed KC code with a PYCOMP call; this is allowed for the KCHG, PMAS, MDCY, CHAF and MWID arrays.
If a value$_i$ is omitted, i.e. with the construction variable=, the current value is written to the output, but the variable itself is not changed.
The writing of info can be switched off by MSTU(13)=0.

Remark : The checks on array bounds are hardwired into this routine. Therefore, if you change array dimensions and MSTU(3), MSTU(6) and/or MSTU(7), as allowed by other considerations, these changes will not be known to PYGIVE. Normally this should not be a problem, however.

14.2 The Physics Routines

Once the initial parton/particle configuration has been specified and default parameter values changed, if so desired, only a PYEXEC call is necessary to simulate the whole fragmentation and decay chain. Therefore a normal user will not directly see any of the other routines in this section. Some of them could be called directly, but the danger of faulty usage is then non-negligible.

The PYTAUD routine provides an optional interface to an external τ decay library, where polarization effects could be included. It is up to you to write the appropriate calls, as explained at the end of this section.

```
CALL PYEXEC
```

Purpose: to administrate the fragmentation and decay chain. PYEXEC may be called several times, but only entries which have not yet been treated (more precisely, which have $1 \leq$K(I,1)≤ 10) can be affected by further calls. This may apply if more partons/particles have been added by you, or if particles previously considered stable are now allowed to decay. The actions that will be taken during a PYEXEC call can be tailored extensively via the PYDAT1–PYDAT3 common blocks, in particular by setting the MSTJ values suitably.

SUBROUTINE PYPREP(IP) : to rearrange parton shower end products (marked with K(I,1)=3) sequentially along strings; also to (optionally) allow small parton systems to collapse into two particles or one only, in the latter case with energy and momentum to be shuffled elsewhere in the event; also to perform checks that e.g. flavours of colour-singlet systems make sense.

SUBROUTINE PYSTRF(IP) : to generate the fragmentation of an arbitrary colour-singlet parton system according to the Lund string fragmentation model. One of the absolutely central routines of PYTHIA.

SUBROUTINE PYINDF(IP) : to handle the fragmentation of a parton system according to independent fragmentation models, and implement energy, momentum and flavour conservation, if so desired. Also the fragmentation of a single parton, not belonging to a parton system, is considered here (this is of course physical nonsense, but may sometimes be convenient for specific tasks).

SUBROUTINE PYDECY(IP) : to perform a particle decay, according to known branching ratios or different kinds of models, depending on our level of knowledge. Various matrix elements are included for specific processes.

SUBROUTINE PYKFDI(KFL1,KFL2,KFL3,KF) : to generate a new quark or diquark flavour and to combine it with an existing flavour to give a hadron.

- KFL1: incoming flavour.
- KFL2: extra incoming flavour, e.g. for formation of final particle, where the flavours are completely specified. Is normally 0.
- KFL3: newly created flavour; is 0 if KFL2 is non-zero.
- KF: produced hadron. Is 0 if something went wrong (e.g. inconsistent combination of incoming flavours).

SUBROUTINE PYPTDI(KFL,PX,PY) : to give transverse momentum, e.g. for a $q\bar{q}$ pair created in the colour field, according to independent Gaussian distributions in p_x and p_y.

SUBROUTINE PYZDIS(KFL1,KFL3,PR,Z) : to generate the longitudinal scaling variable z in jet fragmentation, either according to the Lund symmetric fragmentation function, or according to a choice of other shapes.

SUBROUTINE PYBOEI : to include Bose–Einstein effects according to a simple parameterization. By default, this routine is not called. If called from PYEXEC, this is done after the decay of short-lived resonances, but before the decay of long-lived ones. This means the routine should never be called directly by you, nor would effects be correctly simulated if decays are switched off. See MSTJ(51) - MSTJ(57) for switches of the routine.

FUNCTION PYMASS(KF) : to give the mass for a parton/particle.

SUBROUTINE PYNAME(KF,CHAU) : to give the parton/particle name (as a string of type CHARACTER CHAU*16). The name is read out from the CHAF array.

FUNCTION PYCHGE(KF) : to give three times the charge for a parton/particle. The value is

read out from the KCHG(KC,1) array.

FUNCTION PYCOMP(KF) : to give the compressed parton/particle code KC for a given KF code, as required to find entry into mass and decay data tables. Also checks whether the given KF code is actually an allowed one (i.e. known by the program), and returns 0 if not. Note that KF may be positive or negative, while the resulting KC code is never negative.

Internally PYCOMP uses a binary search in a table, with KF codes arranged in increasing order, based on the KCHG(KC,4) array. This table is constructed the first time PYCOMP is called, at which time MSTU(20) is set to 1. In case of a user change of the KCHG(KC,4) array one should reset MSTU(20)=0 to force a re-initialization at the next PYCOMP call (this is automatically done in PYUPDA calls). To speed up execution, the latest (KF,KC) pair is kept in memory and checked before the standard binary search.

SUBROUTINE PYERRM(MERR,MESSAG) : to keep track of the number of errors and warnings encountered, write out information on them, and abort the program in case of too many errors.

FUNCTION PYANGL(X,Y) : to calculate the angle from the x and y coordinates.

SUBROUTINE PYLOGO : to write a title page for the PYTHIA programs. Called by PYLIST(0).

SUBROUTINE PYTIME(IDATI) : to give the date and time, for use in PYLOGO and elsewhere. Since Fortran 77 does not contain a standard way of obtaining this information, the routine is dummy, to be replaced by you. Some commented-out examples are given, e.g. for Fortran 90 or the GNU Linux libU77. The output is given in an integer array ITIME(6), with components year, month, day, hour, minute and second. If there should be no such information available on a system, it is acceptable to put all the numbers above to 0.

```
CALL PYTAUD(ITAU,IORIG,KFORIG,NDECAY)
```

Purpose: to act as an interface between the standard decay routine PYDECY and a user-supplied τ lepton decay library such as TAUOLA [Jad91]. The latter library would normally know how to handle polarized τ's, given the τ helicity as input, so one task of the interface routine is to construct the τ polarization/helicity from the information available. Input to the routine (from PYDECY) is provided in the first three arguments, while the last argument and some event record information have to be set before return. To use this facility you have to set the switch MSTJ(28), include your own interface routine PYTAUD and see to it that the dummy routine PYTAUD is not linked. The dummy routine is there only to avoid unresolved external references when no user-supplied interface is linked.

ITAU : line number in the event record where the τ is stored. The four-momentum of this τ has first been boosted back to the rest frame of the decaying mother and thereafter rotated to move out along the $+z$ axis. It would have been possible to also perform a final boost to the rest frame of the τ itself, but this has been avoided so as not to suppress the kinematics aspect of close-to-threshold production (e.g. in B decays) vs. high-energy production (e.g. in real W decays). The choice of frame should help the calculation of the helicity configuration. After the PYTAUD call the τ and its decay products will automatically be rotated and boosted back. However, seemingly, the event record does not conserve momentum at this intermediate stage.

IORIG : line number where the mother particle to the τ is stored. Is 0 if the mother is not stored. This does not have to mean the mother is unknown. For instance, in semileptonic B decays the mother is a $W^{\pm}$ with known four-momentum $p_W = p_\tau + p_{\nu_\tau}$, but there is no W line in the event record. When several copies of the mother is stored (e.g. one in the documentation section of the event record and one in the main sec-

tion), IORIG points to the last. If a branchings like $\tau \to \tau\gamma$ occurs, the 'grandmother' is given, i.e. the mother of the direct τ before branching.

KFORIG : flavour code for the mother particle. Is 0 if the mother is unknown. The mother would typically be a resonance such as γ^*/Z^0 (23), $W^\pm$ ($\pm$24), h^0 (25), or $H^\pm$ ($\pm$37). Often the helicity choice would be clear just by the knowledge of this mother species, e.g., $W^\pm$ vs. $H^\pm$. However, sometimes further complications may exist. For instance, the KF code 23 represents a mixture of γ^* and Z^0; a knowledge of the mother mass (in P(IORIG,5)) would here be required to make the choice of helicities. Further, a $W^\pm$ or Z^0 may either be (predominantly) transverse or longitudinal, depending on the production process under study.

NDECAY : the number of decay products of the τ; to be given by the user routine. You must also store the KF flavour codes of those decay products in the positions K(I,2), N+1$\leq$I$\leq$N+NDECAY, of the event record. The corresponding five-momentum (momentum, energy and mass) should be stored in the associated P(I,J) positions, 1$\leq$J$\leq$5. The four-momenta are expected to add up to the four-momentum of the τ in position ITAU. You should not change the N value or any of the other K or V values (neither for the τ nor for its decay products) since this is automatically done in PYDECY.

14.3 The General Switches and Parameters

The common block PYDAT1 contains the main switches and parameters for the fragmentation and decay treatment, but also for some other aspects. Here one may control in detail what the program is to do, if the default mode of operation is not satisfactory.

```
COMMON/PYDAT1/MSTU(200),PARU(200),MSTJ(200),PARJ(200)
```

Purpose: to give access to a number of status codes and parameters which regulate the performance of the program as a whole. Here MSTU and PARU are related to utility functions, as well as a few parameters of the Standard Model, while MSTJ and PARJ affect the underlying physics assumptions. Some of the variables in PYDAT1 are described elsewhere, and are therefore here only reproduced as references to the relevant sections. This in particular applies to many coupling constants, which are found in section 9.4, and switches of the older dedicated e^+e^- machinery, section 6.3.

MSTU(1) - MSTU(3) : variables used by the event study routines, section 15.1.

MSTU(4) : (D=4000) number of lines available in the common block PYJETS. Should always be changed if the dimensions of the K and P arrays are changed by you, but should otherwise never be touched. Maximum allowed value is 10000, unless MSTU(5) is also changed.

MSTU(5) : (D=10000) is used in building up the special colour-flow information stored in K(I,4) and K(I,5) for K(I,3)= 3, 13 or 14. The generic form for j= 4 or 5 is
K(I,j)$= 2\times$MSTU(5)$^2\times$MCFR+MSTU(5)$^2\times$MCTO+MSTU(5)$\times$ICFR+ICTO,
with notation as in section 5.2. One should always have MSTU(5)$\geq$MSTU(4). On a 32 bit machine, values MSTU(5)$>$ 20000 may lead to overflow problems, and should be avoided.

MSTU(6) : (D=500) number of KC codes available in the KCHG, PMAS, MDCY, and CHAF arrays; should be changed if these dimensions are changed.

MSTU(7) : (D=8000) number of decay channels available in the MDME, BRAT and KFDP arrays; should be changed if these dimensions are changed.

MSTU(10) : (D=2) use of parton/particle masses in filling routines (PY1ENT, PY2ENT, PY3ENT, PY4ENT).

= 0 : assume the mass to be zero.
= 1 : keep the mass value stored in P(I,5), whatever it is. (This may be used e.g. to describe kinematics with off-mass-shell partons).
= 2 : find masses according to mass tables as usual.

MSTU(11) - MSTU(12) : variables used by the event study routines, section 15.1.

MSTU(13) : (D=1) writing of information on variable values changed by a PYGIVE call.
= 0 : no information is provided.
= 1 : information is written to standard output.

MSTU(14) : variable used by the event study routines, section 15.1.

MSTU(15) : (D=0) decides how PYLIST shows empty lines, which are interspersed among ordinary particles in the event record.
= 0 : do not print lines with K(I,1) $\leq$ 0.
= 1 : do not print lines with K(I,1) $<$ 0.
= 2 : print all lines.

MSTU(16) : (D=1) choice of mother pointers for the particles produced by a fragmenting parton system.
= 1 : all primary particles of a system point to a line with KF = 92 or 93, for string or independent fragmentation, respectively, or to a line with KF = 91 if a parton system has so small a mass that it is forced to decay into one or two particles. The two (or more) shower initiators of a showering parton system point to a line with KF = 94. The entries with KF = 91–94 in their turn point back to the predecessor partons, so that the KF = 91–94 entries form a part of the event history proper.
= 2 : although the lines with KF = 91–94 are present, and contain the correct mother and daughter pointers, they are not part of the event history proper, in that particles produced in string fragmentation point directly to either of the two endpoint partons of the string (depending on the side they were generated from), particles produced in independent fragmentation point to the respective parton they were generated from, particles in small mass systems point to either endpoint parton, and shower initiators point to the original on-mass-shell counterparts. Also the daughter pointers bypass the KF = 91–94 entries. In independent fragmentation, a parton need not produce any particles at all, and then have daughter pointers 0.

Note : MSTU(16) should not be changed between the generation of an event and the translation of this event record with a PYHEPC call, since this may give an erroneous translation of the event history.

MSTU(17) : (D=0) storage option for MSTU(90) and associated information on z values for heavy-flavour production.
= 0 : MSTU(90) is reset to zero at each PYEXEC call. This is the appropriate course if PYEXEC is only called once per event, as is normally the case when you do not yourself call PYEXEC.
= 1 : you have to reset MSTU(90) to zero yourself before each new event. This is the appropriate course if several PYEXEC calls may appear for one event, i.e. if you call PYEXEC directly.

MSTU(19) : (D=0) advisory warning for unphysical flavour setups in PY2ENT, PY3ENT or PY4ENT calls.
= 0 : yes.
= 1 : no; MSTU(19) is reset to 0 in such a call.

MSTU(20) : (D=0) flag for the initialization status of the PYCOMP routine. A value 0 indicates that tables should be (re)initialized, after which it is set 1. In case you change the KCHG(KC,4) array you should reset MSTU(20)=0 to force a re-initialization at the next PYCOMP call.

MSTU(21) : (D=2) check on possible errors during program execution. Obviously no guaran-

tee is given that all errors will be caught, but some of the most trivial user-caused errors may be found.

= 0 : errors do not cause any immediate action, rather the program will try to cope, which may mean e.g. that it runs into an infinite loop.

= 1 : parton/particle configurations are checked for possible errors. In case of problem, an exit is made from the misbehaving subprogram, but the generation of the event is continued from there on. For the first MSTU(22) errors a message is printed; after that no messages appear.

= 2 : parton/particle configurations are checked for possible errors. In case of problem, an exit is made from the misbehaving subprogram, and subsequently from PYEXEC. You may then choose to correct the error, and continue the execution by another PYEXEC call. For the first MSTU(22) errors a message is printed, after that the last event is printed and execution is stopped.

MSTU(22) : (D=10) maximum number of errors that are printed.

MSTU(23) : (I) count of number of errors experienced to date.

MSTU(24) : (R) type of latest error experienced; reason that event was not generated in full. Is reset at each PYEXEC call.

= 0 : no error experienced.

= 1 : program has reached end of or is writing outside PYJETS memory.

= 2 : unknown flavour code or unphysical combination of codes; may also be caused by erroneous string connection information.

= 3 : energy or mass too small or unphysical kinematical variable setup.

= 4 : program is caught in an infinite loop.

= 5 : momentum, energy or charge was not conserved (even allowing for machine precision errors, see PARU(11)); is evaluated only after event has been generated in full, and does not apply when independent fragmentation without momentum conservation was used.

= 6 : error call from outside the fragmentation/decay package (e.g. the e^+e^- routines).

= 7 : inconsistent particle data input in PYUPDA (MUPDA = 2,3) or other PYUPDA-related problem.

= 8 : problems in more peripheral service routines.

= 9 : various other problems.

MSTU(25) : (D=1) printing of warning messages.

= 0 : no warnings are written.

= 1 : first MSTU(26) warnings are printed, thereafter no warnings appear.

MSTU(26) : (D=10) maximum number of warnings that are printed.

MSTU(27) : (I) count of number of warnings experienced to date.

MSTU(28) : (R) type of latest warning given, with codes parallelling those for MSTU(24), but of a less serious nature.

MSTU(31) : (I) number of PYEXEC calls in present run.

MSTU(32) - MSTU(33) : variables used by the event study routines, section 15.1.

MSTU(41) - MSTU(63) : switches for event-analysis routines, see section 15.5.

MSTU(70) - MSTU(80) : variables used by the event study routines, section 15.1.

MSTU(90) : number of heavy-flavour hadrons (i.e. hadrons containing charm or bottom) produced in the fragmentation stage of the current event, for which the positions in the event record are stored in MSTU(91) - MSTU(98) and the z values in the fragmentation in PARU(91) - PARU(98). At most eight values will be stored (normally this is no problem). No z values can be stored for those heavy hadrons produced when a string has so small mass that it collapses to one or two particles, nor for those produced as one of the final two particles in the fragmentation of a string. If MSTU(17)=1, MSTU(90) should be reset to zero by you before each new event, else this is done automatically.

`MSTU(91) - MSTU(98)` : the first `MSTU(90)` positions will be filled with the line numbers of the heavy-flavour hadrons produced in the current event. See `MSTU(90)` for additional comments. Note that the information is corrupted by calls to `PYEDIT` with options 0–5 and 21–23; calls with options 11–15 work, however.

`MSTU(101) - MSTU(118)` : switches related to couplings, see section 9.4.

`MSTU(121) - MSTU(125)` : internally used in the advanced popcorn code, see subsection 14.3.1.

`MSTU(131) - MSTU(140)` : internally used in the advanced popcorn code, see subsection 14.3.1.

`MSTU(161), MSTU(162)` : information used by event-analysis routines, see section 15.5.

`PARU(1)` : (R) $\pi \approx 3.141592653589793$.

`PARU(2)` : (R) $2\pi \approx 6.283185307179586$.

`PARU(3)` : (D=0.197327) conversion factor for $\mathrm{GeV}^{-1} \to \mathrm{fm}$ or $\mathrm{fm}^{-1} \to \mathrm{GeV}$.

`PARU(4)` : (D=5.06773) conversion factor for $\mathrm{fm} \to \mathrm{GeV}^{-1}$ or $\mathrm{GeV} \to \mathrm{fm}^{-1}$.

`PARU(5)` : (D=0.389380) conversion factor for $\mathrm{GeV}^{-2} \to \mathrm{mb}$ or $\mathrm{mb}^{-1} \to \mathrm{GeV}^2$.

`PARU(6)` : (D=2.56819) conversion factor for $\mathrm{mb} \to \mathrm{GeV}^{-2}$ or $\mathrm{GeV}^2 \to \mathrm{mb}^{-1}$.

`PARU(11)` : (D=0.001) relative error, i.e. non-conservation of momentum and energy divided by total energy, that may be attributable to machine precision problems before a physics error is suspected (see `MSTU(24)=5`).

`PARU(12)` : (D=0.09 GeV^2) effective cut-off in squared mass, below which partons may be recombined to simplify (machine precision limited) kinematics of string fragmentation. (Default chosen to be of the order of a light quark mass, or half a typical light meson mass.)

`PARU(13)` : (D=0.01) effective angular cut-off in radians for recombination of partons, used in conjunction with `PARU(12)`.

`PARU(21)` : (I) contains the total energy W of all first generation partons/particles after a `PYEXEC` call; to be used by the `PYP` function for `I>0`, `J=` 20–25.

`PARU(41) - PARU(63)` : parameters for event-analysis routines, see section 15.5.

`PARU(91) - PARU(98)` : the first `MSTU(90)` positions will be filled with the fragmentation z values used internally in the generation of heavy-flavour hadrons — how these are translated into the actual energies and momenta of the observed hadrons is a complicated function of the string configuration. The particle with z value stored in `PARU(i)` is to be found in line `MSTU(i)` of the event record. See `MSTU(90)` and `MSTU(91) - MSTU(98)` for additional comments.

`PARU(101) - PARU(195)` : various coupling constants and parameters related to couplings, see section 9.4.

`MSTJ(1)` : (D=1) choice of fragmentation scheme.

- `= 0` : no jet fragmentation at all.
- `= 1` : string fragmentation according to the Lund model.
- `= 2` : independent fragmentation, according to specification in `MSTJ(2)` and `MSTJ(3)`.

`MSTJ(2)` : (D=3) gluon jet fragmentation scheme in independent fragmentation.

- `= 1` : a gluon is assumed to fragment like a random d, u or s quark or antiquark.
- `= 2` : as `=1`, but longitudinal (see `PARJ(43)`, `PARJ(44)` and `PARJ(59)`) and transverse (see `PARJ(22)`) momentum properties of quark or antiquark substituting for gluon may be separately specified.
- `= 3` : a gluon is assumed to fragment like a pair of a d, u or s quark and its antiquark, sharing the gluon energy according to the Altarelli-Parisi splitting function.
- `= 4` : as `=3`, but longitudinal (see `PARJ(43)`, `PARJ(44)` and `PARJ(59)`) and transverse (see `PARJ(22)`) momentum properties of quark and antiquark substituting for gluon may be separately specified.

`MSTJ(3)` : (D=0) energy, momentum and flavour conservation options in independent frag-

mentation. Whenever momentum conservation is described below, energy and flavour conservation is also implicitly assumed.

`= 0 :` no explicit conservation of any kind.

`= 1 :` particles share momentum imbalance compensation according to their energy (roughly equivalent to boosting event to c.m. frame). This is similar to the approach in the Ali et al. program [Ali80].

`= 2 :` particles share momentum imbalance compensation according to their longitudinal mass with respect to the imbalance direction.

`= 3 :` particles share momentum imbalance compensation equally.

`= 4 :` transverse momenta are compensated separately within each jet, longitudinal momenta are rescaled so that ratio of final jet to initial parton momentum is the same for all the jets of the event. This is similar to the approach in the Hoyer et al. program [Hoy79].

`= 5 :` only flavour is explicitly conserved.

`= 6 - 10 :` as `=1 - 5`, except that above several colour singlet systems that followed immediately after each other in the event listing (e.g. $q\bar{q}q\bar{q}$) were treated as one single system, whereas here they are treated as separate systems.

`= -1 :` independent fragmentation, where also particles moving backwards with respect to the jet direction are kept, and thus the amount of energy and momentum mismatch may be large.

`MSTJ(11) :` (D=4) choice of longitudinal fragmentation function, i.e. how large a fraction of the energy available a newly-created hadron takes.

`= 1 :` the Lund symmetric fragmentation function, see `PARJ(41) - PARJ(45)`.

`= 2 :` choice of some different forms for each flavour separately, see `PARJ(51) - PARJ(59)`.

`= 3 :` hybrid scheme, where light flavours are treated with symmetric Lund (`=1`), but charm and heavier can be separately chosen, e.g. according to the Peterson/SLAC function (`=2`).

`= 4 :` the Lund symmetric fragmentation function (`=1`), for heavy endpoint quarks modified according to the Bowler (Artru–Mennessier, Morris) space–time picture of string evolution, see `PARJ(46)`.

`= 5 :` as `=4`, but with possibility to interpolate between Bowler and Lund separately for c and b; see `PARJ(46)` and `PARJ(47)`.

`MSTJ(12) :` (D=2) choice of baryon production model.

`= 0 :` no baryon-antibaryon pair production at all; initial diquark treated as a unit.

`= 1 :` diquark-antidiquark pair production allowed; diquark treated as a unit.

`= 2 :` diquark-antidiquark pair production allowed, with possibility for diquark to be split according to the 'popcorn' scheme.

`= 3 :` as `=2`, but additionally the production of first rank baryons may be suppressed by a factor `PARJ(19)`.

`= 4 :` as `=2`, but diquark vertices suffer an extra suppression of the form $1-\exp(\rho\Gamma)$, where $\rho \approx 0.7\mathrm{GeV}^{-2}$ is stored in `PARF(192)`.

`= 5 :` Advanced version of the popcorn model. Independent of `PARJ(3-7)`. Instead depending on `PARJ(8-10)`. When using this option `PARJ(1)` needs to enhanced by approx. a factor 2 (i.e. it losses a bit of its normal meaning), and `PARJ(18)` is suggested to be set to 0.19. See section 14.3.1 for further details.

`MSTJ(13) :` (D=0) generation of transverse momentum for endpoint quark(s) of single quark jet or $q\bar{q}$ jet system (in multijet events no endpoint transverse momentum is ever allowed for).

`= 0 :` no transverse momentum for endpoint quarks.

`= 1 :` endpoint quarks obtain transverse momenta like ordinary $q\bar{q}$ pairs produced in the field (see `PARJ(21)`); for 2-jet systems the endpoints obtain balancing transverse momenta.

`MSTJ(14)` : (D=1) treatment of a colour-singlet parton system with a low invariant mass.

- `= 0` : no precautions are taken, meaning that problems may occur in `PYSTRF` (or `PYINDF`) later on. Warning messages are issued when low masses are encountered, however, or when the flavour or colour configuration appears to be unphysical.
- `= 1` : small parton systems are allowed to collapse into two particles or, failing that, one single particle. Normally all small systems are treated this way, starting with the smallest one, but some systems would require more work and are left untreated; they include diquark-antidiquark pairs below the two-particle threshold. See further `MSTJ(16)` and `MSTJ(17)`.
- `= -1` : special option for `PYPREP` calls, where no precautions are taken (as for `=0`), but, in addition, no checks are made on the presence of small-mass systems or unphysical flavour or colour configurations; i.e. `PYPREP` only rearranges colour strings.

`MSTJ(15)` : (D=0) production probability for new flavours.

- `= 0` : according to standard Lund parameterization, as given by `PARJ(1) - PARJ(20)`.
- `= 1` : according to probabilities stored in `PARF(201) - PARF(1960)`; note that no default values exist here, i.e. `PARF` must be set by you. The `MSTJ(12)` switch can still be used to set baryon production mode, with the modification that `MSTJ(12)=2` here allows an arbitrary number of mesons to be produced between a baryon and an antibaryon (since the probability for diquark $\to$ meson $+$ new diquark is assumed independent of prehistory).

`MSTJ(16)` : (D=2) mode of cluster treatment (where a cluster is a low-mass string that can fragment to two particles at the most).

- `= 0` : old scheme. Cluster decays (to two hadrons) are isotropic. In cluster collapses (to one hadron), energy-momentum compensation is to/from the parton or hadron furthest away in mass.
- `= 1` : intermediate scheme. Cluster decays are anisotropic in a way that is intended to mimic the Gaussian $p_\perp$ suppression and string 'area law' of suppressed rapidity orderings of ordinary string fragmentation. In cluster collapses, energy-momentum compensation is to/from the string piece most closely moving in the same direction as the cluster. Excess energy is put as an extra gluon on this string piece, while a deficit is taken from both endpoints of this string piece as a common fraction of their original momentum.
- `= 2` : new default scheme. Essentially as `=1` above, except that an energy deficit is preferentially taken from the endpoint of the string piece that is moving closest in direction to the cluster.

`MSTJ(17)` : (D=2) number of attempts made to find two hadrons that have a combined mass below the cluster mass, and thus allow a cluster to decay to two hadrons rather than collapse to one. Thus the larger `MSTJ(17)`, the smaller the fraction of collapses. At least one attempt is always made, and this was the old default behaviour.

`MSTJ(21)` : (D=2) form of particle decays.

- `= 0` : all particle decays are inhibited.
- `= 1` : a particle declared unstable in the `MDCY` vector, and with decay channels defined, may decay within the region given by `MSTJ(22)`. A particle may decay into partons, which then fragment further according to the `MSTJ(1)` value.
- `= 2` : as `=1`, except that a $q\overline{q}$ parton system produced in a decay (e.g. of a B meson) is always allowed to fragment according to string fragmentation, rather than according to the `MSTJ(1)` value (this means that momentum, energy and charge are conserved in the decay).

`MSTJ(22)` : (D=1) cut-off on decay length for a particle that is allowed to decay according to `MSTJ(21)` and the `MDCY` value.

= 1 : a particle declared unstable is also forced to decay.

= 2 : a particle is decayed only if its average proper lifetime is smaller than PARJ(71).

= 3 : a particle is decayed only if the decay vertex is within a distance PARJ(72) of the origin.

= 4 : a particle is decayed only if the decay vertex is within a cylindrical volume with radius PARJ(73) in the *xy*-plane and extent to ±PARJ(74) in the *z* direction.

MSTJ(23) : (D=1) possibility of having a shower evolving from a $q\overline{q}$ pair created as decay products. This switch only applies to decays handled by PYDECY rather than PYRESD, and so is of less relevance today.

= 0 : never.

= 1 : whenever the decay channel matrix-element code is MDME(IDC,2)= 4, 32, 33, 44 or 46, the two first decay products (if they are partons) are allowed to shower, like a colour-singlet subsystem, with maximum virtuality given by the invariant mass of the pair.

MSTJ(24) : (D=2) particle masses.

= 0 : discrete mass values are used.

= 1 : particles registered as having a mass width in the PMAS vector are given a mass according to a truncated Breit–Wigner shape, linear in *m*, eq. (240).

= 2 : as =1, but gauge bosons (actually all particles with $|\mathtt{KF}| \leq 100$) are distributed according to a Breit–Wigner quadratic in *m*, as obtained from propagators.

= 3 : as =1, but Breit–Wigner shape is always quadratic in *m*, eq. (241).

MSTJ(26) : (D=2) inclusion of B–$\overline{\mathrm{B}}$ mixing in decays.

= 0 : no.

= 1 : yes, with mixing parameters given by PARJ(76) and PARJ(77). Mixing decays are not specially marked.

= 2 : yes, as =1, but a B ($\overline{\mathrm{B}}$) that decays as a $\overline{\mathrm{B}}$ (B) is marked as K(I,1)=12 rather than the normal K(I,1)=11.

MSTJ(28) : (D=0) call to an external τ decay library like TAUOLA. For this option to be meaningful, it is up to you to write the appropriate interface and include that in the routine PYTAUD, as explained in section 14.2.

= 0 : not done, i.e. the internal PYDECY treatment is used.

= 1 : done whenever the τ mother particle species can be identified, else the internal PYDECY treatment is used. Normally the mother particle should always be identified, but it is possible for you to remove event history information or to add extra τ's directly to the event record, and then the mother is not known.

= 2 : always done.

MSTJ(38) - MSTJ(50) : switches for time-like parton showers, see section 10.4.

MSTJ(51) : (D=0) inclusion of Bose–Einstein effects.

= 0 : no effects included.

= 1 : effects included according to an exponential parameterization $f_2(Q) = 1+\mathtt{PARJ(92)}\times\exp(-Q/\mathtt{PARJ(93)})$, where $f_2(Q)$ represents the ratio of particle production at Q with Bose–Einstein effects to that without, and the relative momentum Q is defined by $Q^2(p_1,p_2) = -(p_1-p_2)^2 = (p_1+p_2)^2 - 4m^2$. Particles with width broader than PARJ(91) are assumed to have time to decay before Bose–Einstein effects are to be considered.

= 2 : effects included according to a Gaussian parameterization $f_2(Q) = 1+\mathtt{PARJ(92)}\times\exp(-(Q/\mathtt{PARJ(93)})^2)$, with notation and comments as above.

MSTJ(52) : (D=3) number of particle species for which Bose–Einstein correlations are to be included, ranged along the chain π^+, π^-, π^0, K^+, K^-, K^0_L, K^0_S, η and η'. Default corresponds to including all pions (π^+, π^-, π^0), 7 to including all Kaons as well,

and 9 is maximum.

`MSTJ(53)` : (D=0) In $e^+e^- \to W^+W^-$, $e^+e^- \to Z^0Z^0$, or if `PARJ(94)`> 0 and there are several strings in the event, apply BE algorithm

- `= 0` : on all pion pairs.
- `= 1` : only on pairs were both pions come from the same W/Z/string.
- `= 2` : only on pairs were the pions come from different W/Z/strings.
- `= -2` : when calculating balancing shifts for pions from same W/Z/string, only consider pairs from this W/Z/string.

Note: if colour reconnections has occurred in an event, the distinction between pions coming from different W/Z's is lost.

`MSTJ(54)` : (D=2) Alternative local energy compensation. (Notation in brackets refer to the one used in [Lön95].)

- `= 0` : global energy compensation (BE_0).
- `= 1` : compensate with identical pairs by negative BE enhancement with a third of the radius (BE_3).
- `= 2` : ditto, but with the compensation constrained to vanish at $Q = 0$, by an additional $1 - \exp(-Q^2R^2/4)$ factor (BE_{32}).
- `= -1` : compensate with pair giving the smallest invariant mass (BE_m).
- `= -2` : compensate with pair giving the smallest string length (BE_λ).

`MSTJ(55)` : (D=0) Calculation of difference vector.

- `= 0` : in the lab frame.
- `= 1` : in the c.m. of the given pair.

`MSTJ(56)` : (D=0) In $e^+e^- \to W^+W^-$ or $e^+e^- \to Z^0Z^0$ include distance between W/Z's.

- `= 0` : radius is the same for all pairs.
- `= 1` : radius for pairs from different W/Z's is $R + \delta R_{WW}$ ($R + \delta R_{ZZ}$), where δR is the generated distance between the decay vertices. (When considering W or Z pairs with an energy well above threshold, this should give more realistic results.)

`MSTJ(57)` : (D=1) Penalty for shifting particles with close-by identical neighbours in local energy compensation, `MSTJ(54) < 0`.

- `= 0` : no penalty.
- `= 1` : penalty.

`MSTJ(91)` : (I) flag when generating gluon jet with options `MSTJ(2)=` 2 or 4 (then `=1`, else `=0`).

`MSTJ(92)` : (I) flag that a $q\bar{q}$ or gg pair or a ggg triplet created in `PYDECY` should be allowed to shower, is 0 if no pair or triplet, is the entry number of the first parton if a pair indeed exists, is the entry number of the first parton, with a $-$ sign, if a triplet indeed exists.

`MSTJ(93)` : (I) switch for `PYMASS` action. Is reset to 0 in `PYMASS` call.

- `= 0` : ordinary action.
- `= 1` : light (d, u, s, c, b) quark masses are taken from `PARF(101) - PARF(105)` rather than `PMAS(1,1) - PMAS(5,1)`. Diquark masses are given as sum of quark masses, without spin splitting term.
- `= 2` : as `=1`. Additionally the constant terms `PARF(121)` and `PARF(122)` are subtracted from quark and diquark masses, respectively.

`MSTJ(101) - MSTJ(121)` : switches for e^+e^- event generation, see section 6.3.

`PARJ(1)` : (D=0.10) is $\mathcal{P}(qq)/\mathcal{P}(q)$, the suppression of diquark-antidiquark pair production in the colour field, compared with quark–antiquark production.

`PARJ(2)` : (D=0.30) is $\mathcal{P}(s)/\mathcal{P}(u)$, the suppression of s quark pair production in the field compared with u or d pair production.

`PARJ(3)` : (D=0.4) is $(\mathcal{P}(us)/\mathcal{P}(ud))/(\mathcal{P}(s)/\mathcal{P}(d))$, the extra suppression of strange diquark production compared with the normal suppression of strange quarks.

PARJ(4) : (D=0.05) is $(1/3)\mathcal{P}(\mathrm{ud}_1)/\mathcal{P}(\mathrm{ud}_0)$, the suppression of spin 1 diquarks compared with spin 0 ones (excluding the factor 3 coming from spin counting).

PARJ(5) : (D=0.5) parameter determining relative occurrence of baryon production by $BM\overline{B}$ and by $B\overline{B}$ configurations in the simple popcorn baryon production model, roughly $\mathcal{P}(BM\overline{B})/(\mathcal{P}(B\overline{B})+\mathcal{P}(BM\overline{B})) =$ PARJ(5)/(0.5+PARJ(5)). This and subsequent baryon parameters are modified in the advanced popcorn scenario, see subsection 14.3.1.

PARJ(6) : (D=0.5) extra suppression for having a $\mathrm{s}\bar{\mathrm{s}}$ pair shared by the B and $\overline{B}$ of a $BM\overline{B}$ situation.

PARJ(7) : (D=0.5) extra suppression for having a strange meson M in a $BM\overline{B}$ configuration.

PARJ(8) - PARJ(10) : used in the advanced popcorn scenario, see subsection 14.3.1.

PARJ(11) - PARJ(17) : parameters that determine the spin of mesons when formed in fragmentation or decays.

- PARJ(11) : (D=0.5) is the probability that a light meson (containing u and d quarks only) has spin 1 (with 1-PARJ(11) the probability for spin 0).
- PARJ(12) : (D=0.6) is the probability that a strange meson has spin 1.
- PARJ(13) : (D=0.75) is the probability that a charm or heavier meson has spin 1.
- PARJ(14) : (D=0.) is the probability that a spin = 0 meson is produced with an orbital angular momentum 1, for a total spin = 1.
- PARJ(15) : (D=0.) is the probability that a spin = 1 meson is produced with an orbital angular momentum 1, for a total spin = 0.
- PARJ(16) : (D=0.) is the probability that a spin = 1 meson is produced with an orbital angular momentum 1, for a total spin = 1.
- PARJ(17) : (D=0.) is the probability that a spin = 1 meson is produced with an orbital angular momentum 1, for a total spin = 2.

Note : the end result of the numbers above is that, with i = 11, 12 or 13, depending on flavour content,

$$\mathcal{P}(S=0,L=0,J=0) = (1-\mathtt{PARJ(i)})\times(1-\mathtt{PARJ(14)}),$$
$$\mathcal{P}(S=0,L=1,J=1) = (1-\mathtt{PARJ(i)})\times\mathtt{PARJ(14)},$$
$$\mathcal{P}(S=1,L=0,J=1) = \mathtt{PARJ(i)}\times(1-\mathtt{PARJ(15)}-\mathtt{PARJ(16)}-\mathtt{PARJ(17)}),$$
$$\mathcal{P}(S=1,L=1,J=0) = \mathtt{PARJ(i)}\times\mathtt{PARJ(15)},$$
$$\mathcal{P}(S=1,L=1,J=1) = \mathtt{PARJ(i)}\times\mathtt{PARJ(16)},$$
$$\mathcal{P}(S=1,L=1,J=2) = \mathtt{PARJ(i)}\times\mathtt{PARJ(17)},$$

where S is the quark 'true' spin and J is the total spin, usually called the spin s of the meson.

PARJ(18) : (D=1.) is an extra suppression factor multiplying the ordinary **SU(6)** weight for spin 3/2 baryons, and hence a means to break **SU(6)** in addition to the dynamic breaking implied by PARJ(2), PARJ(3), PARJ(4), PARJ(6) and PARJ(7).

PARJ(19) : (D=1.) extra baryon suppression factor, which multiplies the ordinary diquark-antidiquark production probability for the breakup closest to the endpoint of a string, but leaves other breaks unaffected. Is only used for MSTJ(12)=3.

PARJ(21) : (D=0.36 GeV) corresponds to the width σ in the Gaussian p_x and p_y transverse momentum distributions for primary hadrons. See also PARJ(22) - PARJ(24).

PARJ(22) : (D=1.) relative increase in transverse momentum in a gluon jet generated with MSTJ(2)= 2 or 4.

PARJ(23), PARJ(24) : (D=0.01, 2.) a fraction PARJ(23) of the Gaussian transverse momentum distribution is taken to be a factor PARJ(24) larger than input in PARJ(21). This gives a simple parameterization of non-Gaussian tails to the Gaussian shape assumed above.

PARJ(25) : (D=1.) extra suppression factor for η production in fragmentation; if an η is rejected a new flavour pair is generated and a new hadron formed.

PARJ(26) : (D=0.4) extra suppression factor for η' production in fragmentation; if an η' is

rejected a new flavour pair is generated and a new hadron formed.

PARJ(31) : (D=0.1 GeV) gives the remaining W_+ below which the generation of a single jet is stopped. (It is chosen smaller than a pion mass, so that no hadrons moving in the forward direction are missed.)

PARJ(32) : (D=1. GeV) is, with quark masses added, used to define the minimum allowable energy of a colour-singlet parton system.

PARJ(33) - PARJ(34) : (D=0.8 GeV, 1.5 GeV) are, together with quark masses, used to define the remaining energy below which the fragmentation of a parton system is stopped and two final hadrons formed. PARJ(33) is normally used, except for MSTJ(11)=2, when PARJ(34) is used.

PARJ(36) : (D=2.) represents the dependence on the mass of the final quark pair for defining the stopping point of the fragmentation. Is strongly correlated to the choice of PARJ(33) - PARJ(35).

PARJ(37) : (D=0.2) relative width of the smearing of the stopping point energy.

PARJ(39) : (D=0.08 GeV^{-2}) refers to the probability for reverse rapidity ordering of the final two hadrons, according to eq. (226), for MSTJ(11)=2 (for other MSTJ(11) values PARJ(42) is used).

PARJ(41), PARJ(42) : (D=0.3, 0.58 GeV^{-2}) give the a and b parameters of the symmetric Lund fragmentation function for MSTJ(11)=1, 4 and 5 (and MSTJ(11)=3 for ordinary hadrons).

PARJ(43), PARJ(44) : (D=0.5, 0.9 GeV^{-2}) give the a and b parameters as above for the special case of a gluon jet generated with IF and MSTJ(2)= 2 or 4.

PARJ(45) : (D=0.5) the amount by which the effective a parameter in the Lund flavour dependent symmetric fragmentation function is assumed to be larger than the normal a when diquarks are produced. More specifically, referring to eq. (213), a_α =PARJ(41) when considering the fragmentation of a quark and = PARJ(41) + PARJ(45) for the fragmentation of a diquark, with corresponding expression for a_β depending on whether the newly created object is a quark or diquark (for an independent gluon jet generated with MSTJ(2)= 2 or 4, replace PARJ(41) by PARJ(43)). In the popcorn model, a meson created in between the baryon and antibaryon has $a_\alpha = a_\beta$ =PARJ(41) + PARJ(45).

PARJ(46), PARJ(47) : (D=2*1.) modification of the Lund symmetric fragmentation for heavy endpoint quarks according to the recipe by Bowler, available when MSTJ(11)= 4 or 5 is selected. The shape is given by eq. (216). If MSTJ(11)=4 then r_Q =PARJ(46) for both c and b, while if MSTJ(11)=5 then r_c =PARJ(46) and r_b =PARJ(47). PARJ(46) and PARJ(47) thus provide a possibility to interpolate between the 'pure' Bowler shape, $r = 1$, and the normal Lund one, $r = 0$. The additional modifications made in PARJ(43) - PARJ(45) are automatically taken into account, if necessary.

PARJ(51) - PARJ(55) : (D=3*0.77, -0.05, -0.005) give a choice of four possible ways to parameterize the fragmentation function for MSTJ(11)=2 (and MSTJ(11)=3 for charm and heavier). The fragmentation of each flavour KF may be chosen separately; for a diquark the flavour of the heaviest quark is used. With c =PARJ(50+KF), the parameterizations are:
$0 \le c \le 1$: Field-Feynman, $f(z) = 1 - c + 3c(1-z)^2$;
$-1 \le c < 0$: Peterson/SLAC, $f(z) = 1/(z(1-1/z-(-c)/(1-z))^2)$;
$c > 1$: power peaked at $z = 0$, $f(z) = (1-z)^{c-1}$;
$c < -1$: power peaked at $z = 1$, $f(z) = z^{-c-1}$.

PARJ(59) : (D=1.) replaces PARJ(51) - PARJ(53) for gluon jet generated with MSTJ(2)= 2 or 4.

PARJ(61) - PARJ(63) : (D=4.5, 0.7, 0.) parameterizes the energy dependence of the primary multiplicity distribution in phase-space decays. The former two correspond to c_1 and c_2 of eq. (249), while the latter allows a further additive term in the multi-

plicity specifically for onium decays.

`PARJ(64)` : (0.003 GeV) minimum kinetic energy in decays (safety margin for numerical precision errors). When violated, typically new masses would be selected if particles have a Breit-Wigner width, or a new decay channel where that is relevant.

`PARJ(65)` : (D=0.5 GeV) mass which, in addition to the spectator quark or diquark mass, is not assumed to partake in the weak decay of a heavy quark in a hadron. This parameter was mainly intended for top decay and is currently not in use.

`PARJ(66)` : (D=0.5) relative probability that colour is rearranged when two singlets are to be formed from decay products. Only applies for `MDME(IDC,2)=` 11–30, i.e. low-mass phase-space decays.

`PARJ(71)` : (D=10 mm) maximum average proper lifetime $c\tau$ for particles allowed to decay in the `MSTJ(22)=2` option. With the default value, K^0_S, Λ, Σ^-, Σ^+, Ξ^-, Ξ^0 and Ω^- are stable (in addition to those normally taken to be stable), but charm and bottom do still decay.

`PARJ(72)` : (D=1000 mm) maximum distance from the origin at which a decay is allowed to take place in the `MSTJ(22)=3` option.

`PARJ(73)` : (D=100 mm) maximum cylindrical distance $\rho = \sqrt{x^2+y^2}$ from the origin at which a decay is allowed to take place in the `MSTJ(22)=4` option.

`PARJ(74)` : (D=1000 mm) maximum z distance from the origin at which a decay is allowed to take place in the `MSTJ(22)=4` option.

`PARJ(76)` : (D=0.7) mixing parameter $x_d = \Delta M/\Gamma$ in B^0–$\overline{B}^0$ system.

`PARJ(77)` : (D=10.) mixing parameter $x_s = \Delta M/\Gamma$ in B^0_s–$\overline{B}^0_s$ system.

`PARJ(80) - PARJ(90)` : parameters for time-like parton showers, see section 10.4.

`PARJ(91)` : (D=0.020 GeV) minimum particle width in `PMAS(KC,2)`, above which particle decays are assumed to take place before the stage where Bose–Einstein effects are introduced.

`PARJ(92)` : (D=1.) nominal strength of Bose–Einstein effects for $Q = 0$, see `MSTJ(51)`. This parameter, often denoted λ, expresses the amount of incoherence in particle production. Due to the simplified picture used for the Bose–Einstein effects, in particular for effects from three nearby identical particles, the actual λ of the simulated events may be larger than the input value.

`PARJ(93)` : (D=0.20 GeV) size of the Bose–Einstein effect region in terms of the Q variable, see `MSTJ(51)`. The more conventional measure, in terms of the radius R of the production volume, is given by $R = \hbar/$`PARJ(93)`≈ 0.2 fm$\times$GeV/`PARJ(93)`=`PARU(3)`/`PARJ(93)`.

`PARJ(94)` : (D=0.0 GeV) Increase radius for pairs from different W/Z/strings.

- `< 0` : if `MSTJ(56) = 1`, the radius for pairs from different W/Z's is increased to $R + \delta R_{WW} +$ `PARU(3)`/abs(`PARJ(94)`).
- `> 0` : the radius for pairs from different strings is increased to $R +$ `PARU(3)`/`PARJ(94)`.

`PARJ(95)` : (R) Set to the energy imbalance after the BE algorithm, before rescaling of momenta.

`PARJ(96)` : (R) Set to the α needed to retain energy-momentum conservation in each event for relevant models.

`PARJ(121) - PARJ(171)` : parameters for e^+e^- event generation, see section 6.3.

`PARJ(180) - PARJ(195)` : various coupling constants and parameters related to couplings, see section 9.4.

14.3.1 The advanced popcorn code for baryon production

In section 12.1.3 a new advanced popcorn code for baryon production model was presented, based on [Edé97]. It partly overwrites and redefines the meaning of some of the parameters

above. Therefore the full description of these new options are given separately in this section, together with a listing of the new routines involved.

In order to use the new options, a few possibilities are open.

- Use of the old diquark and popcorn models, `MSTJ(12) = 1` and `2`, is essentially unchanged. Note, however, that `PARJ(19)` is available for an ad-hoc suppression of first-rank baryon production.
- Use of the old popcorn model with new **SU(6)** weighting:
  - Set `MSTJ(12)=3`.
  - Increase `PARJ(1)` by approximately a factor 1.2 to retain about the same effective baryon production rate as in `MSTJ(12)=2`.
  - Note: the new **SU(6)** weighting e.g. implies that the total production rate of charm and bottom baryons is reduced.
- Use of the old flavour model with new **SU(6)** treatment and modified fragmentation function for diquark vertices (which softens baryon spectra):
  - Set `MSTJ(12)=4`.
  - Increase `PARJ(1)` by about a factor 1.7 and `PARJ(5)` by about a factor 1.2 to restore the baryon and popcorn rates of the `MSTJ(12)=2` default.
- Use of the new flavour model (automatically with modified diquark fragmentation function.)
  - Set `MSTJ(12)=5`.
  - Increase `PARJ(1)` by approximately a factor 2.
  - Change `PARJ(18)` from 1 to approx. 0.19.
  - Instead of `PARJ(3) - PARJ(7)`, tune `PARJ(8)`, `PARJ(9)`, `PARJ(10)` and `PARJ(18)`. (Here `PARJ(10)` is used only in collisions having remnants of baryon beam particles.)
  - Note: the proposed parameter values are based on a global fit to all baryon production rates. This e.g. means that the proton rate is lower than in the `MSTJ(12)=2` option, with current data somewhere in between. The `PARJ(1)` value would have to be about 3 times higher in `MSTJ(12)=5` than in `=2` to have the same total baryon production rate (=proton+neutron), but then other baryon rates would not match at all.
- The new options `MSTJ(12)=4` and `=5` (and, to some extent, `=3`) soften baryon spectra in such a way that `PARJ(45)` (the change of *a* for diquarks in the Lund symmetric fragmentation function) is available for a retune. It affects i.e. baryon-antibaryon rapidity correlations and the baryon excess over antibaryons in quark jets.

The changes in and additions to the commonblocks are as follows.

`MSTU(121) - MSTU(125)` : Internal flags and counters; only MSTU(123) may be touched by you.

`MSTU(121)` : Popcorn meson counter.

`MSTU(122)` : Points at the proper diquark production weights, to distinguish between ordinary popcorn and rank 0 diquark systems. Only needed if `MSTJ(12)=5`.

`MSTU(123)` : Initialization flag. If `MSTU(123)` is 0 in a `PYKFDI` call, `PYKFIN` is called and `MSTU(123)` set to 1. Would need to be reset by you if flavour parameters are changed in the middle of a run.

`MSTU(124)` : First parton flavour in decay call, stored to easily find random flavour partner in a popcorn system.

`MSTU(125)` : Maximum number of popcorn mesons allowed in decay flavour generation. If a larger popcorn system passes the fake string suppressions, the error `KF=0` is returned and the flavour generation for the decay is restarted.

`MSTU(131) - MSTU(140)` : Store of popcorn meson flavour codes in decay algorithm. Purely internal.

`MSTJ(12)` : (D=2) Main switch for choice of baryon production model. Suppression of rank 1 baryons by a parameter `PARJ(19)` is no longer governed by the `MSTJ(12)` switch, but instead turned on by setting `PARJ(19)<1`. Three new options are available:

- `= 3` : as `=2`, but additionally the production of first rank baryons may be suppressed by a factor `PARJ(19)`.
- `= 4` : as `=2`, but diquark vertices suffers an extra suppression of the form $1 - \exp(\rho\Gamma)$, where $\rho \approx 0.7\mathrm{GeV}^{-2}$ is stored in `PARF(192)`.
- `= 5` : Advanced version of the popcorn model. Independent of `PARJ(3-7)`. Instead depending on `PARJ(8-10)`. When using this option `PARJ(1)` needs to enhanced by approx. a factor 2 (i.e. it losses a bit of its normal meaning), and `PARJ(18)` is suggested to be set to 0.19.

`PARJ(8), PARJ(9)` : (D=0.6, 1.2 GeV^{-1}) The new popcorn parameters β_u and $\delta\beta = \beta_s - \beta_u$. Used to suppress popcorn mesons of total invariant mass $M_\perp$ by $\exp(-\beta_q * M_\perp)$. Larger `PARJ(9)` leads to a stronger suppression of popcorn systems surrounded by an $s\bar{s}$ pair, and also a little stronger suppression of strangeness in diquarks.

`PARJ(10)` : (D=0.6 GeV^{-1}) Corresponding parameter for suppression of leading rank mesons of transverse mass $M_\perp$ in the fragmentation of diquark jets, used if `MSTJ(12)`=5. The treatment of original diquarks is flavour independent, i.e. `PARJ(10)` is used even if the diquark contains s or heavier quarks.

`PARF(131) - PARF(190)` : Different diquark and popcorn weights, calculated in `PYKFIN`, which is automatically called from `PYKFDI`.

- `PARF(131)` : Popcorn ratio $BM\overline{B}/B\overline{B}$ in the old model.
- `PARF(132-134)` : Leading rank meson ratio MB/B in the old model, for original diquark with 0, 1 and 2 s-quarks, respectively.
- `PARF(135-137)` : Colour fluctuation quark ratio, i.e. the relative probability that the heavier quark in a diquark fits into the baryon at the opposite side of the popcorn meson. For sq, original sq and original cq diquarks, respectively.
- `PARF(138)` : The extra suppression of strange colour fluctuation quarks, due to the requirement of surrounding a popcorn meson. (In the old model, it is simply `PARJ(6)`.)
- `PARF(139)` : Preliminary suppression of a popcorn meson in the new model. A system of N popcorn mesons is started with weight proportional to `PARF(139)`^{N}. It is then tested against the correct weight, derived from the mass of the system. For strange colour fluctuation quarks, the weight is `PARF(138)*PARF(139)`.
- `PARF(140)` : Preliminary suppression of leading rank mesons in diquark strings, irrespective of flavour. Corresponds to `PARF(139)`.
- `PARF(141-145)` : Maximal **SU(6)** factors for different types of diquarks.
- `PARF(146)` : Σ/Λ suppression if `MSTJ(12)=5`, derived from `PARJ(18)`.
- `PARF(151-190)` : Production ratios for different diquarks. Stored in four groups, handling $q \to B\overline{B}$, $q \to BM...\overline{B}$, $qq \to MB$ and finally $qq \to MB$ in the case of original diquarks. In each group is stored:
  - 1 : s/u colour fluctuation ratio.
  - 2,3 : s/u ratio for the vertex quark if the colour fluctuation quark is light or strange, respectively.
  - 4 : q/q' vertex quark ratio if the colour fluctuation quark is light and $= q$.
  - 5-7 : (spin 1)/(spin 0) ratio for su, us and ud, where the first flavour is the colour fluctuation quark.
  - 8-10 : Unused.

`PARF(191)` : (D=0.2) Non-constituent mass in GeV of a ud_0 diquark. Used in combination with diquark constituent mass differences to derive relative production rates for different diquark flavours in the `MSTJ(12)`=5 option.

`PARF(192)` : (D=0.5) Parameter for the low-Γ suppression of diquark vertices in the `MSTJ(12)`≥ 4 options. `PARF(192)` represents $e^{-\rho}$, i.e. the suppression is of the form 1-`PARF(192)`$^{\Gamma}$, Γ in GeV^2.

`PARF(193,194)` : (I) Store of some popcorn weights used by the present popcorn system.

`PARF(201-1400)` : (I) Weights for every possible popcorn meson construction in the `MSTJ(12)`=5 option. Calculated from input parameters and meson masses in `PYKFIN`. When $q_1q_2 \to M + q_1q_3$, the weights for M and the new diquark depends not only on q_1 and q_2. It is also important if this is a 'true' popcorn system, or a system which started with a diquark at the string end, and if M is the final meson of the popcorn system, i.e. if the q_1q_3 diquark will go into a baryon or not. With five possible flavours for q_1 and q_2 this gives 80 different situations when selecting M and q_3. However, quarks heavier than s only exist in the string endpoints, and if more popcorn mesons are to be produced, the q_1q_3 diquark does not influence the weights and the q_1 dependence reduces to what β factor (`PARJ(8-10)`) that is used. Then 40 distinct situations remains, i.e.:

'true popcorn'	final meson	q_1	q_2
YES	YES	d,u,s	d,u,s
	NO	<s,s	d,u,s
NO	YES	d,u,s,>s	d,u,s,c,b
	NO	1 case	d,u,s,c,b

This table also shows the order in which the situations are stored. E.g. situation no. 1 is 'YES,YES,d,d', situation no.11 is 'YES,NO,$< s$,u'.

In every situation q_3 can be d, u or s. if $q_3 = q_2$ there are in the program three possible flavour mixing states available for the meson. This gives five possible meson flavours, and for each one of them there are six possible L, S spin states. Thus 30 `PARF` positions are reserved for each situation, and these are used as follows:

For each spin multiplet (in the same order as in `PARF(1-60)`) five positions are reserved. First are stored the weights for the the $q_3 \neq q_2$ mesons, with q_3 in increasing order. If $q_2 >$ s, this occupies three spots, and the final two are unused. If $q_2 \leq$ s, the final three spots are used for the diagonal states when $q_3 = q_2$.

In summary, all commonblock variables are completely internal, except `MSTU(123)`, `MSTJ(12)`, `PARJ(8) - PARJ(10)` and `PARF(191), PARF(192)`. Among these, `PARF(191)` and `PARF(192)` should not need to be changed. `MSTU(123)` should be 0 when starting, and reset to 0 whenever changing a switch or parameter which influences flavour weight With `MSTJ(12)=4`, `PARJ(5)` may need to increase. With `MSTJ(12)=5`, a preliminary tune suggests `PARJ(8) = 0.6`, `PARJ(9) = 1.2`, `PARJ(10) = 0.6`, `PARJ(1) = 0.20` and `PARJ(18)=0.19`.

Three new subroutines are added, but are only needed for internal use.

`SUBROUTINE PYKFIN` : to calculate a large set of diquark and popcorn weights from input parameters. Is called from `PYKFDI` if `MSTU(123)`=0. Sets `MSTU(123)` to 1.

`SUBROUTINE PYNMES(KFDIQ)` : to calculate number of popcorn mesons to be generated in a popcorn system, or the number of leading rank mesons when fragmenting a diquark string. Stores the number in `MSTU(121)`. Always returns 0 if `MSTJ(12)`< 2. Returns 0 or 1 if `MSTJ(12)`< 5.

`KFDIQ` : Flavour of the diquark in a diquark string. If starting a popcorn system inside a string, `KFDIQ` is 0.

`SUBROUTINE PYDCYK(KFL1,KFL2,KFL3,KF)` : to generate flavours in the phase space model of hadron decays, and in cluster decays. Is essentially the same as a `PYKFDI` call, but also takes into account the effects of string dynamics in flavour production in the `MSTJ(12)`≥ 4 options. This is done in order to get a reasonable interpretation of the input parameters also for hadron decays with these options.

`KFL1,KFL2,KFL3,KF` : See `SUBROUTINE PYKFDI`.

Internally the diquark codes have been extended to store the necessary further popcorn infor-

mation. As before, an initially existing diquark has a code of the type $1000q_a + 100q_b + 2s + 1$, where $q_a > q_b$. Diquarks created in the fragmentation process now have the longer code $10000q_c + 1000q_a + 100q_b + 2s + 1$, i.e. one further digit is set. Here q_c is the curtain quark, i.e. the flavour of the quark-antiquark pair that is shared between the baryon and the antibaryon, either q_a or q_b. The non-curtain quark, the other of q_a and q_b, may have its antiquark partner in a popcorn meson. In case there are no popcorn mesons this information is not needed, but is still set at random to be either of q_a and q_b. The extended code is used internally in PYSTRF and PYDECY and in some routines called by them, but is not visible in any event listings.

14.4 Further Parameters and Particle Data

The PYUPDA routine is the main tool for updating particle data tables. The following common blocks are maybe of a more peripheral interest, with the exception of the MDCY array, which allows a selective inhibiting of particle decays, and setting masses of not yet discovered particles, such as PMAS(25,1), the (Standard Model) Higgs mass.

```
CALL PYUPDA(MUPDA,LFN)
```

Purpose: to give you the ability to update particle data, or to keep several versions of modified particle data for special purposes (e.g. bottom studies).

MUPDA : gives the type of action to be taken.

= 1 : write a table of particle data, that you then can edit at leisure. For ordinary listing of decay data, PYLIST(12) should be used, but that listing could not be read back in by the program.
For each compressed flavour code KC = 1–500, one line is written containing the corresponding uncompressed KF flavour code (1X,I9) in KCHG(KC,4), the particle and antiparticle names (2X,A16,2X,A16) in CHAF, the electric (I3), colour charge (I3) and particle/antiparticle distinction (I3) codes in KCHG, the mass (F12.5), the mass width (F12.5), maximum broadening (F12.5) and average proper lifetime (1P,E13.5) in PMAS, the resonance width treatment (I3) in MWID and the on/off decay switch (I3) in MDCY(KC,1).
After a KC line follows one line for each possible decay channel, containing the MDME codes (10X,2I5), the branching ratio (F12.6) in BRAT, and the KFDP codes for the decay products (5I10), with trailing 0's if the number of decay products is smaller than 5.

= 2 : read in particle data, as written with =1 and thereafter edited by you, and use this data subsequently in the current run. This also means e.g. the mapping between the full KF and compressed KC flavour codes. Reading is done with fixed format, which means that you have to preserve the format codes described for =1 during the editing. A number of checks will be made to see if input looks reasonable, with warnings if not. If some decay channel is said not to conserve charge, it should be taken seriously. Warnings that decay is kinematically unallowed need not be as serious, since that particular decay mode may not be switched on unless the particle mass is increased.

= 3 : read in particle data, like option 2, but use it as a complement to rather than a replacement of existing data. The input file should therefore only contain new particles and particles with changed data. New particles are added to the bottom of the KC and decay channel tables. Changed particles retain their KC codes and hence the position of particle data, but their old decay channels are removed, this space is recuperated, and new decay channels are added at the end. Thus also the decay channel numbers of unchanged particles are affected.

= 4 : write current particle data as data lines, which can be edited into BLOCK DATA PYDATA for a permanent replacement of the particle data. This option is intended for the program author only, not for you.

LFN : the file number which the data should be written to or read from. You must see to it that this file is properly opened for read or write (since the definition of file names is platform dependent).

```
COMMON/PYDAT2/KCHG(500,4),PMAS(500,4),PARF(2000),VCKM(4,4)
```

Purpose: to give access to a number of flavour treatment constants or parameters and particle/parton data. Particle data is stored by compressed code KC rather than by the full KF code. You are reminded that the way to know the KC value is to use the PYCOMP function, i.e. KC = PYCOMP(KF).

KCHG(KC,1) : three times particle/parton charge for compressed code KC.

KCHG(KC,2) : colour information for compressed code KC.

= 0 : colour-singlet particle.

= 1 : quark or antidiquark.

= -1 : antiquark or diquark.

= 2 : gluon.

KCHG(KC,3) : particle/antiparticle distinction for compressed code KC.

= 0 : the particle is its own antiparticle.

= 1 : a nonidentical antiparticle exists.

KCHG(KC,4) : equals the uncompressed particle code KF (always with a positive sign). This gives the inverse mapping of what is provided by the PYCOMP routine.

PMAS(KC,1) : particle/parton mass m (in GeV) for compressed code KC.

PMAS(KC,2) : the total width Γ (in GeV) of an assumed symmetric Breit–Wigner mass shape for compressed particle code KC.

PMAS(KC,3) : the maximum deviation (in GeV) from the PMAS(KC,1) value at which the Breit–Wigner shape above is truncated. Is used in ordinary particle decays, but not in the resonance treatment; cf. the CKIN variables.

PMAS(KC,4) : the average lifetime τ for compressed particle code KC, with $c\tau$ in mm, i.e. τ in units of about 3.33×10^{-12} s.

PARF(1) - PARF(60) : give a parameterization of the $d\bar{d}$–$u\bar{u}$–$s\bar{s}$ flavour mixing in production of flavour-diagonal mesons. Numbers are stored in groups of 10, for the six multiplets pseudoscalar, vector, axial vector ($S = 0$), scalar, axial vector ($S = 1$) and tensor, in this order; see section 12.1.2. Within each group, the first two numbers determine the fate of a $d\bar{d}$ flavour state, the second two that of a $u\bar{u}$ one, the next two that of an $s\bar{s}$ one, while the last four are unused. Call the numbers of a pair p_1 and p_2. Then the probability to produce the state with smallest KF code is $1 - p_1$, the probability for the middle one is $p_1 - p_2$ and the probability for the one with largest code is p_2, i.e. p_1 is the probability to produce either of the two 'heavier' ones.

PARF(61) - PARF(80) : give flavour **SU(6)** weights for the production of a spin 1/2 or spin 3/2 baryon from a given diquark–quark combination. Should not be changed.

PARF(91) - PARF(96) : (D = 0.0099, 0.0056, 0.199, 1.35, 4.5, 165 GeV) default nominal quark masses, used to give the starting value for running masses calculated in PYMRUN.

PARF(101) - PARF(105) : contain d, u, s, c and b constituent masses, in the past used in mass formulae for undiscovered hadrons, and should not be changed.

PARF(111), PARF(112) : (D=0.0, 0.11 GeV) constant terms in the mass formulae for heavy mesons and baryons, respectively (with diquark getting 2/3 of baryon).

PARF(113), PARF(114) : (D=0.16, 0.048 GeV) factors which, together with Clebsch-

Gordan coefficients and quark constituent masses, determine the mass splitting due to spin-spin interactions for heavy mesons and baryons, respectively. The latter factor is also used for the splitting between spin 0 and spin 1 diquarks.

`PARF(115) - PARF(118) :` (D=0.50, 0.45, 0.55, 0.60 GeV), constant mass terms, added to the constituent masses, to get the mass of heavy mesons with orbital angular momentum $L = 1$. The four numbers are for pseudovector mesons with quark spin 0, and for scalar, pseudovector and tensor mesons with quark spin 1, respectively.

`PARF(121), PARF(122) :` (D=0.1, 0.2 GeV) constant terms, which are subtracted for quark and diquark masses, respectively, in defining the allowed phase space in particle decays into partons (e.g. $B^0 \to \bar{c}du\bar{d}$).

`PARF(201) - PARF(1960) :` (D=1760*0) relative probabilities for flavour production in the `MSTJ(15)=1` option; to be defined by you before any PYTHIA calls. (The standard meaning is changed in the advanced baryon popcorn code, described in subsection 14.3.1, where many of the `PARF` numbers are used for other purposes.)

The index in `PARF` is of the compressed form

$120 + 80 \times$KTAB1$+25 \times$KTABS$+$KTAB3.

Here KTAB1 is the old flavour, fixed by preceding fragmentation history, while KTAB3 is the new flavour, to be selected according to the relevant relative probabilities (except for the very last particle, produced when joining two jets, where both KTAB1 and KTAB3 are known). Only the most frequently appearing quarks/diquarks are defined, according to the code $1 = d$, $2 = u$, $3 = s$, $4 = c$, $5 = b$, $6 = t$ (obsolete!), $7 = dd_1$, $8 = ud_0$, $9 = ud_1$, $10 = uu_1$, $11 = sd_0$, $12 = sd_1$, $13 = su_0$, $14 = su_1$, $15 = ss_1$, $16 = cd_0$, $17 = cd_1$, $18 = cu_0$, $19 = cu_1$, $20 = cs_0$, $21 = cs_1$, $22 = cc_1$. These are thus the only possibilities for the new flavour to be produced; for an occasional old flavour not on this list, the ordinary relative flavour production probabilities will be used.

Given the initial and final flavour, the intermediate hadron that is produced is almost fixed. (Initial and final diquark here corresponds to 'popcorn' production of mesons intermediate between a baryon and an antibaryon). The additional index KTABS gives the spin type of this hadron, with

0 = pseudoscalar meson or Λ-like spin 1/2 baryon,
1 = vector meson or Σ-like spin 1/2 baryon,
2 = tensor meson or spin 3/2 baryon.

(Some meson multiplets, not frequently produced, are not accessible by this parameterization.)

Note that some combinations of KTAB1, KTAB3 and KTABS do not correspond to a physical particle (a Λ-like baryon must contain three different quark flavours, a Σ-like one at least two), and that you must see to it that the corresponding `PARF` entries are vanishing. One additional complication exist when KTAB3 and KTAB1 denote the same flavour content (normally KTAB3=KTAB1, but for diquarks the spin freedom may give KTAB3=KTAB1$\pm$1): then a flavour neutral meson is to be produced, and here $d\bar{d}$, $u\bar{u}$ and $s\bar{s}$ states mix (heavier flavour states do not, and these are therefore no problem). For these cases the ordinary KTAB3 value gives the total probability to produce either of the mesons possible, while KTAB3=23 gives the relative probability to produce the lightest meson state (π^0, ρ^0, a_2^0), KTAB3=24 relative probability for the middle meson (η, ω, f_2^0), and KTAB3 = 25 relative probability for the heaviest one (η', ϕ, $f_2'^0$). Note that, for simplicity, these relative probabilities are assumed the same whether initial and final diquark have the same spin or not; the total probability may well be assumed different, however.

As a general comment, the sum of `PARF` values for a given KTAB1 need not be normalized to unity, but rather the program will find the sum of relevant weights and normalize to that. The same goes for the KTAB3=23–25 weights. This makes it straightforward to use one common setup of `PARF` values and still switch between

different MSTJ(12) baryon production modes, with the exception of the advanced popcorn scenarios.

VCKM(I,J) : squared matrix elements of the Cabibbo-Kobayashi-Maskawa flavour mixing matrix.

I : up type generation index, i.e. 1 = u, 2 = c, 3 = t and 4 = t′.

J : down type generation index, i.e. 1 = d, 2 = s, 3 = b and 4 = b′.

```
COMMON/PYDAT3/MDCY(500,3),MDME(8000,2),BRAT(8000),KFDP(8000,5)
```

Purpose: to give access to particle decay data and parameters. In particular, the MDCY(KC,1) variables may be used to switch on or off the decay of a given particle species, and the MDME(IDC,1) ones to switch on or off an individual decay channel of a particle. For quarks, leptons and gauge bosons, a number of decay channels are included that are not allowed for on-mass-shell particles, see MDME(IDC,2)=102. These channels are not directly used to perform decays, but rather to denote allowed couplings in a more general sense, and to switch on or off such couplings, as described elsewhere. Particle data is stored by compressed code KC rather than by the full KF code. You are reminded that the way to know the KC value is to use the PYCOMP function, i.e. KC = PYCOMP(KF).

MDCY(KC,1) : switch to tell whether a particle with compressed code KC may be allowed to decay or not.

= 0 : the particle is not allowed to decay.

= 1 : the particle is allowed to decay (if decay information is defined below for the particle).

Warning: these values may be overwritten for resonances in a PYINIT call, based on the MSTP(41) option you have selected. If you want to allow resonance decays in general but switch off the decay of one particular resonance, this is therefore better done after the PYINIT call.

MDCY(KC,2) : gives the entry point into the decay channel table for compressed particle code KC. Is 0 if no decay channels have been defined.

MDCY(KC,3) : gives the total number of decay channels defined for compressed particle code KC, independently of whether they have been assigned a non-vanishing branching ratio or not. Thus the decay channels are found in positions MDCY(KC,2) to MDCY(KC,2)+MDCY(KC,3)-1.

MDME(IDC,1) : on/off switch for individual decay channel IDC. In addition, a channel may be left selectively open; this has some special applications in the event generation machinery. Effective branching ratios are automatically recalculated for the decay channels left open. Also process cross sections are affected; see section 7.6.2. If a particle is allowed to decay by the MDCY(KC,1) value, at least one channel must be left open by you. A list of decay channels with current IDC numbers may be obtained with PYLIST(12).

= -1 : this is a non-Standard Model decay mode, which by default is assumed not to exist. Normally, this option is used for decays involving fourth generation or $H^{\pm}$ particles.

= 0 : channel is switched off.

= 1 : channel is switched on.

= 2 : channel is switched on for a particle but off for an antiparticle. It is also on for a particle which is its own antiparticle, i.e. here it means the same as =1.

= 3 : channel is switched on for an antiparticle but off for a particle. It is off for a particle which is its own antiparticle.

= 4 : in the production of a pair of equal or charge conjugate resonances in PYTHIA,

say $h^0 \to W^+W^-$, either one of the resonances is allowed to decay according to this group of channels, but not both. If the two particles of the pair are different, the channel is on. For ordinary particles, not resonances, this option only means that the channel is switched off.

`= 5 :` as =4, but an independent group of channels, such that in a pair of equal or charge conjugate resonances the decay of either resonance may be specified independently. If the two particles in the pair are different, the channel is off. For ordinary particles, not resonances, this option only means that the channel is switched off.

Warning: the two values -1 and 0 may look similar, but in fact are quite different. In neither case the channel so set is generated, but in the latter case the channel still contributes to the total width of a resonance, and thus affects both simulated line shape and the generated cross section when PYTHIA is run. The value 0 is appropriate to a channel we assume exists, even if we are not currently simulating it, while -1 should be used for channels we believe do not exist. In particular, you are warned unwittingly to set fourth generation channels 0 (rather than -1), since by now the support for a fourth generation is small.

Remark: all the options above may be freely mixed. The difference, for those cases where both make sense, between using values 2 and 3 and using 4 and 5 is that the latter automatically include charge conjugate states, e.g. $h^0 \to W^+W^- \to e^+\nu_e d\bar{u}$ or $\bar{d}ue^-\bar{\nu}_e$, but the former only one of them. In calculations of the joint branching ratio, this makes a factor 2 difference.

Example: to illustrate the above options, consider the case of a W^+W^- pair. One might then set the following combination of switches for the W:

channel	value	comment
$u\bar{d}$	1	allowed for W^+ and W^- in any combination,
$u\bar{s}$	0	never produced but contributes to W width,
$c\bar{d}$	2	allowed for W^+ only,
$c\bar{s}$	3	allowed for W^- only, i.e. properly $W^- \to \bar{c}s$,
$t\bar{b}$	0	never produced but contributes to W width
		if the channel is kinematically allowed,
$\nu_e e^+$	4	allowed for one of W^+ or W^-, but not both,
$\nu_\mu \mu^+$	4	allowed for one of W^+ or W^-, but not both,
		and not in combination with $\nu_e e^+$,
$\nu_\tau \tau^+$	5	allowed for the other W, but not both,
$\nu_\chi \chi^-$	-1	not produced and does not contribute to W width.

A W^+W^- final state $u\bar{d} + \bar{c}s$ is allowed, but not its charge conjugate $\bar{u}d + c\bar{s}$, since the latter decay mode is not allowed for a W^+. The combination $\nu_e e^+ + \bar{\nu}_\tau \tau^-$ is allowed, since the two channels belong to different groups, but not $\nu_e e^+ + \bar{\nu}_\mu \mu^-$, where both belong to the same. Both $u\bar{d} + \bar{\nu}_\tau \tau^-$ and $\bar{u}d + \nu_\tau \tau^+$ are allowed, since there is no clash. The full rulebook, for this case, is given by eq. (111). A term r_i^2 means channel i is allowed for W^+ and W^- simultaneously, a term $r_i r_j$ that channels i and j may be combined, and a term $2r_i r_j$ that channels i and j may be combined two ways, i.e. that also a charge conjugate combination is allowed.

`MDME(IDC,2) :` information on special matrix-element treatment for decay channel IDC. Is mainly intended for the normal-particle machinery in `PYDECY`, so many of the codes are superfluous in the more sophisticated resonance decay treatment by `PYRESD`, see section 2.1.2. In addition to the outline below, special rules apply for the order in which decay products should be given, so that matrix elements and colour flow is

properly treated. One such example is the weak matrix elements, which only will be correct if decay products are given in the right order. The program does not police this, so if you introduce channels of your own and use these codes, you should be guided by the existing particle data.

`= 0 :` no special matrix-element treatment; partons and particles are copied directly to the event record, with momentum distributed according to phase space.

`= 1 :` ω and ϕ decays into three pions, eq. (244).

`= 2 :` π^0 or η Dalitz decay to γe^+e^-, eq. (246).

`= 3 :` used for vector meson decays into two pseudoscalars, to signal non-isotropic decay angle according to eq. (245), where relevant.

`= 4 :` decay of a spin 1 onium resonance to three gluons or to a photon and two gluons, eq. (43). The gluons may subsequently develop a shower if `MSTJ(23)=1`.

`= 11 :` phase-space production of hadrons from the quarks available.

`= 12 :` as `=11`, but for onia resonances, with the option of modifying the multiplicity distribution separately.

`= 13 :` as `=11`, but at least three hadrons to be produced (useful when the two-body decays are given explicitly).

`= 14 :` as `=11`, but at least four hadrons to be produced.

`= 15 :` as `=11`, but at least five hadrons to be produced.

`= 22 - 30 :` phase-space production of hadrons from the quarks available, with the multiplicity fixed to be `MDME(IDC,2)-20`, i.e. 2–10.

`= 31 :` two or more quarks and particles are distributed according to phase space. If three or more products, the last product is a spectator quark, i.e. sitting at rest with respect to the decaying hadron.

`= 32 :` a $q\overline{q}$ or gg pair, distributed according to phase space (in angle), and allowed to develop a shower if `MSTJ(23)=1`.

`= 33 :` a triplet $qX\overline{q}$, where X is either a gluon or a colour-singlet particle; the final particle ($\overline{q}$) is assumed to sit at rest with respect to the decaying hadron, and the two first particles (q and X) are allowed to develop a shower if `MSTJ(23)=1`. Nowadays superfluous.

`= 41 :` weak decay, where particles are distributed according to phase space, multiplied by a factor from the expected shape of the momentum spectrum of the direct product of the weak decay (the ν_τ in τ decay).

`= 42 :` weak decay matrix element for quarks and leptons. Products may be given either in terms of quarks or hadrons, or leptons for some channels. If the spectator system is given in terms of quarks, it is assumed to collapse into one particle from the onset. If the virtual W decays into quarks, these quarks are converted to particles, according to phase space in the W rest frame, as in `=11`. Is intended for τ, charm and bottom.

`= 43 :` as `=42`, but if the W decays into quarks, these will either appear as jets or, for small masses, collapse into a one- or two-body system. Nowadays superfluous.

`= 44 :` weak decay matrix element for quarks and leptons, where the spectator system may collapse into one particle for a small invariant mass. If the first two decay products are a $q\overline{q}'$ pair, they may develop a parton shower if `MSTJ(23)=1`. Was intended for top and beyond, but is nowadays superfluous.

`= 46 :` W (`KF` = 89) decay into $q\overline{q}'$ or $\ell\nu_\ell$ according to relative probabilities given by couplings (as stored in the `BRAT` vector) times a dynamical phase-space factor given by the current W mass. In the decay, the correct $V-A$ angular distribution is generated if the W origin is known (heavy quark or lepton). This is therefore the second step of a decay with `MDME=45`. A $q\overline{q}'$ pair may subsequently develop a shower if `MSTJ(23)=1`. Was intended for top and beyond, but is nowadays superfluous.

= 48 : as =42, but require at least three decay products.

= 50 : (default behaviour, also obtained for any other code value apart from the ones listed below) do not include any special threshold factors. That is, a decay channel is left open even if the sum of daughter nominal masses is above the mother actual mass, which is possible if at least one of the daughters can be pushed off the mass shell. Is intended for decay treatment in PYRESD with PYWIDT calls, and has no special meaning for ordinary PYDECY calls.

= 51 : a step threshold, i.e. a channel is switched off when the sum of daughter nominal masses is above the mother actual mass. Is intended for decay treatment in PYRESD with PYWIDT calls, and has no special meaning for ordinary PYDECY calls.

= 52 : a β-factor threshold, i.e. $\sqrt{(1-m_1^2/m^2-m_2^2/m^2)^2-4m_1^2m_2^2/m^4}$, assuming that the values stored in PMAS(KC,2) and BRAT(IDC) did not include any threshold effects at all. Is intended for decay treatment in PYRESD with PYWIDT calls, and has no special meaning for ordinary PYDECY calls.

= 53 : as =52, but assuming that PMAS(KC,2) and BRAT(IDC) did include the threshold effects, so that the weight should be the ratio of the β value at the actual mass to that at the nominal one. Is intended for decay treatment in PYRESD with PYWIDT calls, and has no special meaning for ordinary PYDECY calls.

= 101 : this is not a proper decay channel, but only to be considered as a continuation line for the decay product listing of the immediately preceding channel. Since the KFDP array can contain five decay products per channel, with this code it is possible to define channels with up to ten decay products. It is not allowed to have several continuation lines after each other.

= 102 : this is not a proper decay channel for a decaying particle on the mass shell (or nearly so), and is therefore assigned branching ratio 0. For a particle off the mass shell, this decay mode is allowed, however. By including this channel among the others, the switches MDME(IDC,1) may be used to allow or forbid these channels in hard processes, with cross sections to be calculated separately. As an example, $\gamma \to u\bar{u}$ is not possible for a massless photon, but is an allowed channel in e^+e^- annihilation.

BRAT(IDC) : give branching ratios for the different decay channels. In principle, the sum of branching ratios for a given particle should be unity. Since the program anyway has to calculate the sum of branching ratios left open by the MDME(IDC,1) values and normalize to that, you need not explicitly ensure this normalization, however. (Warnings are printed in PYUPDA(2) or PYUPDA(3) calls if the sum is not unity, but this is entirely intended as a help for finding user mistypings.) For decay channels with MDME(IDC,2)$>$ 80 the BRAT values are dummy.

KFDP(IDC,J) : contain the decay products in the different channels, with five positions J= 1–5 reserved for each channel IDC. The decay products are given following the standard KF code for partons and particles, with 0 for trailing empty positions. Note that the MDME(IDC+1,2)=101 option allows you to double the maximum number of decay product in a given channel from 5 to 10, with the five latter products stored KFDP(IDC+1,J).

```
COMMON/PYDAT4/CHAF(500,2)
CHARACTER CHAF*16
```

Purpose: to give access to character type variables.

CHAF(KC,1) : particle name according to KC code.

`CHAF(KC,2)` : antiparticle name according to `KC` code when an antiparticle exists, else blank.

14.5 Miscellaneous Comments

The previous sections have dealt with the subroutine options and variables one at a time. This is certainly important, but for a full use of the capabilities of the program, it is also necessary to understand how to make different pieces work together. This is something that cannot be explained fully in a manual, but must also be learnt by trial and error. This section contains some examples of relationships between subroutines, common blocks and parameters. It also contains comments on issues that did not fit in naturally anywhere else, but still might be useful to have on record.

14.5.1 Interfacing to detector simulation

Very often, the output of the program is to be fed into a subsequent detector simulation program. It therefore becomes necessary to set up an interface between the `PYJETS` common block and the detector model. Preferrably this should be done via the `HEPEVT` standard common block, see section 5.4, but sometimes this may not be convenient. If a `PYEDIT(2)` call is made, the remaining entries exactly correspond to those an ideal detector could see: all non-decayed particles, with the exception of neutrinos. The translation of momenta should be trivial (if need be, a `PYROBO` call can be made to rotate the 'preferred' z direction to whatever is the longitudinal direction of the detector), and so should the translation of particle codes. In particular, if the detector simulation program also uses the standard Particle Data Group codes, no conversion at all is needed. The problem then is to select which particles are allowed to decay, and how decay vertex information should be used.

Several switches regulate which particles are allowed to decay. First, the master switch `MSTJ(21)` can be used to switch on/off all decays (and it also contains a choice of how fragmentation should be interfaced). Second, a particle must have decay modes defined for it, i.e. the corresponding `MDCY(KC,2)` and `MDCY(KC,3)` entries must be non-zero for compressed code `KC = PYCOMP(KF)`. This is true for all colour neutral particles except the neutrinos, the photon, the proton and the neutron. (This statement is actually not fully correct, since irrelevant 'decay modes' with `MDME(IDC,2)=102` exist in some cases.) Third, the individual switch in `MDCY(KC,1)` must be on. Of all the particles with decay modes defined, only $\mu^{\pm}$, $\pi^{\pm}$, $K^{\pm}$ and K^0_L are by default considered stable.

Finally, if `MSTJ(22)` does not have its default value 1, checks are also made on the lifetime of a particle before it is allowed to decay. In the simplest alternative, `MSTJ(22)=2`, the comparison is based on the average lifetime, or rather $c\tau$, measured in mm. Thus if the limit `PARJ(71)` is (the default) 10 mm, then decays of K^0_S, Λ, Σ^-, Σ^+, Ξ^-, Ξ^0 and Ω^- are all switched off, but charm and bottom still decay. No $c\tau$ values below 1 μm are defined. With the two options `MSTJ(22)=` 3 or 4, a spherical or cylindrical volume is defined around the origin, and all decays taking place inside this volume are ignored.

Whenever a particle is in principle allowed to decay, i.e. `MSTJ(21)` and `MDCY` on, an proper lifetime is selected once and for all and stored in `V(I,5)`. The `K(I,1)` is then also changed to 4. For `MSTJ(22)=1`, such a particle will also decay, but else it could remain in the event record. It is then possible, at a later stage, to expand the volume inside which decays are allowed, and do a new `PYEXEC` call to have particles fulfilling the new conditions (but not the old) decay. As a further option, the `K(I,1)` code may be put to 5, signalling that the particle will definitely decay in the next `PYEXEC` call, at the vertex position given (by you) in the `V` vector.

This then allows the PYTHIA decay routines to be used inside a detector simulation program, as follows. For a particle which did not decay before entering the detector, its point of decay is still well defined (in the absence of deflections by electric or magnetic fields), eq. (243). If it interacts before that point, the detector simulation program is left to handle things. If not, the `V`

vector is updated according to the formula above, `K(I,1)` is set to 5, and `PYEXEC` is called, to give a set of decay products, that can again be tracked.

A further possibility is to force particles to decay into specific decay channels; this may be particularly interesting for charm or bottom physics. The choice of channels left open is determined by the values of the switches `MDME(IDC,1)` for decay channel IDC (use `PYLIST(12)` to obtain the full listing). One or several channels may be left open; in the latter case effective branching ratios are automatically recalculated without the need for your intervention. It is also possible to differentiate between which channels are left open for particles and which for antiparticles. Lifetimes are not affected by the exclusion of some decay channels. Note that, whereas forced decays can enhance the efficiency for several kinds of studies, it can also introduce unexpected biases, in particular when events may contain several particles with forced decays, cf. section 7.6.2.

14.5.2 Parameter values

A non-trivial question is to know which parameter values to use. The default values stored in the program are based on comparisons with LEP $e^+e^- \to Z^0$ data at around 91 GeV [LEP90], using a parton-shower picture followed by string fragmentation. Some examples of more recent parameter sets are found in [Kno96]. If fragmentation is indeed a universal phenomenon, as we would like to think, then the same parameters should also apply at other energies and in other processes. The former aspect, at least, seems to be borne out by comparisons with lower-energy PETRA/PEP data and higher-energy LEP2 data. Note, however, that the choice of parameters is intertwined with the choice of perturbative QCD description. If instead matrix elements are used, a best fit to 30 GeV data would require the values `PARJ(21)=0.40`, `PARJ(41)=1.0` and `PARJ(42)=0.7`. With matrix elements one does not expect an energy independence of the parameters, since the effective minimum invariant mass cut-off is then energy dependent, i.e. so is the amount of soft gluon emission effects lumped together with the fragmentation parameters. This is indeed confirmed by the LEP data. A mismatch in the perturbative QCD treatment could also lead to small differences between different processes.

It is often said that the string fragmentation model contains a wealth of parameters. This is certainly true, but it must be remembered that most of these deal with flavour properties, and to a large extent factorize from the treatment of the general event shape. In a fit to the latter it is therefore usually enough to consider the parameters of the perturbative QCD treatment, like Λ in α_s and a shower cut-off Q_0 (or α_s itself and y_{min}, if matrix elements are used), the a and b parameter of the Lund symmetric fragmentation function (`PARJ(41)` and `PARJ(42)`) and the width of the transverse momentum distribution (σ =`PARJ(21)`). In addition, the a and b parameters are very strongly correlated by the requirement of having the correct average multiplicity, such that in a typical χ^2 plot, the allowed region corresponds to a very narrow but very long valley, stretched diagonally from small (a,b) pairs to large ones. As to the flavour parameters, these are certainly many more, but most of them are understood qualitiatively within one single framework, that of tunnelling pair production of flavours.

Since the use of independent fragmentation has fallen in disrespect, it should be pointed out that the default parameters here are not particularly well tuned to the data. This especially applies if one, in addition to asking for independent fragmentation, also asks for another setup of fragmentation functions, i.e. other than the standard Lund symmetric one. In particular, note that most fits to the popular Peterson/SLAC heavy-flavour fragmentation function are based on the actual observed spectrum. In a Monte Carlo simulation, one must then start out with something harder, to compensate for the energy lost by initial-state photon radiation and gluon bremsstrahlung. Since independent fragmentation is not collinear safe (i.e, the emission of a collinear gluon changes the properties of the final event), the tuning is strongly dependent on the perturbative QCD treatment chosen. All the parameters needed for a tuning of independent fragmentation are available, however.

14.6 Examples

A 10 GeV u quark jet going out along the $+z$ axis is generated with

```
CALL PY1ENT(0,2,10D0,0D0,0D0)
```

Note that such a single jet is not required to conserve energy, momentum or flavour. In the generation scheme, particles with negative p_z are produced as well, but these are automatically rejected unless `MSTJ(3)=-1`. While frequently used in former days, the one-jet generation option is not of much current interest.

In e.g. a leptoproduction event a typical situation could be a u quark going out in the $+z$ direction and a ud_0 target remnant essentially at rest. (Such a process can be simulated by PYTHIA, but here we illustrate how to do part of it yourself.) The simplest procedure is probably to treat the process in the c.m. frame and boost it to the lab frame afterwards. Hence, if the c.m. energy is 20 GeV and the boost $\beta_z = 0.996$ (corresponding to $x_B = 0.045$), then

```
CALL PY2ENT(0,2,2101,20D0)
CALL PYROBO(0,0,0D0,0D0,0D0,0D0,0.996D0)
```

The jets could of course also be defined and allowed to fragment in the lab frame with

```
CALL PY1ENT(-1,2,223.15D0,0D0,0D0)
CALL PY1ENT(2,12,0.6837D0,3.1416D0,0D0)
CALL PYEXEC
```

Note here that the target diquark is required to move in the backwards direction with $E - p_z = m_{\mathrm{p}}(1 - x_B)$ to obtain the correct invariant mass for the system. This is, however, only an artefact of using a fixed diquark mass to represent a varying target remnant mass, and is of no importance for the fragmentation. If one wants a nicer-looking event record, it is possible to use the following

```
CALL PY1ENT(-1,2,223.15D0,0D0,0D0)
MSTU(10)=1
P(2,5)=0.938D0*(1D0-0.045D0)
CALL PY1ENT(2,2101,0D0,0D0,0D0)
MSTU(10)=2
CALL PYEXEC
```

A 30 GeV $\mathrm{u\bar{u}g}$ event with $E_{\mathrm{u}} = 8$ GeV and $E_{\bar{\mathrm{u}}} = 14$ GeV is simulated with

```
CALL PY3ENT(0,2,21,-2,30D0,2D0*8D0/30D0,2D0*14D0/30D0)
```

The event will be given in a standard orientation with the u quark along the $+z$ axis and the $\bar{\mathrm{u}}$ in the $\pm z, +x$ half plane. Note that the flavours of the three partons have to be given in the order they are found along a string, if string fragmentation options are to work. Also note that, for 3-jet events, and particularly 4-jet ones, not all setups of kinematical variables x lie within the kinematically allowed regions of phase space.

All common block variables can obviously be changed by including the corresponding common block in the user-written main program. Alternatively, the routine `PYGIVE` can be used to feed in values, with some additional checks on array bounds then performed. A call

```
 CALL PYGIVE('MSTJ(21)=3;PMAS(C663,1)=210.;CHAF(401,1)=funnyino;'//
&'PMAS(21,4)=')
```

will thus change the value of MSTJ(21) to 3, the value of PMAS(PYCOMP(663),1) = PMAS(136,1) to 210., the value of CHAF(401,1) to 'funnyino', and print the current value of PMAS(21,4). Since old and new values of parameters changed are written to output, this may offer a convenient way of documenting non-default values used in a given run. On the other hand, if a variable is changed back and forth frequently, the resulting voluminous output may be undesirable, and a direct usage of the common blocks is then to be recommended (the output can also be switched off, see MSTU(13)).

A general rule of thumb is that none of the physics routines (PYSTRF, PYINDF, PYDECY, etc.) should ever be called directly, but only via PYEXEC. This routine may be called repeatedly for one single event. At each call only those entries that are allowed to fragment or decay, and have not yet done so, are treated. Thus

```
CALL PY2ENT(1,1,-1,20D0)        ! fill 2 partons without fragmenting
MSTJ(1)=0                       ! inhibit jet fragmentation
MSTJ(21)=0                      ! inhibit particle decay
MDCY(PYCOMP(111),1)=0           ! inhibit pi0 decay
CALL PYEXEC                     ! will not do anything
MSTJ(1)=1                       !
CALL PYEXEC                     ! partons will fragment, no decays
MSTJ(21)=2                      !
CALL PYEXEC                     ! particles decay, except pi0
CALL PYEXEC                     ! nothing new can happen
MDCY(PYCOMP(111),1)=1           !
CALL PYEXEC                     ! pi0:s decay
```

A partial exception to the rule above is PYSHOW. Its main application is for internal use by PYEEVT, PYDECY, and PYEVNT, but it can also be directly called by you. Note that a special format for storing colour-flow information in K(I,4) and K(I,5) must then be used. For simple cases, the PY2ENT can be made to take care of that automatically, by calling with the first argument negative.

```
CALL PY2ENT(-1,1,-2,80D0)       ! store d ubar with colour flow
CALL PYSHOW(1,2,80D0)           ! shower partons
CALL PYEXEC                     ! subsequent fragmentation/decay
```

For more complicated configurations, PYJOIN should be used.

It is always good practice to list one or a few events during a run to check that the program is working as intended. With

```
CALL PYLIST(1)
```

all particles will be listed and in addition total charge, momentum and energy of stable entries will be given. For string fragmentation these quantities should be conserved exactly (up to machine precision errors), and the same goes when running independent fragmentation with one of the momentum conservation options. PYLIST(1) gives a format that comfortably fits in an 80 column window, at the price of not giving the complete story. With PYLIST(2) a more extensive listing is obtained, and PYLIST(3) also gives vertex information. Further options are available, like PYLIST(12), which gives a list of particle data.

An event, as stored in the PYJETS common block, will contain the original partons and the whole decay chain, i.e. also particles which subsequently decayed. If parton showers are used, the amount of parton information is also considerable: first the on-shell partons before showers have been considered, then a K(I,1)=22 line with total energy of the showering subsystem, after that the complete shower history tree-like structure, starting off with the same initial partons (now off-shell), and finally the end products of the shower rearranged along the string directions. This detailed record is useful in many connections, but if one only wants to retain the final particles, superfluous information may be removed with PYEDIT. Thus e.g.

```
CALL PYEDIT(2)
```

will leave you with the final charged and neutral particles, except for neutrinos.

The information in PYJETS may be used directly to study an event. Some useful additional quantities derived from these, such as charge and rapidity, may easily be found via the PYK and PYP functions. Thus electric charge =PYP(I,6) (as integer, three times charge =PYK(I,6)) and true rapidity y with respect to the z axis = PYP(I,17).

A number of utility (MSTU, PARU) and physics (MSTJ, PARJ) switches and parameters are available in common block PYDAT1. All of these have sensible default values. Particle data is stored in common blocks PYDAT2, PYDAT3 and PYDAT4. Note that the data in the arrays KCHG, PMAS, MDCY CHAF and MWID is not stored by KF code, but by the compressed code KC. This code is not to be learnt by heart, but instead accessed via the conversion function PYCOMP, KC = PYCOMP(KF).

In the particle tables, the following particles are considered stable: the photon, $e^{\pm}$, $\mu^{\pm}$, $\pi^{\pm}$, $K^{\pm}$, K^0_L, p, $\overline{p}$, n, $\overline{n}$ and all the neutrinos. It is, however, always possible to inhibit the decay of any given particle by putting the corresponding MDCY value zero or negative, e.g. MDCY(PYCOMP(310),1)=0 makes K^0_S and MDCY(PYCOMP(3122),1)=0 Λ stable. It is also possible to select stability based on the average lifetime (see MSTJ(22)), or based on whether the decay takes place within a given spherical or cylindrical volume around the origin.

The Field-Feynman jet model [Fie78] is available in the program by changing the following values: MSTJ(1)=2 (independent fragmentation), MSTJ(3)=-1 (retain particles with $p_z < 0$; is not mandatory), MSTJ(11)=2 (choice of longitudinal fragmentation function, with the a parameter stored in PARJ(51) - PARJ(53)), MSTJ(12)=0 (no baryon production), MSTJ(13)=1 (give endpoint quarks $p_\perp$ as quarks created in the field), MSTJ(24)=0 (no mass broadening of resonances), PARJ(2)=0.5 (s/u ratio for the production of new $q\overline{q}$ pairs), PARJ(11)=PARJ(12)=0.5 (probability for mesons to have spin 1) and PARJ(21)=0.35 (width of Gaussian transverse momentum distribution). In addition only d, u and s single quark jets may be generated following the FF recipe. Today the FF 'standard jet' concept is probably dead and buried, so the numbers above should more be taken as an example of the flexibility of the program, than as something to apply in practice.

A wide range of independent fragmentation options are implemented, to be accessed with the master switch MSTJ(1)=2. In particular, with MSTJ(2)=1 a gluon jet is assumed to fragment like a random d, $\overline{d}$, u, $\overline{u}$, s or $\overline{s}$ jet, while with MSTJ(2)=3 the gluon is split into a $d\overline{d}$, $u\overline{u}$ or $s\overline{s}$ pair of jets sharing the energy according to the Altarelli-Parisi splitting function. Whereas energy, momentum and flavour is not explicitly conserved in independent fragmentation, a number of options are available in MSTJ(3) to ensure this 'post facto', e.g. MSTJ(3)=1 will boost the event to ensure momentum conservation and then (in the c.m. frame) rescale momenta by a common factor to obtain energy conservation, whereas MSTJ(3)=4 rather uses a method of stretching the jets in longitudinal momentum along the respective jet axis to keep angles between jets fixed.

15 Event Study and Analysis Routines

After an event has been generated, one may wish to list it, or process it further in various ways. The first section below describes some simple study routines of this kind, while the subsequent ones describe more sophisticated analysis routines.

To describe the complicated geometries encountered in multihadronic events, a number of event measures have been introduced. These measures are intended to provide a global view of the properties of a given event, wherein the full information content of the event is condensed into one or a few numbers. A steady stream of such measures are proposed for different purposes. Many are rather specialized or never catch on, but a few become standards, and are useful to have easy access to. PYTHIA therefore contains a number of routines that can be called for any event, and that will directly access the event record to extract the required information.

In the presentation below, measures have been grouped in three kinds. The first contains simple event shape quantities, such as sphericity and thrust. The second is jet finding algorithms. The third is a mixed bag of particle multiplicities and compositions, factorial moments and energy–energy correlations, put together in a small statistics package.

None of the measures presented here are Lorentz invariant. The analysis will be performed in whatever frame the event happens to be given in. It it therefore up to you to decide whether the frame in which events were generated is the right one, or whether events beforehand should be boosted, e.g. to the c.m. frame. You can also decide which particles you want to have affected by the analysis.

15.1 Event Study Routines

After a PYEVNT call, or another similar physics routine call, the event generated is stored in the PYJETS common block, and whatever physical variable is desired may be constructed from this record. An event may be rotated, boosted or listed, and particle data may be listed or modified. Via the functions PYK and PYP the values of some frequently appearing variables may be obtained more easily. As described in subsequent sections, also more detailed event shape analyses may be performed simply.

```
CALL PYROBO(IMI,IMA,THE,PHI,BEX,BEY,BEZ)
```

Purpose: to perform rotations and Lorentz boosts (in that order, if both in the same call) of jet/particle momenta and vertex position variables.

IMI, IMA : range of entries affected by transformation, IMI$\leq$I$\leq$IMA. If 0 or below, IMI defaults to 1 and IMA to N. Lower and upper bounds given by positive MSTU(1) and MSTU(2) override the IMI and IMA values, and also the 1 to N constraint.

THE, PHI : standard polar coordinates θ, φ, giving the rotated direction of a momentum vector initially along the $+z$ axis.

BEX, BEY, BEZ : gives the direction and size β of a Lorentz boost, such that a particle initially at rest will have $\mathbf{p}/E = \beta$ afterwards.

Remark: all entries in the range IMI–IMA (or modified as described above) are affected, unless the status code of an entry is K(I,1)≤ 0.

```
CALL PYEDIT(MEDIT)
```

Purpose: to exclude unstable or undetectable jets/particles from the event record. One may also use PYEDIT to store spare copies of events (specifically initial parton configuration) that can be recalled to allow e.g. different fragmentation schemes to be run through with one and the same parton configuration. Finally, an event which has

been analyzed with PYSPHE, PYTHRU or PYCLUS (see section 15.5) may be rotated to align the event axis with the z direction.

MEDIT : tells which action is to be taken.

- = 0 : empty (K(I,1)=0) and documentation (K(I,1)>20) lines are removed. The jets/particles remaining are compressed in the beginning of the PYJETS common block and the N value is updated accordingly. The event history is lost, so that information stored in K(I,3), K(I,4) and K(I,5) is no longer relevant.
- = 1 : as =0, but in addition all jets/particles that have fragmented/decayed (K(I,1)>10) are removed.
- = 2 : as =1, but also all neutrinos and unknown particles (i.e. compressed code KC= 0) are removed.
- = 3 : as =2, but also all uncharged, colour neutral particles are removed, leaving only charged, stable particles (and unfragmented partons, if fragmentation has not been performed).
- = 5 : as =0, but also all partons which have branched or been rearranged in a parton shower and all particles which have decayed are removed, leaving only the fragmenting parton configuration and the final-state particles.
- = 11 : remove lines with K(I,1)<0. Update event history information (in K(I,3) - K(I,5)) to refer to remaining entries.
- = 12 : remove lines with K(I,1)=0. Update event history information (in K(I,3) - K(I,5)) to refer to remaining entries.
- = 13 : remove lines with K(I,1)= 11, 12 or 15, except for any line with K(I,2)=94. Update event history information (in K(I,3) - K(I,5)) to refer to remaining entries. In particular, try to trace origin of daughters, for which the mother is decayed, back to entries not deleted.
- = 14 : remove lines with K(I,1)= 13 or 14, and also any line with K(I,2)=94. Update event history information (in K(I,3) - K(I,5)) to refer to remaining entries. In particular, try to trace origin of rearranged jets back through the parton-shower history to the shower initiator.
- = 15 : remove lines with K(I,1)>20. Update event history information (in K(I,3) - K(I,5)) to refer to remaining entries.
- = 16 : try to reconstruct missing daughter pointers of decayed particles from the mother pointers of decay products. These missing pointers typically come from the need to use K(I,4) and K(I,5) also for colour flow information.
- = 21 : all partons/particles in current event record are stored (as a spare copy) in bottom of common block PYJETS (is e.g. done to save original partons before calling PYEXEC).
- = 22 : partons/particles stored in bottom of event record with =21 are placed in beginning of record again, overwriting previous information there (so that e.g. a different fragmentation scheme can be used on the same partons). Since the copy at bottom is unaffected, repeated calls with =22 can be made.
- = 23 : primary partons/particles in the beginning of event record are marked as not fragmented or decayed, and number of entries N is updated accordingly. Is simpe substitute for =21 plus =22 when no fragmentation/decay products precede any of the original partons/particles.
- = 31 : rotate largest axis, determined by PYSPHE, PYTHRU or PYCLUS, to sit along the z direction, and the second largest axis into the xz plane. For PYCLUS it can be further specified to $+z$ axis and xz plane with $x > 0$, respectively. Requires that one of these routines has been called before.
- = 32 : mainly intended for PYSPHE and PYTHRU, this gives a further alignment of the event, in addition to the one implied by =31. The 'slim' jet, defined as the side ($z > 0$ or $z < 0$) with the smallest summed $p_\perp$ over square root of number of particles, is rotated into the $+z$ hemisphere. In the opposite hemisphere

(now $z < 0$), the side of $x > 0$ and $x < 0$ which has the largest summed $|p_z|$ is rotated into the $z < 0, x > 0$ quadrant. Requires that PYSPHE or PYTHRU has been called before.

Remark: all entries 1 through N are affected by the editing. For options 0–5 lower and upper bounds can be explicitly given by MSTU(1) and MSTU(2).

```
CALL PYLIST(MLIST)
```

Purpose: to list an event, jet or particle data, or current parameter values.

MLIST : determines what is to be listed.

= 0 : writes a title page with program version number and last date of change; is mostly for internal use.

= 1 : gives a simple list of current event record, in an 80 column format suitable for viewing directly in a terminal window. For each entry, the following information is given: the entry number I, the parton/particle name (see below), the status code (K(I,1)), the flavour code KF (K(I,2)), the line number of the mother (K(I,3)), and the three-momentum, energy and mass (P(I,1) - P(I,5)). If MSTU(3) is non-zero, lines immediately after the event record proper are also listed. A final line contains information on total charge, momentum, energy and invariant mass.

The particle name is given by a call to the routine PYNAME. For an entry which has decayed/fragmented (K(I,1)= 11–20), this particle name is given within parentheses. Similarly, a documentation line (K(I,1)= 21–30) has the name enclosed in expression signs (!...!) and an event/jet axis information line the name within inequality signs (<...>). If the last character of the name is a '?', it is a signal that the complete name has been truncated to fit in, and can therefore not be trusted; this is very rare. For partons which have been arranged along strings (K(I,1)= 1, 2, 11 or 12), the end of the parton name column contains information about the colour string arrangement: an A for the first entry of a string, an I for all intermediate ones, and a V for the final one (a poor man's rendering of a vertical doublesided arrow, ↕).

It is possible to insert lines just consisting of sequences of ====== to separate different sections of the event record, see MSTU(70) - MSTU(80).

= 2 : gives a more extensive list of the current event record, in a 132 column format, suitable for wide terminal windows. For each entry, the following information is given: the entry number I, the parton/particle name (with padding as described for =1), the status code (K(I,1)), the flavour code KF (K(I,2)), the line number of the mother (K(I,3)), the decay product/colour-flow pointers (K(I,4), K(I,5)), and the three-momentum, energy and mass (P(I,1) - P(I,5)). If MSTU(3) is non-zero, lines immediately after the event record proper are also listed. A final line contains information on total charge, momentum, energy and invariant mass. Lines with only ====== may be inserted as for =1.

= 3 : gives the same basic listing as =2, but with an additional line for each entry containing information on production vertex position and time (V(I,1) - V(I,4)) and, for unstable particles, proper lifetime (V(I,5)).

= 5 : gives a simple listing of the event record stored in the HEPEVT common block. This is mainly intended as a tool to check how conversion with the PYHEPC routine works. The listing does not contain vertex information, and the flavour code is not displayed as a name.

= 7 : gives a simple listing of the parton-level event record for an external process, as stored in the HEPEUP common block. This is mainly intended as a tool to

check how reading from HEPEUP input works. The listing does not contain lifetime or spin information, and the flavour code is not displayed as a name. It also does not show other HEPEUP numbers, such as the event weight.

= 11 : provides a simple list of all parton/particle codes defined in the program, with KF code and corresponding particle name. The list is grouped by particle kind, and only within each group in ascending order.

= 12 : provides a list of all parton/particle and decay data used in the program. Each parton/particle code is represented by one line containing KF flavour code, KC compressed code, particle name, antiparticle name (where appropriate), electrical and colour charge and peresence or not of an antiparticle (stored in KCHG), mass, resonance width and maximum broadening, average proper lifetime (in PMAS) and whether the particle is considered stable or not (in MDCY). Immediately after a particle, each decay channel gets one line, containing decay channel number (IDC read from MDCY), on/off switch for the channel, matrix element type (MDME), branching ratio (BRAT), and decay products (KFDP). The MSTU(1) and MSTU(2) flags can be used to set the range of KF codes for which particles are listed.

= 13 : gives a list of current parameter values for MSTU, PARU, MSTJ and PARJ, and the first 200 entries of PARF. This is useful to keep check of which default values were changed in a given run.

Remark: for options 1–3 and 12 lower and upper bounds of the listing can be explicitly given by MSTU(1) and MSTU(2).

```
KK = PYK(I,J)
```

Purpose: to provide various integer-valued event data. Note that many of the options available (in particular I$>$ 0, J$\geq$ 14) which refer to event history will not work after a PYEDIT call. Further, the options 14–18 depend on the way the event history has been set up, so with the explosion of different allowed formats these options are no longer as safe as they may have been. For instance, option 16 can only work if MSTU(16)=2.

I=0, J= : properties referring to the complete event.

= 1 : N, total number of lines in event record.

= 2 : total number of partons/particles remaining after fragmentation and decay.

= 6 : three times the total charge of remaining (stable) partons and particles.

I>0, J= : properties referring to the entry in line no. I of the event record.

= 1 - 5 : K(I,1) - K(I,5), i.e. parton/particle status KS, flavour code KF and origin/decay product/colour-flow information.

= 6 : three times parton/particle charge.

= 7 : 1 for a remaining entry, 0 for a decayed, fragmented or documentation entry.

= 8 : KF code (K(I,2)) for a remaining entry, 0 for a decayed, fragmented or documentation entry.

= 9 : KF code (K(I,2)) for a parton (i.e. not colour neutral entry), 0 for a particle.

= 10 : KF code (K(I,2)) for a particle (i.e. colour neutral entry), 0 for a parton.

= 11 : compressed flavour code KC.

= 12 : colour information code, i.e. 0 for colour neutral, 1 for colour triplet, -1 for antitriplet and 2 for octet.

= 13 : flavour of ‘heaviest’ quark or antiquark (i.e. with largest code) in hadron or diquark (including sign for antiquark), 0 else.

= 14 : generation number. Beam particles or virtual exchange particles are generation 0, original jets/particles generation 1 and then 1 is added for each step in the fragmentation/decay chain.

= 15 : line number of ancestor, i.e. predecessor in first generation (generation 0 en-

tries are disregarded).

`= 16 :` rank of a hadron in the jet it belongs to. Rank denotes the ordering in flavour space, with hadrons containing the original flavour of the jet having rank 1, increasing by 1 for each step away in flavour ordering. All decay products inherit the rank of their parent. Whereas the meaning of a first-rank hadron in a quark jet is always well-defined, the definition of higher ranks is only meaningful for independently fragmenting quark jets. In other cases, rank refers to the ordering in the actual simulation, which may be of little interest.

`= 17 :` generation number after a collapse of a parton system into one particle, with 0 for an entry not coming from a collapse, and -1 for entry with unknown history. A particle formed in a collapse is generation 1, and then one is added in each decay step.

`= 18 :` number of decay/fragmentation products (only defined in a collective sense for fragmentation).

`= 19 :` origin of colour for showering parton, 0 else.

`= 20 :` origin of anticolour for showering parton, 0 else.

`= 21 :` position of colour daughter for showering parton, 0 else.

`= 22 :` position of anticolour daughter for showering parton, 0 else.

```
PP = PYP(I,J)
```

Purpose: to provide various real-valued event data. Note that some of the options available (`I`>0, `J`$=20$–25), which are primarily intended for studies of systems in their respective c.m. frame, requires that a `PYEXEC` call has been made for the current initial parton/particle configuration, but that the latest `PYEXEC` call has not been followed by a `PYROBO` one.

`I=0, J= :` properties referring to the complete event.

`= 1 - 4 :` sum of p_x, p_y, p_z and E, respectively, for the stable remaining entries.

`= 5 :` invariant mass of the stable remaining entries.

`= 6 :` sum of electric charge of the stable remaining entries.

`I>0, J= :` properties referring to the entry in line no. `I` of the event record.

`= 1 - 5 : P(I,1) - P(I,5)`, i.e. normally p_x, p_y, p_z, E and m for jet/particle.

`= 6 :` electric charge e.

`= 7 :` squared momentum $|\mathbf{p}|^2 = p_x^2 + p_y^2 + p_z^2$.

`= 8 :` absolute momentum $|\mathbf{p}|$.

`= 9 :` squared transverse momentum $p_\perp^2 = p_x^2 + p_y^2$.

`= 10 :` transverse momentum $p_\perp$.

`= 11 :` squared transverse mass $m_\perp^2 = m^2 + p_x^2 + p_y^2$.

`= 12 :` transverse mass $m_\perp$.

`= 13 - 14 :` polar angle θ in radians (between 0 and π) or degrees, respectively.

`= 15 - 16 :` azimuthal angle φ in radians (between $-\pi$ and π) or degrees, respectively.

`= 17 :` true rapidity $y = (1/2)\ln((E+p_z)/(E-p_z))$.

`= 18 :` rapidity y_π obtained by assuming that the particle is a pion when calculating the energy E, to be used in the formula above, from the (assumed known) momentum $\mathbf{p}$.

`= 19 :` pseudorapidity $\eta = (1/2)\ln((p+p_z)/(p-p_z))$.

`= 20 :` momentum fraction $x_p = 2|\mathbf{p}|/W$, where W is the total energy of the event, i.e. of the initial jet/particle configuration.

`= 21 :` $x_\mathrm{F} = 2p_z/W$ (Feynman-x if system is studied in the c.m. frame).

`= 22 :` $x_\perp = 2p_\perp/W$.

`= 23 :` $x_E = 2E/W$.

`= 24 :` $z_+ = (E+p_z)/W$.

= 25 : $z_- = (E - p_z)/W$.

```
COMMON/PYDAT1/MSTU(200),PARU(200),MSTJ(200),PARJ(200)
```

Purpose: to give access to a number of status codes that regulate the behaviour of the event study routines. The main reference for PYDAT1 is in section 14.3.

MSTU(1),MSTU(2) : (D=0,0) can be used to replace the ordinary lower and upper limits (normally 1 and N) for the action of PYROBO, and most PYEDIT and PYLIST calls. Are reset to 0 in a PYEXEC call.

MSTU(3) : (D=0) number of lines with extra information added after line N. Is reset to 0 in a PYEXEC call, or in an PYEDIT call when particles are removed.

MSTU(11) : (D=6) file number to which all program output is directed. It is your responsibility to see to it that the corresponding file is also opened for output.

MSTU(12) : (D=1) writing of title page (version number and last date of change for PYTHIA) on output file.

= 0 : not done.

= 1 : title page is written at first occasion, at which time MSTU(12) is set =0.

MSTU(32) : (I) number of entries stored with PYEDIT(21) call.

MSTU(33) : (I) if set 1 before a PYROBO call, the V vectors (in the particle range to be rotated/boosted) are set 0 before the rotation/boost. MSTU(33) is set back to 0 in the PYROBO call.

MSTU(70) : (D=0) the number of lines consisting only of equal signs (======) that are inserted in the event listing obtained with PYLIST(1), PYLIST(2) or PYLIST(3), so as to distinguish different sections of the event record on output. At most 10 such lines can be inserted; see MSTU(71) - MSTU(80). Is reset at PYEDIT calls with arguments 0–5.

MSTU(71) - MSTU(80) : line numbers below which lines consisting only of equal signs (======) are inserted in event listings. Only the first MSTU(70) of the 10 allowed positions are enabled.

15.2 Event Shapes

In this section we study general event shape variables: sphericity, thrust, Fox-Wolfram moments, and jet masses. These measures are implemented in the routines PYSPHE, PYTHRU, PYFOWO and PYJMAS, respectively.

Each event is assumed characterized by the particle four-momentum vectors $p_i = (\mathbf{p}_i, E_i)$, with $i = 1, 2, \cdots, n$ an index running over the particles of the event.

15.2.1 Sphericity

The sphericity tensor is defined as [Bjo70]

$$S^{\alpha\beta} = \frac{\sum_i p_i^\alpha p_i^\beta}{\sum_i |\mathbf{p}_i|^2} , \tag{253}$$

where $\alpha, \beta = 1, 2, 3$ corresponds to the x, y and z components. By standard diagonalization of $S^{\alpha\beta}$ one may find three eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3$, with $\lambda_1 + \lambda_2 + \lambda_3 = 1$. The sphericity of the event is then defined as

$$S = \frac{3}{2}(\lambda_2 + \lambda_3) , \tag{254}$$

so that $0 \leq S \leq 1$. Sphericity is essentially a measure of the summed $p_\perp^2$ with respect to the event axis; a 2-jet event corresponds to $S \approx 0$ and an isotropic event to $S \approx 1$.

The aplanarity A, with definition $A = \frac{3}{2}\lambda_3$, is constrained to the range $0 \leq A \leq \frac{1}{2}$. It measures the transverse momentum component out of the event plane: a planar event has $A \approx 0$ and an isotropic one $A \approx \frac{1}{2}$.

Eigenvectors $\mathbf{v}_j$ can be found that correspond to the three eigenvalues λ_j of the sphericity tensor. The $\mathbf{v}_1$ one is called the sphericity axis (or event axis, if it is clear from the context that sphericity has been used), while the sphericity event plane is spanned by $\mathbf{v}_1$ and $\mathbf{v}_2$.

The sphericity tensor is quadratic in particle momenta. This means that the sphericity value is changed if one particle is split up into two collinear ones which share the original momentum. Thus sphericity is not an infrared safe quantity in QCD perturbation theory. A useful generalization of the sphericity tensor is

$$S^{(r)\alpha\beta} = \frac{\sum_i |\mathbf{p}_i|^{r-2} p_i^\alpha p_i^\beta}{\sum_i |\mathbf{p}_i|^r} \,, \tag{255}$$

where r is the power of the momentum dependence. While $r = 2$ thus corresponds to sphericity, $r = 1$ corresponds to linear measures calculable in perturbation theory [Par78]:

$$S^{(1)\alpha\beta} = \frac{\sum_i \frac{p_i^\alpha p_i^\beta}{|\mathbf{p}_i|}}{\sum_i |\mathbf{p}_i|} \,. \tag{256}$$

Eigenvalues and eigenvectors may be defined exactly as before, and therefore also equivalents of S and A. These have no standard names; we may call them linearized sphericity $S_{\rm lin}$ and linearized aplanarity $A_{\rm lin}$. Quantities derived from the linear matrix that are standard in the literature are instead the combinations [Ell81]

$$C = 3(\lambda_1\lambda_2 + \lambda_1\lambda_3 + \lambda_2\lambda_3) \,, \tag{257}$$

$$D = 27\lambda_1\lambda_2\lambda_3 \,. \tag{258}$$

Each of these is constrained to be in the range between 0 and 1. Typically, C is used to measure the 3-jet structure and D the 4-jet one, since C is vanishing for a perfect 2-jet event and D is vanishing for a planar event. The C measure is related to the second Fox-Wolfram moment (see below), $C = 1 - H_2$.

Noninteger r values may also be used, and corresponding generalized sphericity and aplanarity measures calculated. While perturbative arguments favour $r = 1$, we know that the fragmentation 'noise', e.g. from transverse momentum fluctuations, is proportionately larger for low momentum particles, and so $r > 1$ should be better for experimental event axis determinations. The use of too large an r value, on the other hand, puts all the emphasis on a few high-momentum particles, and therefore involves a loss of information. It should then come as no surprise that intermediate r values, of around 1.5, gives the best performance for event axis determinations in 2-jet events, where the theoretical meaning of the event axis is well-defined. The gain in accuracy compared with the more conventional choices $r = 2$ or $r = 1$ is rather modest, however.

15.2.2 Thrust

The quantity thrust T is defined by [Bra64]

$$T = \max_{|\mathbf{n}|=1} \frac{\sum_i |\mathbf{n} \cdot \mathbf{p}_i|}{\sum_i |\mathbf{p}_i|} , \tag{259}$$

and the thrust axis $\mathbf{v}_1$ is given by the $\mathbf{n}$ vector for which maximum is attained. The allowed range is $1/2 \leq T \leq 1$, with a 2-jet event corresponding to $T \approx 1$ and an isotropic event to $T \approx 1/2$.

In passing, we note that this is not the only definition found in the literature. The definitions agree for events studied in the c.m. frame and where all particles are detected. However, a definition like

$$T = 2 \max_{|\mathbf{n}|=1} \frac{\left| \sum_i \theta(\mathbf{n} \cdot \mathbf{p}_i)\, \mathbf{p}_i \right|}{\sum_i |\mathbf{p}_i|} = 2 \max_{\theta_i=0,1} \frac{\left| \sum_i \theta_i\, \mathbf{p}_i \right|}{\sum_i |\mathbf{p}_i|} \tag{260}$$

(where $\theta(x)$ is the step function, $\theta(x) = 1$ if $x > 0$, else $\theta(x) = 0$) gives different results than the one above if e.g. only charged particles are detected. It would even be possible to have $T > 1$; to avoid such problems, often an extra fictitious particle is introduced to balance the total momentum [Bra79].

Eq. (259) may be rewritten as

$$T = \max_{\varepsilon_i=\pm 1} \frac{\left| \sum_i \varepsilon_i\, \mathbf{p}_i \right|}{\sum_i |\mathbf{p}_i|} . \tag{261}$$

(This may also be viewed as applying eq. (260) to an event with $2n$ particles, n carrying the momenta $\mathbf{p}_i$ and n the momenta $-\mathbf{p}_i$, thus automatically balancing the momentum.) To find the thrust value and axis this way, 2^{n-1} different possibilities would have to be tested. The reduction by a factor of 2 comes from T being unchanged when all $\varepsilon_i \to -\varepsilon_i$. Therefore this approach rapidly becomes prohibitive. Other exact methods exist, which 'only' require about $4n^2$ combinations to be tried.

In the implementation in PYTHIA, a faster alternative method is used, in which the thrust axis is iterated from a starting direction $\mathbf{n}^{(0)}$ according to

$$\mathbf{n}^{(j+1)} = \frac{\sum_i \varepsilon(\mathbf{n}^{(j)} \cdot \mathbf{p}_i)\, \mathbf{p}_i}{\left| \sum_i \varepsilon(\mathbf{n}^{(j)} \cdot \mathbf{p}_i)\, \mathbf{p}_i \right|} \tag{262}$$

(where $\varepsilon(x) = 1$ for $x > 0$ and $\varepsilon(x) = -1$ for $x < 0$). It is easy to show that the related thrust value will never decrease, $T^{(j+1)} \geq T^{(j)}$. In fact, the method normally converges in 2–4 iterations. Unfortunately, this convergence need not be towards the correct thrust axis but is occasionally only towards a local maximum of the thrust function [Bra79]. We know of no foolproof way around this complication, but the danger of an error may be lowered if several different starting axes $\mathbf{n}^{(0)}$ are tried and found to agree. These $\mathbf{n}^{(0)}$ are suitably constructed from the n' (by default 4) particles with the largest momenta in the event, and the $2^{n'-1}$ starting directions $\sum_i \varepsilon_i\, \mathbf{p}_i$ constructed from these are tried in falling order of the corresponding absolute momentum values.

When a predetermined number of the starting axes have given convergence towards the same (best) thrust axis this one is accepted.

In the plane perpendicular to the thrust axis, a major [MAR79] axis and value may be defined in just the same fashion as thrust, i.e.

$$M_a = \max_{|\mathbf{n}|=1,\,\mathbf{n}\cdot\mathbf{v}_1=0} \frac{\sum_i |\mathbf{n}\cdot\mathbf{p}_i|}{\sum_i |\mathbf{p}_i|} \; . \tag{263}$$

In a plane more efficient methods can be used to find an axis than in three dimensions [Wu79], but for simplicity we use the same method as above. Finally, a third axis, the minor axis, is defined perpendicular to the thrust and major ones, and a minor value M_i is calculated just as thrust and major. The difference between major and minor is called oblateness, $O = M_a - M_i$. The upper limit on oblateness depends on the thrust value in a not so simple way. In general $O \approx 0$ corresponds to an event symmetrical around the thrust axis and high O to a planar event.

As in the case of sphericity, a generalization to arbitrary momentum dependence may easily be obtained, here by replacing the $\mathbf{p}_i$ in the formulae above by $|\mathbf{p}_i|^{r-1}\mathbf{p}_i$. This possibility is included, although so far it has not found any experimental use.

15.2.3 Fox-Wolfram moments

The Fox-Wolfram moments H_l, $l = 0,1,2,\ldots$, are defined by [Fox79]

$$H_l = \sum_{i,j} \frac{|\mathbf{p}_i|\,|\mathbf{p}_j|}{E_{\text{vis}}^2} P_l(\cos\theta_{ij}) \; , \tag{264}$$

where θ_{ij} is the opening angle between hadrons i and j and E_{vis} the total visible energy of the event. Note that also autocorrelations, $i = j$, are included. The $P_l(x)$ are the Legendre polynomials,

$$\begin{aligned}
P_0(x) &= 1 \; , \\
P_1(x) &= x \; , \\
P_2(x) &= \frac{1}{2}(3x^2 - 1) \; , \\
P_3(x) &= \frac{1}{2}(5x^3 - 3x) \; , \\
P_4(x) &= \frac{1}{8}(35x^4 - 30x^2 + 3) \; .
\end{aligned} \tag{265}$$

To the extent that particle masses may be neglected, $H_0 \equiv 1$. It is customary to normalize the results to H_0, i.e. to give $H_{l0} = H_l/H_0$. If momentum is balanced then $H_1 \equiv 0$. 2-jet events tend to give $H_l \approx 1$ for l even and ≈ 0 for l odd.

15.2.4 Jet masses

The particles of an event may be divided into two classes. For each class a squared invariant mass may be calculated, M_1^2 and M_2^2. If the assignment of particles is adjusted such that the sum $M_1^2 + M_2^2$ is minimized, the two masses thus obtained are called heavy and light jet mass, M_{H} and M_{L}. It has been shown that these quantities are well behaved in perturbation theory [Cla79]. In e^+e^- annihilation, the heavy jet mass obtains a contribution from $q\bar{q}g$ 3-jet events, whereas the light mass is non-vanishing only when 4-jet events also are included. In the c.m. frame of an event one has the limits $0 \le M_{\text{H}}^2 \le E_{\text{cm}}^2/3$.

In general, the subdivision of particles tends to be into two hemispheres, separated by a plane perpendicular to an event axis. As with thrust, it is time-consuming to find the exact solution. Different approximate strategies may therefore be used. In the program, the sphericity axis is used to perform a fast subdivision into two hemispheres, and thus into two preliminary jets. Thereafter one particle at a time is tested to determine whether the sum $M_1^2 + M_2^2$ would be decreased if that particle were to be assigned to the other jet. The procedure is stopped when no further significant change is obtained. Often the original assignment is retained as it is, i.e. the sphericity axis gives a good separation. This is not a full guarantee, since the program might get stuck in a local minimum which is not the global one.

15.3 Cluster Finding

Global event measures, like sphericity or thrust, can only be used to determine the jet axes for back-to-back 2-jet events. To determine the individual jet axes in events with three or more jets, or with two (main) jets which are not back-to-back, cluster algorithms are customarily used. In these, nearby particles are grouped together into a variable number of clusters. Each cluster has a well-defined direction, given by a suitably weighted average of the constituent particle directions.

The cluster algorithms traditionally used in e^+e^- and in pp physics differ in several respects. The former tend to be spherically symmetric, i.e. have no preferred axis in space, and normally all particles have to be assigned to some jet. The latter pick the beam axis as preferred direction, and make use of variables related to this choice, such as rapidity and transverse momentum; additionally only a fraction of all particles are assigned to jets.

This reflects a difference in the underlying physics: in pp collisions, the beam remnants found at low transverse momenta are not related to any hard processes, and therefore only provide an unwanted noise to many studies. (Of course, also hard processes may produce particles at low transverse momenta, but at a rate much less than that from soft or semi-hard processes.) Further, the kinematics of hard processes is, to a good approximation, factorized into the hard subprocess itself, which is boost invariant in rapidity, and parton-distribution effects, which determine the overall position of a hard scattering in rapidity. Hence rapidity, azimuthal angle and transverse momentum is a suitable coordinate frame to describe hard processes in.

In standard e^+e^- annihilation events, on the other hand, the hard process c.m. frame tends to be almost at rest, and the event axis is just about randomly distributed in space, i.e. with no preferred rôle for the axis defined by the incoming $e^\pm$. All particle production is initiated by and related to the hard subprocess. Some of the particles may be less easy to associate to a specific jet, but there is no compelling reason to remove any of them from consideration.

This does not mean that the separation above is always required. 2γ events in e^+e^- may have a structure with 'beam jets' and 'hard scattering' jets, for which the pp type algorithms might be well suited. Conversely, a heavy particle produced in pp collisions could profitably be studied, in its own rest frame, with e^+e^- techniques.

In the following, particles are only characterized by their three-momenta or, alternatively, their energy and direction of motion. No knowledge is therefore assumed of particle types, or even of mass and charge. Clearly, the more is known, the more sophisticated clustering algorithms can be used. The procedure then also becomes more detector-dependent, and therefore less suitable for general usage.

PYTHIA contains two cluster finding routines. `PYCLUS` is of the e^+e^- type and `PYCELL` of the pp one. Each of them allows some variations of the basic scheme.

15.3.1 Cluster finding in an e^+e^- type of environment

The usage of cluster algorithms for e^+e^- applications started in the late 1970's. A number of different approaches were proposed [Bab80], see review in [Mor98]. Of these, we will here only discuss those based on binary joining. In this kind of approach, initially each final-state

particle is considered to be a cluster. Using some distance measure, the two nearest clusters are found. If their distance is smaller than some cut-off value, the two clusters are joined into one. In this new configuration, the two clusters that are now nearest are found and joined, and so on until all clusters are separated by a distance larger than the cut-off. The clusters remaining at the end are often also called jets. Note that, in this approach, each single particle belongs to exactly one cluster. Also note that the resulting jet picture explicitly depends on the cut-off value used. Normally the number of clusters is allowed to vary from event to event, but occasionally it is more useful to have the cluster algorithm find a predetermined number of jets (like 3).

The obvious choice for a distance measure is to use squared invariant mass, i.e. for two clusters i and j to define the distance to be

$$m_{ij}^2 = (E_i + E_j)^2 - (\mathbf{p}_i + \mathbf{p}_j)^2 \ . \tag{266}$$

(Equivalently, one could have used the invariant mass as measure rather than its square; this is just a matter of convenience.) In fact, a number of people (including one of the authors) tried this measure long ago and gave up on it, since it turns out to have severe instability problems. The reason is well understood: in general, particles tend to cluster closer in invariant mass in the region of small momenta. The clustering process therefore tends to start in the center of the event, and only subsequently spread outwards to encompass also the fast particles. Rather than clustering slow particles around the fast ones (where the latter naïvely should best represent the jet directions), the invariant mass measure will tend to cluster fast particles around the slow ones.

Another instability may be seen by considering the clustering in a simple 2-jet event. By the time that clustering has reached the level of three clusters, the 'best' the clustering algorithm can possibly have achieved, in terms of finding three low-mass clusters, is to have one fast cluster around each jet, plus a third slow cluster in the middle. In the last step this third cluster would be joined with one of the fast ones, to produce two final asymmetric clusters: one cluster would contain all the slow particles, also those that visually look like belonging to the opposite jet. A simple binary joining process, with no possibility to reassign particles between clusters, is therefore not likely to be optimal.

The solution adopted [Sjö83] is to reject invariant mass as distance measure. Instead a jet is defined as a collection of particles which have a limited transverse momentum with respect to a common jet axis, and hence also with respect to each other. This picture is clearly inspired by the standard fragmentation picture, e.g. in string fragmentation. A distance measure d_{ij} between two particles (or clusters) with momenta $\mathbf{p}_i$ and $\mathbf{p}_j$ should thus not depend critically on the longitudinal momenta but only on the relative transverse momentum. A number of such measures were tried, and the one eventually selected was

$$d_{ij}^2 = \frac{1}{2}\left(|\mathbf{p}_i|\,|\mathbf{p}_j| - \mathbf{p}_i \cdot \mathbf{p}_j\right) \frac{4\,|\mathbf{p}_i|\,|\mathbf{p}_j|}{(|\mathbf{p}_i| + |\mathbf{p}_j|)^2} = \frac{4\,|\mathbf{p}_i|^2\,|\mathbf{p}_j|^2 \sin^2(\theta_{ij}/2)}{(|\mathbf{p}_i| + |\mathbf{p}_j|)^2} \ . \tag{267}$$

For small relative angle θ_{ij}, where $2\sin(\theta_{ij}/2) \approx \sin\theta_{ij}$ and $\cos\theta_{ij} \approx 1$, this measure reduces to

$$d_{ij} \approx \frac{|\mathbf{p}_i \times \mathbf{p}_j|}{|\mathbf{p}_i + \mathbf{p}_j|} \ , \tag{268}$$

where '$\times$' represents the cross product. We therefore see that d_{ij} in this limit has the simple physical interpretation as the transverse momentum of either particle with respect to the direction given by the sum of the two particle momenta. Unlike the approximate expression, however, d_{ij} does not vanish for two back-to-back particles, but is here more related to the invariant mass between them.

The basic scheme is of the binary joining type, i.e. initially each particle is assumed to be a cluster by itself. Then the two clusters with smallest relative distance d_{ij} are found and, if $d_{ij} < d_{\rm join}$, with $d_{\rm join}$ some predetermined distance, the two clusters are joined to one, i.e. their

four-momenta are added vectorially to give the energy and momentum of the new cluster. This is repeated until the distance between any two clusters is $> d_{\rm join}$. The number and momenta of these final clusters then represent our reconstruction of the initial jet configuration, and each particle is assigned to one of the clusters.

To make this scheme workable, two further ingredients are necessary, however. Firstly, after two clusters have been joined, some particles belonging to the new cluster may actually be closer to another cluster. Hence, after each joining, all particles in the event are reassigned to the closest of the clusters. For particle i, this means that the distance d_{ij} to all clusters j in the event has to be evaluated and compared. After all particles have been considered, and only then, are cluster momenta recalculated to take into account any reassignments. To save time, the assignment procedure is not iterated until a stable configuration is reached, but, since all particles are reassigned at each step, such an iteration is effectively taking place in parallel with the cluster joining. Only at the very end, when all $d_{ij} > d_{\rm join}$, is the reassignment procedure iterated to convergence — still with the possibility to continue the cluster joining if some d_{ij} should drop below $d_{\rm join}$ due to the reassignment.

Occasionally, it may occur that the reassignment step leads to an empty cluster, i.e. one to which no particles are assigned. Since such a cluster has a distance $d_{ij} = 0$ to any other cluster, it is automatically removed in the next cluster joining. However, it is possible to run the program in a mode where a minimum number of jets is to be reconstructed. If this minimum is reached with one cluster empty, the particle is found which has largest distance to the cluster it belongs to. That cluster is then split into two, namely the large-distance particle and a remainder. Thereafter the reassignment procedure is continued as before.

Secondly, the large multiplicities normally encountered means that, if each particle initially is to be treated as a separate cluster, the program will become very slow. Therefore a smaller number of clusters, for a normal e^+e^- event typically 8–12, is constructed as a starting point for the iteration above, as follows. The particle with the highest momentum is found, and thereafter all particles within a distance $d_{ij} < d_{\rm init}$ from it, where $d_{\rm init} \ll d_{\rm join}$. Together these are allowed to form a single cluster. For the remaining particles, not assigned to this cluster, the procedure is iterated, until all particles have been used up. Particles in the central momentum region, $|\mathbf{p}| < 2d_{\rm init}$ are treated separately; if their vectorial momentum sum is above $2d_{\rm init}$ they are allowed to form one cluster, otherwise they are left unassigned in the initial configuration. The value of $d_{\rm init}$, as long as reasonably small, has no physical importance, in that the same final cluster configuration will be found as if each particle initially is assumed to be a cluster by itself: the particles clustered at this step are so nearby anyway that they almost inevitably must enter the same jet; additionally the reassignment procedure allows any possible 'mistake' to be corrected in later steps of the iteration.

Thus the jet reconstruction depends on one single parameter, $d_{\rm join}$, with a clearcut physical meaning of a transverse momentum 'jet-resolution power'. Neglecting smearing from fragmentation, d_{ij} between two clusters of equal energy corresponds to half the invariant mass of the two original partons. If one only wishes to reconstruct well separated jets, a large $d_{\rm join}$ should be chosen, while a small $d_{\rm join}$ would allow the separation of close jets, at the cost of sometimes artificially dividing a single jet into two. In particular, b quark jets may here be a nuisance. The value of $d_{\rm join}$ to use for a fixed jet-resolution power in principle should be independent of the c.m. energy of events, although fragmentation effects may give a contamination of spurious extra jets that increases slowly with $E_{\rm cm}$ for fixed $d_{\rm join}$. Therefore a $d_{\rm join} = 2.5$ GeV was acceptable at PETRA/PEP, while 3–4 GeV may be better for applications at LEP and beyond.

This completes the description of the main option of the PYCLUS routine. Variations are possible. One such is to skip the reassignment step, i.e. to make use only of the simple binary joining procedure, without any possibility to reassign particles between jets. (This option is included mainly as a reference, to check how important reassignment really is.) The other main alternative is to replace the distance measure used above with the one used in the JADE algorithm [JAD86].

The JADE cluster algorithm is an attempt to save the invariant mass measure. The distance

measure is defined to be

$$y_{ij} = \frac{2E_iE_j(1-\cos\theta_{ij})}{E_{\rm vis}^2} \ . \tag{269}$$

Here $E_{\rm vis}$ is the total visible energy of the event. The usage of $E_{\rm vis}^2$ in the denominator rather than $E_{\rm cm}^2$ tends to make the measure less sensitive to detector acceptance corrections; in addition the dimensionless nature of y_{ij} makes it well suited for a comparison of results at different c.m. energies. For the subsequent discussions, this normalization will be irrelevant, however.

The y_{ij} measure is very closely related to the squared mass distance measure: the two coincide (up to the difference in normalization) if $m_i = m_j = 0$. However, consider a pair of particles or clusters with non-vanishing individual masses and a fixed pair mass. Then, the larger the net momentum of the pair, the smaller the y_{ij} measure. This somewhat tends to favour clustering of fast particles, and makes the algorithm less unstable than the one based on true invariant mass.

The successes of the JADE algorithm are well known: one obtains a good agreement between the number of partons generated on the matrix-element (or parton-shower) level and the number of clusters reconstructed from the hadrons, such that QCD aspects like the running of α_s can be studied with a small dependence on fragmentation effects. Of course, the insensitivity to fragmentation effects depends on the choice of fragmentation model. Fragmentation effects are small in the string model, but not necessarily in independent fragmentation scenarios. Although independent fragmentation in itself is not credible, this may be seen as a signal for caution.

One should note that the JADE measure still suffers from some of the diseases of the simple mass measure (without reassignments), namely that particles which go in opposite directions may well be joined into the same cluster. Therefore, while the JADE algorithm is a good way to find the number of jets, it is inferior to the standard d_{ij} measure for a determination of jet directions and energies [Bet92]. The d_{ij} measure also gives narrower jets, which agree better with the visual impression of jet structure.

Later, the 'Durham algorithm' was introduced [Cat91], which works as the JADE one but with a distance measure

$$\tilde{y}_{ij} = \frac{2\min(E_i^2, E_j^2)(1-\cos\theta_{ij})}{E_{cm}^2} \ . \tag{270}$$

Like the d_{ij} measure, this is a transverse momentum, but $\tilde{y}_{ij}$ has the geometrical interpretation as the transverse momentum of the softer particle with respect to the direction of the harder one, while d_{ij} is the transverse momentum of either particle with respect to the common direction given by the momentum vector sum. The two definitions agree when one cluster is much softer than the other, so the soft gluon exponentiation proven for the Durham measure also holds for the d_{ij} one.

The main difference therefore is that the standard `PYCLUS` option allows reassignments, while the Durham algorithm does not. The latter is therefore more easily calculable on the perturbative parton level. This point is sometimes overstressed, and one could give counterexamples why reassignments in fact may bring better agreement with the underlying perturbative level. In particular, without reassignments, one will make the recombination that seems the 'best' in the current step, even when that forces you to make 'worse' choices in subsequent steps. With reassignments, it is possible to correct for mistakes due to the too local sensitivity of a simple binary joining scheme.

15.3.2 Cluster finding in a pp type of environment

The `PYCELL` cluster finding routines is of the kind pioneered by UA1 [UA183], and commonly used in pp physics. It is based on a choice of pseudorapidity η, azimuthal angle φ and transverse momentum $p_\perp$ as the fundamental coordinates. This choice is discussed in the introduction to cluster finding above, with the proviso that the theoretically preferred true rapidity has to be replaced by pseudorapidity, to make contact with the real-life detector coordinate system.

A fix detector grid is assumed, with the pseudorapidity range $|\eta| < \eta_{\max}$ and the full azimuthal range each divided into a number of equally large bins, giving a rectangular grid. The particles of an event impinge on this detector grid. For each cell in (η,φ) space, the transverse energy (normally $\approx p_\perp$) which enters that cell is summed up to give a total cell $E_\perp$ flow.

Clearly the model remains very primitive in a number of respects, compared with a real detector. There is no magnetic field allowed for, i.e. also charged particles move in straight tracks. The dimensions of the detector are not specified; hence the positions of the primary vertex and any secondary vertices are neglected when determining which cell a particle belongs to. The rest mass of particles is not taken into account, i.e. what is used is really $p_\perp = \sqrt{p_x^2 + p_y^2}$, while in a real detector some particles would decay or annihilate, and then deposit additional amounts of energy.

To take into account the energy resolution of the detector, it is possible to smear the $E_\perp$ contents, bin by bin. This is done according to a Gaussian, with a width assumed proportional to the $\sqrt{E_\perp}$ of the bin. The Gaussian is cut off at zero and at some predetermined multiple of the unsmeared $E_\perp$, by default twice it. Alternatively, the smearing may be performed in E rather than in $E_\perp$. To find the E, it is assumed that the full energy of a cell is situated at its center, so that one can translate back and forth with $E = E_\perp \cosh\eta_{\text{center}}$.

The cell with largest $E_\perp$ is taken as a jet initiator if its $E_\perp$ is above some threshold. A candidate jet is defined to consist of all cells which are within some given radius R in the (η,φ) plane, i.e. which have $(\eta - \eta_{\text{initiator}})^2 + (\varphi - \varphi_{\text{initiator}})^2 < R^2$. Coordinates are always given with respect to the center of the cell. If the summed $E_\perp$ of the jet is above the required minimum jet energy, the candidate jet is accepted, and all its cells are removed from further consideration. If not, the candidate is rejected. The sequence is now repeated with the remaining cell of highest $E_\perp$, and so on until no single cell fulfils the jet initiator condition.

The number of jets reconstructed can thus vary from none to a maximum given by purely geometrical considerations, i.e. how many circles of radius R are needed to cover the allowed (η,φ) plane. Normally only a fraction of the particles are assigned to jets.

One could consider to iterate the jet assignment process, using the $E_\perp$-weighted center of a jet to draw a new circle of radius R. In the current algorithm there is no such iteration step. For an ideal jet assignment it would also be necessary to improve the treatment when two jet circles partially overlap.

A final technical note. A natural implementation of a cell finding algorithm is based on having a two-dimensional array of $E_\perp$ values, with dimensions to match the detector grid. Very often most of the cells would then be empty, in particular for low-multiplicity events in fine-grained calorimeters. Our implementation is somewhat atypical, since cells are only reserved space (contents and position) when they are shown to be non-empty. This means that all non-empty cells have to be looped over to find which are within the required distance R of a potential jet initiator. The algorithm is therefore faster than the ordinary kind if the average cell occupancy is low, but slower if it is high.

15.4 Event Statistics

All the event-analysis routines above are defined on an event-by-event basis. Once found, the quantities are about equally often used to define inclusive distributions as to select specific classes of events for continued study. For instance, the thrust routine might be used either to find the inclusive T distribution or to select events with $T < 0.9$. Other measures, although still defined for the individual event, only make sense to discuss in terms of averages over many events. A small set of such measures is found in `PYTABU`. This routine has to be called once after each event to accumulate statistics, and once in the end to print the final tables. Of course, among the wealth of possibilities imaginable, the ones collected here are only a small sample, selected because the authors at some point have found a use for them.

15.4.1 Multiplicities

Three options are available to collect information on multiplicities in events. One gives the flavour content of the final state in hard interaction processes, e.g. the relative composition of $d\bar{d}/u\bar{u}/s\bar{s}/c\bar{c}/b\bar{b}$ in e^+e^- annihilation events. Additionally it gives the total parton multiplicity distribution at the end of parton showering. Another gives the inclusive rate of all the different particles produced in events, either as intermediate resonances or as final-state particles. The number is subdivided into particles produced from fragmentation (primary particles) and those produced in decays (secondary particles).

The third option tabulates the rate of exclusive final states, after all allowed decays have occurred. Since only events with up to 8 final-state particles are analyzed, this is clearly not intended for the study of complete high-energy events. Rather the main application is for an analysis of the decay modes of a single particle. For instance, the decay data for D mesons is given in terms of channels that also contain unstable particles, such as ρ and η, which decay further. Therefore a given final state may receive contributions from several tabulated decay channels; e.g. $K\pi\pi$ from $K^*\pi$ and $K\rho$, and so on.

15.4.2 Energy-Energy Correlation

The Energy-Energy Correlation is defined by [Bas78]

$$\mathrm{EEC}(\theta) = \sum_{i<j} \frac{2E_iE_j}{E_{\mathrm{vis}}^2}\,\delta(\theta-\theta_{ij}) \;, \tag{271}$$

and its Asymmetry by

$$\mathrm{EECA}(\theta) = \mathrm{EEC}(\pi-\theta) - \mathrm{EEC}(\theta) \;. \tag{272}$$

Here θ_{ij} is the opening angle between the two particles i and j, with energies E_i and E_j. In principle, normalization should be to E_{cm}, but if not all particles are detected it is convenient to normalize to the total visible energy E_{vis}. Taking into account the autocorrelation term $i=j$, the total EEC in an event then is unity. The δ function peak is smeared out by the finite bin width $\Delta\theta$ in the histogram, i.e., it is replaced by a contribution $1/\Delta\theta$ to the bin which contains θ_{ij}.

The formulae above refer to an individual event, and are to be averaged over all events to suppress statistical fluctuations, and obtain smooth functions of θ.

15.4.3 Factorial moments

Factorial moments may be used to search for intermittency in events [Bia86]. The whole field has been much studied in past years, and a host of different measures have been proposed. We only implement one of the original prescriptions.

To calculate the factorial moments, the full rapidity (or pseudorapidity) and azimuthal ranges are subdivided into bins of successively smaller size, and the multiplicity distributions in bins is studied. The program calculates pseudorapidity with respect to the z axis; if desired, one could first find an event axis, e.g. the sphericity or thrust axis, and subsequently rotate the event to align this axis with the z direction.

The full rapidity range $|y| < y_{\max}$ (or pseudorapidity range $|\eta| < \eta_{\max}$) and azimuthal range $0 < \varphi < 2\pi$ are subdivided into m_y and m_φ equally large bins. In fact, the whole analysis is performed thrice: once with $m_\varphi = 1$ and the y (or η) range gradually divided into 1, 2, 4, 8, 16, 32, 64, 128, 256 and 512 bins, once with $m_y = 1$ and the φ range subdivided as above, and finally once with $m_y = m_\varphi$ according to the same binary sequence. Given the multiplicity n_j in bin j, the i:th factorial moment is defined by

$$F_i = (m_y m_\varphi)^{i-1} \sum_j \frac{n_j(n_j-1)\cdots(n_j-i+1)}{n(n-1)\cdots(n-i+1)} \;. \tag{273}$$

Here $n = \sum_j n_j$ is the total multiplicity of the event within the allowed y (or η) limits. The calculation is performed for the second through the fifth moments, i.e. F_2 through F_5.

The F_i as given here are defined for the individual event, and have to be averaged over many events to give a reasonably smooth behaviour. If particle production is uniform and uncorrelated according to Poissonian statistics, one expects $\langle F_i \rangle \equiv 1$ for all moments and all bin sizes. If, on the other hand, particles are locally clustered, factorial moments should increase when bins are made smaller, down to the characteristic dimensions of the clustering.

15.5 Routines and Common Block Variables

The six routines PYSPHE, PYTHRU, PYCLUS, PYCELL, PYJMAS and PYFOWO give you the possibility to find some global event shape properties. The routine PYTABU performs a statistical analysis of a number of different quantities like particle content, factorial moments and the energy–energy correlation.

Note that, by default, all remaining partons/particles except neutrinos are used in the analysis. Neutrinos may be included with MSTU(41)=1. Also note that axes determined are stored in PYJETS, but are not proper four-vectors and, as a general rule (with some exceptions), should therefore not be rotated or boosted.

```
CALL PYSPHE(SPH,APL)
```

Purpose: to diagonalize the momentum tensor, i.e. find the eigenvalues $\lambda_1 > \lambda_2 > \lambda_3$, with sum unity, and the corresponding eigenvectors.
Momentum power dependence is given by PARU(41); default corresponds to sphericity, PARU(41)=1. gives measures linear in momenta. Which particles (or partons) are used in the analysis is determined by the MSTU(41) value.

SPH : $\frac{3}{2}(\lambda_2 + \lambda_3)$, i.e. sphericity (for PARU(41)=2.).
= -1. : analysis not performed because event contained less than two particles (or two exactly back-to-back particles, in which case the two transverse directions would be undefined).

APL : $\frac{3}{2}\lambda_3$, i.e. aplanarity (for PARU(41)=2.).
= -1. : as SPH=-1.

Remark: the lines N+1 through N+3 (N-2 through N for MSTU(43)=2) in PYJETS will, after a call, contain the following information:
K(N+i,1) = 31;
K(N+i,2) = 95;
K(N+i,3) : i, the axis number, $i = 1,2,3$;
K(N+i,4), K(N+i,5) = 0;
P(N+i,1) - P(N+i,3) : the i'th eigenvector, x, y and z components;
P(N+i,4) : λ_i, the i'th eigenvalue;
P(N+i,5) = 0;
V(N+i,1) - V(N+i,5) = 0.
Also, the number of particles used in the analysis is given in MSTU(62).

```
CALL PYTHRU(THR,OBL)
```

Purpose: to find the thrust, major and minor axes and corresponding projected momentum quantities, in particular thrust and oblateness. The performance of the program is affected by MSTU(44), MSTU(45), PARU(42) and PARU(48). In particular, PARU(42) gives the momentum dependence, with the default value =1 corresponding to linear dependence. Which particles (or partons) are used in the analysis is determined by the MSTU(41) value.

THR : thrust (for PARU(42)=1.).

= -1. : analysis not performed because event contained less than two particles.

= -2. : remaining space in PYJETS (partly used as working area) not large enough to allow analysis.

OBL : oblateness (for PARU(42)=1.).

= -1., -2. : as for THR.

Remark: the lines N+1 through N+3 (N-2 through N for MSTU(43)=2) in PYJETS will, after a call, contain the following information:

K(N+i,1) = 31;

K(N+i,2) = 96;

K(N+i,3) : i, the axis number, $i = 1,2,3$;

K(N+i,4), K(N+i,5) = 0;

P(N+i,1) - P(N+i,3) : the thrust, major and minor axis, respectively, for $i = 1,2$ and 3;

P(N+i,4) : corresponding thrust, major and minor value;

P(N+i,5) = 0;

V(N+i,1) - V(N+i,5) = 0.

Also, the number of particles used in the analysis is given in MSTU(62).

```
CALL PYCLUS(NJET)
```

Purpose: to reconstruct an arbitrary number of jets using a cluster analysis method based on particle momenta.

Three different distance measures are available, see section 15.3. The choice is controlled by MSTU(46). The distance scale $d_{\rm join}$, above which two clusters may not be joined, is normally given by PARU(44). In general, $d_{\rm join}$ may be varied to describe different 'jet-resolution powers'; the default value, 2.5 GeV, is fairly well suited for e^+e^- physics at 30–40 GeV. With the alternative mass distance measure, PARU(44) can be used to set the absolute maximum cluster mass, or PARU(45) to set the scaled one, i.e. in $y = m^2/E_{\rm cm}^2$, where $E_{\rm cm}$ is the total invariant mass of the particles being considered.

It is possible to continue the cluster search from the configuration already found, with a new higher $d_{\rm join}$ scale, by selecting MSTU(48) properly. In MSTU(47) one can also require a minimum number of jets to be reconstructed; combined with an artificially large $d_{\rm join}$ this can be used to reconstruct a predetermined number of jets. Which particles (or partons) are used in the analysis is determined by the MSTU(41) value, whereas assumptions about particle masses is given by MSTU(42). The parameters PARU(43) and PARU(48) regulate more technical details (for events at high energies and large multiplicities, however, the choice of a larger PARU(43) may be necessary to obtain reasonable reconstruction times).

NJET : the number of clusters reconstructed.

= -1 : analysis not performed because event contained less than MSTU(47) (normally 1) particles, or analysis failed to reconstruct the requested number of jets.

= -2 : remaining space in PYJETS (partly used as working area) not large enough to allow analysis.

Remark: if the analysis does not fail, further information is found in MSTU(61) - MSTU(63) and PARU(61) - PARU(63). In particular, PARU(61) contains the invariant mass for the system analyzed, i.e. the number used in determining the denominator of $y = m^2/E_{\rm cm}^2$. PARU(62) gives the generalized thrust, i.e. the sum of (absolute values of) cluster momenta divided by the sum of particle momenta (roughly the same as multicity [Bra79]). PARU(63) gives the minimum distance d (in $p_\perp$ or m) between two clusters in the final cluster configuration, 0 in case of only one cluster.

Further, the lines N+1 through N+NJET (N-NJET+1 through N for MSTU(43)=2) in PYJETS will, after a call, contain the following information:
K(N+i,1) = 31;
K(N+i,2) = 97;
K(N+i,3) : i, the jet number, with the jets arranged in falling order of absolute momentum;
K(N+i,4) : the number of particles assigned to jet i;
K(N+i,5) = 0;
P(N+i,1) - P(N+i,5) : momentum, energy and invariant mass of jet i;
V(N+i,1) - V(N+i,5) = 0.
Also, for a particle which was used in the analysis, K(I,4)= i, where I is the particle number and i the number of the jet it has been assigned to. Undecayed particles not used then have K(I,4)=0. An exception is made for lines with K(I,1)=3 (which anyhow are not normally interesting for cluster search), where the colour-flow information stored in K(I,4) is left intact.
MSTU(3) is only set equal to the number of jets for positive NJET and MSTU(43)=1.

```
CALL PYCELL(NJET)
```

Purpose: to provide a simpler cluster routine more in line with what is currently used in the study of high-$p_\perp$ collider events.
A detector is assumed to stretch in pseudorapidity between -PARU(51) and +PARU(51) and be segmented in MSTU(51) equally large η (pseudorapidity) bins and MSTU(52) φ (azimuthal) bins. Transverse energy $E_\perp$ for undecayed entries are summed up in each bin. For MSTU(53) non-zero, the energy is smeared by calorimetric resolution effects, cell by cell. This is done according to a Gaussian distribution; if MSTU(53)=1 the standard deviation for the $E_\perp$ is PARU(55)$\times\sqrt{E_\perp}$, if MSTU(53)=2 the standard deviation for the E is PARU(55)$\times\sqrt{E}$, $E_\perp$ and E expressed in GeV. The Gaussian is cut off at 0 and at a factor PARU(56) times the correct $E_\perp$ or E. Cells with an $E_\perp$ below a given threshold PARU(58) are removed from further consideration; by default PARU(58)=0. and thus all cells are kept.
All bins with $E_\perp >$PARU(52) are taken to be possible initiators of jets, and are tried in falling $E_\perp$ sequence to check whether the total $E_\perp$ summed over cells no more distant than PARU(54) in $\sqrt{(\Delta\eta)^2+(\Delta\varphi)^2}$ exceeds PARU(53). If so, these cells define one jet, and are removed from further consideration. Contrary to PYCLUS, not all particles need be assigned to jets. Which particles (or partons) are used in the analysis is determined by the MSTU(41) value.

NJET : the number of jets reconstructed (may be 0).
= -2 : remaining space in PYJETS (partly used as working area) not large enough to allow analysis.

Remark: the lines N+1 through N+NJET (N-NJET+1 through N for MSTU(43)=2) in PYJETS will, after a call, contain the following information:
K(N+i,1) = 31;
K(N+i,2) = 98;
K(N+i,3) : i, the jet number, with the jets arranged in falling order in $E_\perp$;
K(N+i,4) : the number of particles assigned to jet i;
K(N+i,5) = 0;
V(N+i,1) - V(N+i,5) = 0.
Further, for MSTU(54)=1
P(N+i,1), P(N+i,2) = position in η and φ of the center of the jet initiator cell, i.e. geometrical center of jet;
P(N+i,3), P(N+i,4) = position in η and φ of the $E_\perp$-weighted center of the jet,

i.e. the center of gravity of the jet;
`P(N+i,5)` = sum $E_\perp$ of the jet;
while for `MSTU(54)=2`
`P(N+i,1) - P(N+i,5)` : the jet momentum vector, constructed from the summed $E_\perp$ and the η and φ of the $E_\perp$-weighted center of the jet as
$(p_x, p_y, p_z, E, m) = E_\perp(\cos\varphi, \sin\varphi, \sinh\eta, \cosh\eta, 0)$;
and for `MSTU(54)=3`
`P(N+i,1) - P(N+i,5)` : the jet momentum vector, constructed by adding vectorially the momentum of each cell assigned to the jet, assuming that all the $E_\perp$ was deposited at the center of the cell, and with the jet mass in `P(N+i,5)` calculated from the summed E and $\mathbf{p}$ as $m^2 = E^2 - p_x^2 - p_y^2 - p_z^2$.
Also, the number of particles used in the analysis is given in `MSTU(62)`, and the number of cells hit in `MSTU(63)`.
`MSTU(3)` is only set equal to the number of jets for positive `NJET` and `MSTU(43)=1`.

```
CALL PYJMAS(PMH,PML)
```

Purpose: to reconstruct high and low jet mass of an event. A simplified algorithm is used, wherein a preliminary division of the event into two hemispheres is done transversely to the sphericity axis. Then one particle at a time is reassigned to the other hemisphere if that reduces the sum of squares of the two jet masses, $m_{\rm H}^2 + m_{\rm L}^2$. The procedure is stopped when no further significant change (see `PARU(48)`) is obtained. Often, the original assignment is retained as it is. Which particles (or partons) are used in the analysis is determined by the `MSTU(41)` value, whereas assumptions about particle masses is given by `MSTU(42)`.

`PMH` : heavy jet mass (in GeV).
`= -2.` : remaining space in `PYJETS` (partly used as working area) not large enough to allow analysis.

`PML` : light jet mass (in GeV).
`= -2.` : as for `PMH=-2.`

Remark: After a successful call, `MSTU(62)` contains the number of particles used in the analysis, and `PARU(61)` the invariant mass of the system analyzed. The latter number is helpful in constructing scaled jet masses.

```
CALL PYFOWO(H10,H20,H30,H40)
```

Purpose: to do an event analysis in terms of the Fox-Wolfram moments. The moments H_i are normalized to the lowest one, H_0. Which particles (or partons) are used in the analysis is determined by the `MSTU(41)` value.

`H10` : H_1/H_0. Is $= 0$ if momentum is balanced.
`H20` : H_2/H_0.
`H30` : H_3/H_0.
`H40` : H_4/H_0.

Remark: the number of particles used in the analysis is given in `MSTU(62)`.

```
CALL PYTABU(MTABU)
```

Purpose: to provide a number of event-analysis options which can be be used on each new event, with accumulated statistics to be written out on request. When errors are quoted, these refer to the uncertainty in the average value for the event sample as a whole, rather than to the spread of the individual events, i.e. errors decrease

like one over the square root of the number of events analyzed. For a correct use of PYTABU, it is not permissible to freely mix generation and analysis of different classes of events, since only one set of statistics counters exists. A single run may still contain sequential 'subruns', between which statistics is reset. Whenever an event is analyzed, the number of particles/partons used is given in MSTU(62).

MTABU : determines which action is to be taken. Generally, a last digit equal to 0 indicates that the statistics counters for this option is to be reset; since the counters are reset (by DATA statements) at the beginning of a run, this is not used normally. Last digit 1 leads to an analysis of current event with respect to the desired properties. Note that the resulting action may depend on how the event generated has been rotated, boosted or edited before this call. The statistics accumulated is output in tabular form with last digit 2, while it is dumped in the PYJETS common block for last digit 3. The latter option may be useful for interfacing to graphics output.

Warning: this routine cannot be used on weighted events, i.e. in the statistics calculation all events are assumed to come with the same weight.

= 10 : statistics on parton multiplicity is reset.

= 11 : the parton content of the current event is analyzed, classified according to the flavour content of the hard interaction and the total number of partons. The flavour content is assumed given in MSTU(161) and MSTU(162); these are automatically set e.g. in PYEEVT and PYEVNT calls.

= 12 : gives a table on parton multiplicity distribution.

= 13 : stores the parton multiplicity distribution of events in /PYJETS/, using the following format:
N = total number of different channels found;
K(I,1) = 32;
K(I,2) = 99;
K(I,3), K(I,4) = the two flavours of the flavour content;
K(I,5) = total number of events found with flavour content of K(I,3) and K(I,4);
P(I,1) - P(I,5) = relative probability to find given flavour content and a total of 1, 2, 3, 4 or 5 partons, respectively;
V(I,1) - V(I,5) = relative probability to find given flavour content and a total of 6–7, 8–10, 11–15, 16–25 or above 25 partons, respectively.
In addition, MSTU(3)=1 and
K(N+1,1) = 32;
K(N+1,2) = 99;
K(N+1,5) = number of events analyzed.

= 20 : statistics on particle content is reset.

= 21 : the particle/parton content of the current event is analyzed, also for particles which have subsequently decayed and partons which have fragmented (unless this has been made impossible by a preceding PYEDIT call). Particles are subdivided into primary and secondary ones, the main principle being that primary particles are those produced in the fragmentation of a string, while secondary come from decay of other particles.

= 22 : gives a table of particle content in events.

= 23 : stores particle content in events in /PYJETS/, using the following format:
N = number of different particle species found;
K(I,1) = 32;
K(I,2) = 99;
K(I,3) = particle KF code;
K(I,5) = total number of particles and antiparticles of this species;
P(I,1) = average number of primary particles per event;
P(I,2) = average number of secondary particles per event;

P(I,3) = average number of primary antiparticles per event;
P(I,4) = average number of secondary antiparticles per event;
P(I,5) = average total number of particles or antiparticles per event.
In addition, MSTU(3)=1 and
K(N+1,1) = 32;
K(N+1,2) = 99;
K(N+1,5) = number of events analyzed;
P(N+1,1) = average primary multiplicity per event;
P(N+1,2) = average final multiplicity per event;
P(N+1,3) = average charged multiplicity per event.

= 30 : statistics on factorial moments is reset.

= 31 : analyzes the factorial moments of the multiplicity distribution in different bins of rapidity and azimuth. Which particles (or partons) are used in the analysis is determined by the MSTU(41) value. The selection between usage of true rapidity, pion rapidity or pseudorapidity is regulated by MSTU(42). The z axis is assumed to be event axis; if this is not desirable find an event axis e.g. with PYSPHE or PYTHRU and use PYEDIT(31). Maximum (pion-, pseudo-) rapidity, which sets the limit for the rapidity plateau or the experimental acceptance, is given by PARU(57).

= 32 : prints a table of the first four factorial moments for various bins of pseudorapidity and azimuth. The moments are properly normalized so that they would be unity (up to statistical fluctuations) for uniform and uncorrelated particle production according to Poissonian statistics, but increasing for decreasing bin size in case of 'intermittent' behaviour. The error on the average value is based on the actual statistical sample (i.e. does not use any assumptions on the distribution to relate errors to the average values of higher moments). Note that for small bin sizes, where the average multiplicity is small and the factorial moment therefore only very rarely is non-vanishing, moment values may fluctuate wildly and the errors given may be too low.

= 33 : stores the factorial moments in /PYJETS/, using the format:
N = 30, with I = $i = 1$–10 corresponding to results for slicing the rapidity range in 2^{i-1} bins, I = $i = 11$–20 to slicing the azimuth in 2^{i-11} bins, and I = $i = 21$–30 to slicing both rapidity and azimuth, each in 2^{i-21} bins;
K(I,1) = 32;
K(I,2) = 99;
K(I,3) = number of bins in rapidity;
K(I,4) = number of bins in azimuth;
P(I,1) = rapidity bin size;
P(I,2) - P(I,5) = $\langle F_2\rangle$–$\langle F_5\rangle$, i.e. mean of second, third, fourth and fifth factorial moment;
V(I,1) = azimuthal bin size;
V(I,2) - V(I,5) = statistical errors on $\langle F_2\rangle$–$\langle F_5\rangle$.
In addition, MSTU(3) = 1 and
K(31,1) = 32;
K(31,2) = 99;
K(31,5) = number of events analyzed.

= 40 : statistics on energy–energy correlation is reset.

= 41 : the energy–energy correlation EEC of the current event is analyzed. Which particles (or partons) are used in the analysis is determined by the MSTU(41) value. Events are assumed given in their c.m. frame. The weight assigned to a pair i and j is $2E_iE_j/E_{\rm vis}^2$, where $E_{\rm vis}$ is the sum of energies of all analyzed particles in the event. Energies are determined from the momenta of particles, with mass determined according to the MSTU(42) value. Statistics

is accumulated for the relative angle θ_{ij}, ranging between 0 and 180 degrees, subdivided into 50 bins.

`= 42 :` prints a table of the energy–energy correlation EEC and its asymmetry EECA, with errors. The definition of errors is not unique. In our approach each event is viewed as one observation, i.e. an EEC and EECA distribution is obtained by summing over all particle pairs of an event, and then the average and spread of this event-distribution is calculated in the standard fashion. The quoted error is therefore inversely proportional to the square root of the number of events. It could have been possible to view each single particle pair as one observation, which would have given somewhat lower errors, but then one would also be forced to do a complicated correction procedure to account for the pairs in an event not being uncorrelated (two hard jets separated by a given angle typically corresponds to several pairs at about that angle). Note, however, that in our approach the squared error on an EECA bin is smaller than the sum of the squares of the errors on the corresponding EEC bins (as it should be). Also note that it is not possible to combine the errors of two nearby bins by hand from the information given, since nearby bins are correlated (again a trivial consequence of the presence of jets).

`= 43 :` stores the EEC and EECA in `/PYJETS/`, using the format:
`N` = 25;
`K(I,1)` = 32;
`K(I,2)` = 99;
`P(I,1)` = EEC for angles between `I-1` and `I`, in units of 3.6°;
`P(I,2)` = EEC for angles between `50-I` and `51-I`, in units of 3.6°;
`P(I,3)` = EECA for angles between `I-1` and `I`, in units of 3.6°;
`P(I,4), P(I,5) :` lower and upper edge of angular range of bin `I`, expressed in radians;
`V(I,1) - V(I,3) :` errors on the EEC and EECA values stored in `P(I,1) - P(I,3)` (see `=42` for comments);
`V(I,4), V(I,5) :` lower and upper edge of angular range of bin `I`, expressed in degrees.
In addition, `MSTU(3)=1` and
`K(26,1)` = 32;
`K(26,2)` = 99;
`K(26,5)` = number of events analyzed.

`= 50 :` statistics on complete final states is reset.

`= 51 :` analyzes the particle content of the final state of the current event record. During the course of the run, statistics is thus accumulated on how often different final states appear. Only final states with up to 8 particles are analyzed, and there is only reserved space for up to 200 different final states. Most high energy events have multiplicities far above 8, so the main use for this tool is to study the effective branching ratios obtained with a given decay model for e.g. charm or bottom hadrons. Then `PY1ENT` may be used to generate one decaying particle at a time, with a subsequent analysis by `PYTABU`. Depending on at what level this studied is to be carried out, some particle decays may be switched off, like π^0.

`= 52 :` gives a list of the (at most 200) channels with up to 8 particles in the final state, with their relative branching ratio. The ordering is according to multiplicity, and within each multiplicity according to an ascending order of `KF` codes. The `KF` codes of the particles belonging to a given channel are given in descending order.

`= 53 :` stores the final states and branching ratios found in `/PYJETS/`, using the format:

N = number of different explicit final states found (at most 200);
K(I,1) = 32;
K(I,2) = 99;
K(I,5) = multiplicity of given final state, a number between 1 and 8;
P(I,1) - P(I,5), V(I,1) - V(I,3) : the KF codes of the up to 8 particles of the given final state, converted to real numbers, with trailing zeroes for positions not used;
V(I,5) : effective branching ratio for the given final state.
In addition, MSTU(3)=1 and
K(N+1,1) = 32;
K(N+1,2) = 99;
K(N+1,5) = number of events analyzed;
V(N+1,5) = summed branching ratio for finals states not given above, either because they contained more than 8 particles or because all 200 channels have been used up.

```
COMMON/PYDAT1/MSTU(200),PARU(200),MSTJ(200),PARJ(200)
```

Purpose: to give access to a number of status codes and parameters which regulate the performance of fragmentation and event analysis routines. Most parameters are described in section 14.3; here only those related to the event-analysis routines are described.

MSTU(41) : (D=2) partons/particles used in the event-analysis routines PYSPHE, PYTHRU, PYCLUS, PYCELL, PYJMAS, PYFOWO and PYTABU (PYTABU(11) excepted).
- = 1 : all partons/particles that have not fragmented/decayed.
- = 2 : ditto, with the exception of neutrinos and unknown particles.
- = 3 : only charged, stable particles, plus any partons still not fragmented.

MSTU(42) : (D=2) assumed particle masses, used in calculating energies $E^2 = \mathbf{p}^2 + m^2$, as subsequently used in PYCLUS, PYJMAS and PYTABU (in the latter also for pseudorapidity, pion rapidity or true rapidity selection).
- = 0 : all particles are assumed massless.
- = 1 : all particles, except the photon, are assumed to have the charged pion mass.
- = 2 : the true masses are used.

MSTU(43) : (D=1) storing of event-analysis information (mainly jet axes), in PYSPHE, PYTHRU, PYCLUS and PYCELL.
- = 1 : stored after the event proper, in positions N+1 through N+MSTU(3). If several of the routines are used in succession, all but the latest information is overwritten.
- = 2 : stored with the event proper, i.e. at the end of the event listing, with N updated accordingly. If several of the routines are used in succession, all the axes determined are available.

MSTU(44) : (D=4) is the number of the fastest (i.e. with largest momentum) particles used to construct the (at most) 10 most promising starting configurations for the thrust axis determination.

MSTU(45) : (D=2) is the number of different starting configurations above, which have to converge to the same (best) value before this is accepted as the correct thrust axis.

MSTU(46) : (D=1) distance measure used for the joining of clusters in PYCLUS.
- = 1 : d_{ij}, i.e. approximately relative transverse momentum. Anytime two clusters have been joined, particles are reassigned to the cluster they now are closest to. The distance cut-off $d_{\rm join}$ is stored in PARU(44).
- = 2 : distance measure as in =1, but particles are never reassigned to new jets.
- = 3 : JADE distance measure y_{ij}, but with dimensions to correspond approximately to total invariant mass. Particles may never be reassigned between clusters.

The distance cut-off $m_{\min}$ is stored in PARU(44).

= 4 : as =3, but a scaled JADE distance y_{ij} is used instead of m_{ij}. The distance cut-off $y_{\min}$ is stored in PARU(45).

= 5 : Durham distance measure $\tilde{y}_{ij}$, but with dimensions to correspond approximately to transverse momentum. Particles may never be reassigned between clusters. The distance cut-off $p_{\perp\min}$ is stored in PARU(44).

= 6 : as =5, but a scaled Durham distance $\tilde{y}_{ij}$ is used instead of $p_{\perp ij}$. The distance cut-off $\tilde{y}_{\min}$ is stored in PARU(45).

MSTU(47) : (D=1) the minimum number of clusters to be reconstructed by PYCLUS.

MSTU(48) : (D=0) mode of operation of the PYCLUS routine.

= 0 : the cluster search is started from scratch.

= 1 : the clusters obtained in a previous cluster search on the same event (with MSTU(48)=0) are to be taken as the starting point for subsequent cluster joining. For this call to have any effect, the joining scale in PARU(44) or PARU(45) must have been changed. If the event record has been modified after the last PYCLUS call, or if any other cluster search parameter setting has been changed, the subsequent result is unpredictable.

MSTU(51) : (D=25) number of pseudorapidity bins that the range between -PARU(51) and +PARU(51) is divided into to define cell size for PYCELL.

MSTU(52) : (D=24) number of azimuthal bins, used to define the cell size for PYCELL.

MSTU(53) : (D=0) smearing of correct energy, imposed cell-by-cell in PYCELL, to simulate calorimeter resolution effects.

= 0 : no smearing.

= 1 : the transverse energy in a cell, $E_\perp$, is smeared according to a Gaussian distribution with standard deviation PARU(55)$\times\sqrt{E_\perp}$, where $E_\perp$ is given in GeV. The Gaussian is cut off so that $0 < E_{\perp\text{smeared}} <$PARU(56)$\times E_{\perp\text{true}}$.

= 2 : as =1, but it is the energy E rather than the transverse energy $E_\perp$ that is smeared.

MSTU(54) : (D=1) form for presentation of information about reconstructed clusters in PYCELL, as stored in PYJETS according to the MSTU(43) value.

= 1 : the P vector in each line contains η and φ for the geometric origin of the jet, η and φ for the weighted center of the jet, and jet $E_\perp$, respectively.

= 2 : the P vector in each line contains a massless four-vector giving the direction of the jet, obtained as
$(p_x, p_y, p_z, E, m) = E_\perp(\cos\varphi, \sin\varphi, \sinh\eta, \cosh\eta, 0)$,
where η and φ give the weighted center of a jet and $E_\perp$ its transverse energy.

= 3 : the P vector in each line contains a massive four-vector, obtained by adding the massless four-vectors of all cells that form part of the jet, and calculating the jet mass from $m^2 = E^2 - p_x^2 - p_y^2 - p_z^2$. For each cell, the total $E_\perp$ is summed up, and then translated into a massless four-vector assuming that all the $E_\perp$ was deposited in the center of the cell.

MSTU(61) : (I) first entry for storage of event-analysis information in last event analyzed with PYSPHE, PYTHRU, PYCLUS or PYCELL.

MSTU(62) : (R) number of particles/partons used in the last event analysis with PYSPHE, PYTHRU, PYCLUS, PYCELL, PYJMAS, PYFOWO or PYTABU.

MSTU(63) : (R) in a PYCLUS call, the number of preclusters constructed in order to speed up analysis (should be equal to MSTU(62) if PARU(43)=0.). In a PYCELL call, the number of cells hit.

MSTU(161), MSTU(162) : hard flavours involved in current event, as used in an analysis with PYTABU(11). Either or both may be set 0, to indicate the presence of one or none hard flavours in event. Is normally set by high-level routines, like PYEEVT or PYEVNT, but can also be set by you.

PARU(41) : (D=2.) power of momentum-dependence in PYSPHE, default corresponds to sphericity, =1. to linear event measures.

PARU(42) : (D=1.) power of momentum-dependence in PYTHRU, default corresponds to thrust.

PARU(43) : (D=0.25 GeV) maximum distance $d_{\rm init}$ allowed in PYCLUS when forming starting clusters used to speed up reconstruction. The meaning of the parameter is in $p_\perp$ for MSTU(46)≤ 2 or ≥ 5 and in m else. If =0., no preclustering is obtained. If chosen too large, more joining may be generated at this stage than is desirable. The main application is at high energies, where some speedup is imperative, and the small details are not so important anyway.

PARU(44) : (D=2.5 GeV) maximum distance $d_{\rm join}$, below which it is allowed to join two clusters into one in PYCLUS. Is used for MSTU(46)≤ 3 and =5, i.e. both for $p_\perp$ and mass distance measure.

PARU(45) : (D=0.05) maximum distance $y_{\rm join} = m^2/E^2_{\rm vis}$ or ditto with $m^2 \to p^2_\perp$, below which it is allowed to join two clusters into one in PYCLUS for MSTU(46)=4, =6.

PARU(48) : (D=0.0001) convergence criterion for thrust (in PYTHRU) or generalized thrust (in PYCLUS), or relative change of $m^2_{\rm H} + m^2_{\rm L}$ (in PYJMAS), i.e. when the value changes by less than this amount between two iterations the process is stopped.

PARU(51) : (D=2.5) defines maximum absolute pseudorapidity used for detector assumed in PYCELL.

PARU(52) : (D=1.5 GeV) gives minimum $E_\perp$ for a cell to be considered as a potential jet initiator by PYCELL.

PARU(53) : (D=7.0 GeV) gives minimum summed $E_\perp$ for a collection of cells to be accepted as a jet.

PARU(54) : (D=1.) gives the maximum distance in $R = \sqrt{(\Delta\eta)^2 + (\Delta\varphi)^2}$ from cell initiator when grouping cells to check whether they qualify as a jet.

PARU(55) : (D=0.5) when smearing the transverse energy (or energy, see MSTU(53)) in PYCELL, the calorimeter cell resolution is taken to be PARU(55)$\times\sqrt{E_\perp}$ (or PARU(55)$\times\sqrt{E}$) for $E_\perp$ (or E) in GeV.

PARU(56) : (D=2.) maximum factor of upward fluctuation in transverse energy or energy in a given cell when calorimeter resolution is included in PYCELL (see MSTU(53)).

PARU(57) : (D=3.2) maximum rapidity (or pseudorapidity or pion rapidity, depending on MSTU(42)) used in the factorial moments analysis in PYTABU.

PARU(58) : (D=0. GeV) in a PYCELL call, cells with a transverse energy $E_\perp$ below PARP(58) are removed from further consideration. This may be used to represent a threshold in an actual calorimeter, or may be chosen just to speed up the algorithm in a high-multiplicity environment.

PARU(61) : (I) invariant mass W of a system analyzed with PYCLUS or PYJMAS, with energies calculated according to the MSTU(42) value.

PARU(62) : (R) the generalized thrust obtained after a successful PYCLUS call, i.e. ratio of summed cluster momenta and summed particle momenta.

PARU(63) : (R) the minimum distance d between two clusters in the final cluster configuration after a successful PYCLUS call; is 0 if only one cluster left.

15.6 Histograms

The GBOOK package was written in 1979, at a time when HBOOK [Bru87] was not available in Fortran 77. It has been used since as a small and simple histogramming program. For this version of PYTHIA the program has been updated to run together with PYTHIA in double precision. Only the one-dimensional histogram part has been retained, and subroutine names have been changed to fit PYTHIA conventions. These modified routines are now distributed together with PYTHIA. They would not be used for final graphics, but may be handy for simple checks, and are extensively used to provide free-standing examples of analysis programs, to be

found on the PYTHIA web page.

There is a maximum of 1000 histograms at your disposal, numbered in the range 1 to 1000. Before a histogram can be filled, space must be reserved (booked) for it, and histogram information provided. Histogram contents are stored in a commonblock of dimension 20000, in the order they are booked. Each booked histogram requires NX+28 numbers, where NX is the number of x bins and the 28 include limits, under/overflow and the title. If you run out of space, the program can be recompiled with larger dimensions. The histograms can be manipulated with a few routines. Histogram output is 'line printer' style, i.e. no graphics.

```
CALL PYBOOK(ID,TITLE,NX,XL,XU)
```

Purpose: to book a one-dimensional histogram.
`ID :` histogram number, integer between 1 and 1000.
`TITLE :` histogram title, at most 60 characters.
`NX :` number of bins in the histogram; integer between 1 and 100.
`XL, XU :` lower and upper bound, respectively, on the *x* range covered by the histogram.

```
CALL PYFILL(ID,X,W)
```

Purpose: to fill a one-dimensional histogram.
`ID :` histogram number.
`X :` *x* coordinate of point.
`W :` weight to be added in this point.

```
CALL PYFACT(ID,F)
```

Purpose: to rescale the contents of a histogram.
`ID :` histogram number.
`F :` rescaling factor, i.e. a factor that all bin contents (including overflow etc.) are multiplied by.
Remark: a typical rescaling factor could be f = 1/(bin size * number of events) = `NX/(XU-XL)` * 1/(number of events).

```
CALL PYOPER(ID1,OPER,ID2,ID3,F1,F2)
```

Purpose: this is a general-purpose routine for editing one or several histograms, which all are assumed to have the same number of bins. Operations are carried out bin by bin, including overflow bins etc.
`OPER:` gives the type of operation to be carried out, a one-character string or a `CHARACTER*1` variable.

- `= '+', '-', '*', '/' :` add, subtract, multiply or divide the contents in `ID1` and `ID2` and put the result in `ID3`. `F1` and `F2`, if not 1D0, give factors by which the `ID1` and `ID2` bin contents are multiplied before the indicated operation. (Division with vanishing bin content will give 0.)
- `= 'A', 'S', 'L' :` for `'S'` the square root of the content in `ID1` is taken (result 0 for negative bin contents) and for `'L'` the 10-logarithm is taken (a nonpositive bin content is before that replaced by 0.8 times the smallest positive bin content). Thereafter, in all three cases, the content is multiplied by `F1` and added with `F2`, and the result is placed in `ID3`. Thus `ID2` is dummy in these cases.
- `= 'M' :` intended for statistical analysis, bin-by-bin mean and standard deviation of a variable, assuming that `ID1` contains accumulated weights, `ID2` accumulated

weight*variable and ID3 accumulated weight*variable-squared. Afterwards ID2 will contain the mean values (=ID2/ID1) and ID3 the standard deviations ($= \sqrt{\mathtt{ID3/ID1} - (\mathtt{ID2/ID1})^2}$). In the end, F1 multiplies ID1 (for normalization purposes), while F2 is dummy.

ID1, ID2, ID3 : histogram numbers, used as described above.

F1, F2 : factors or offsets, used as described above.

```
CALL PYHIST
```

Purpose: to print all histograms that have been filled, and thereafter reset their bin contents to 0.

```
CALL PYPLOT(ID)
```

Purpose: to print out a single histogram.

ID : histogram to be printed.

```
CALL PYNULL(ID)
```

Purpose: to reset all bin contents, including overflow etc., to 0.

ID : histogram to be reset.

```
CALL PYDUMP(MDUMP,LFN,NHI,IHI)
```

Purpose: to dump the contents of existing histograms on an external file, from which they could be read in to another program.

MDUMP : the action to be taken.

- = 1 : dump histograms, each with the first line giving histogram number and title, the second the number of *x* bins and lower and upper limit, the third the total number of entries and under-, inside- and overflow, and subsequent ones the bin contents grouped five per line. If NHI=0 all existing histograms are dumped and IHI is dummy, else the NHI histograms with numbers IHI(1) through IHI(NHI) are dumped.
- = 2 : read in histograms dumped with MDUMP=1 and book and fill histograms according to this information. (With modest modifications this option could instead be used to write the info to HBOOK/HPLOT format, or whatever.) NHI and IHI are dummy.
- = 3 : dump histogram contents in column style, where the first column contains the *x* values (average of respective bin) of the first histogram, and subsequent columns the histogram contents. All histograms dumped this way must have the same number of *x* bins, but it is not checked whether the *x* range is also the same. If NHI=0 all existing histograms are dumped and IHI is dummy, else the NHI histograms with numbers IHI(1) through IHI(NHI) are dumped. A file written this way can be read e.g. by GNUPLOT [Gnu99].

LFN : the file number to which the contents should be written. You must see to it that this file is properly opened for write (since the definition of file names is platform dependent).

NHI : number of histograms to be dumped; if 0 then all existing histograms are dumped.

IHI : array containing histogram numbers in the first NHI positions for NHI nonzero.

```
COMMON/PYBINS/IHIST(4),INDX(1000),BIN(20000)
```

Purpose: to contain all information on histograms.

`IHIST(1)` : (D=1000) maximum allowed histogram number, i.e. dimension of the `INDX` array.

`IHIST(2)` : (D=20000) size of histogram storage, i.e. dimension of the `BIN` array.

`IHIST(3)` : (D=55) maximum number of lines per page assumed for printing histograms. 18 lines are reserved for title, bin contents and statistics, while the rest can be used for the histogram proper.

`IHIST(4)` : internal counter for space usage in the `BIN` array.

`INDX` : gives the initial address in `BIN` for each histogram. If this array is expanded, also `IHIST(1)` should be changed.

`BIN` : gives bin contents and some further histogram information for the booked histograms. If this array is expanded, also `IHIST(2)` should be changed.

16 Summary and Outlook

A complete description of the PYTHIA program would have to cover four aspects:

1. the basic philosophy and principles underlying the programs;
2. the detailed physics scenarios implemented, with all the necessary compromises and approximations;
3. the structure of the implementation, including program flow, internal variable names and programming tricks; and
4. the manual, which describes how to use the programs.

Of these aspects, the first has been dealt with in reasonable detail. The second is unevenly covered: in depth for aspects which are not discussed anywhere else, more summarily for areas where separate up-to-date papers already exist. The third is not included at all, but 'left as an exercise' for the reader, to figure out from the code itself. The fourth, finally, should be largely covered, although many further comments could have been made, in particular about the interplay between different parts of the programs. Still, in the end, no manual, however complete, can substitute for 'hands on' experience.

The PYTHIA program is continuously being developed. We are aware of many shortcomings, some of which hopefully will be addressed in the future. Mainly this is a matter of including new interesting physics scenarios and improving the existing ones, but also some cleanup and reorganization would be appropriate. No timetable is set up for such future changes, however. After all, this is not a professionally maintained software product, but part of a small physics research project. Very often, developments of the programs have come about as a direct response to the evolution of the physics stage, i.e. experimental results and studies for future accelerators. Hopefully, the program will keep on evolving in step with the new challenges opening up.

In the longer future, a radically new version of the program is required. Given the decisions by the big laboratories and collaborations to discontinue the use of Fortran and instead adopt C++, it is natural to attempt to move also event generators in that direction. User-friendly interfaces will have to hide the considerable underlying complexity from the non-expert. The PYTHIA 7 project got going in the beginning of 1998, and is an effort to reformulate the event generation process in object oriented language. Even if much of the physics will be carried over unchanged, none of the existing code will survive. The structure of the event record and the whole administrative apparatus is completely different from the current one, in order to allow a much more general and flexible formulation of the event generation process. A strategy document [Lön99] was followed by a first 'proof of concept version' in June 2000 [Ber01], containing the generic event generation machinery, some processes, and the string fragmentation routines. In the next few years, the hope is to produce useful versions, even if still limited in scope. Due to the considerable complexity of the undertaking, it will still be several years before the C++ version of PYTHIA will contain more and better physics than the Fortran one. The two versions therefore will coexist for several years, with the Fortran one used for physics 'production' and the C++ one for exploration of the object-oriented approach that will be standard at the LHC.

References

[Abb87] A. Abbasabadi and W. Repko, Phys. Lett. **B199** (1987) 286; Phys. Rev. **D37** (1988) 2668;
W. Repko and G.L. Kane, private communication

[Ada93] WA82 Collaboration, M. Adamovich et al., Phys. Lett. **B305** (1993) 402;
E769 Collaboration, G.A. Alves et al., Phys. Rev. Lett. **72** (1994) 812;
E791 Collaboration, E.M. Aitala et al., Phys. Lett. **B371** (1996) 157

[AFS87] AFS Collaboration, T. Åkesson et al., Z. Phys. **C34** (1987) 163;
UA2 Collaboration, J. Alitti et al., Phys. Lett. **B268** (1991) 145;
L. Keeble (CDF Collaboration), FERMILAB-CONF-92-161-E (1992)

[ALE92] ALEPH Collaboration, D. Buskulic et al., Phys. Lett. **B292** (1992) 210

[Ali80] A. Ali, J.G. Körner, G. Kramer and J. Willrodt, Nucl. Phys. **B168** (1980) 409;
A. Ali, E. Pietarinen, G. Kramer and J. Willrodt, Phys. Lett. **B93** (1980) 155

[Ali80a] A. Ali, J.G. Körner, Z. Kunszt, E. Pietarinen, G. Kramer, G. Schierholz and J. Willrodt, Nucl. Phys. **B167** (1980) 454

[Ali82] A. Ali, Phys. Lett. **B110** (1982) 67;
A. Ali and F. Barreiro, Phys. Lett. **B118** (1982) 155; Nucl. Phys. **B236** (1984) 269

[Ali88] A. Ali et al., in 'Proceedings of the HERA Workshop', ed. R.D. Peccei (DESY, Hamburg, 1988), Vol. 1, p. 395;
M. Bilenky and G. d'Agostini, private communication (1991)

[Alt77] G. Altarelli and G. Parisi, Nucl. Phys. **B126** (1977) 298

[Alt78] G. Altarelli and G. Martinelli, Phys. Lett. **76B** (1978) 89 ;
A. Mendéz, Nucl. Phys. **B145** (1978) 199;
R. Peccei and R. Rückl, Nucl. Phys. **B162** (1980) 125;
Ch. Rumpf, G. Kramer and J. Willrodt, Z. Phys. **C7** (1981) 337

[Alt89] G. Altarelli, B. Mele and M. Ruiz-Altaba, Z. Phys. **C45** (1989) 109

[Ama80] D. Amati, A. Bassetto, M. Ciafaloni, G. Marchesini and G. Veneziano, Nucl. Phys. **B173** (1980) 429;
G. Curci, W. Furmanski and R. Petronzio, Nucl. Phys. **B175** (1980) 27

[And79] B. Andersson, G. Gustafson and C. Peterson, Z. Phys. **C1** (1979) 105;
B. Andersson and G. Gustafson, Z. Phys. **C3** (1980) 22;
B. Andersson, G. Gustafson and T. Sjöstrand, Z. Phys. **C6** (1980) 235; Z. Phys. **C12** (1982) 49

[And80] B. Andersson, G. Gustafson and T. Sjöstrand, Phys. Lett. **B94** (1980) 211

[And81] B. Andersson, G. Gustafson, I. Holgersson and O. Månsson, Nucl. Phys. **B178** (1981) 242

[And81a] B. Andersson, G. Gustafson, G. Ingelman and T. Sjöstrand, Z. Phys. **C9** (1981) 233

[And82] B. Andersson, G. Gustafson and T. Sjöstrand, Nucl. Phys. **B197** (1982) 45

[And82a] B. Andersson and G. Gustafson, LU TP 82-5 (1982)

[And83] B. Andersson, G. Gustafson, G. Ingelman and T. Sjöstrand, Phys. Rep. **97** (1983) 31

[And83a] B. Andersson, G. Gustafson and B. Söderberg, Z. Phys. **C20** (1983) 317

[And85] B. Andersson, G. Gustafson and T. Sjöstrand, Physica Scripta **32** (1985) 574

[And89] B. Andersson, P. Dahlqvist and G. Gustafson, Z. Phys. **C44** (1989) 455;
B. Andersson, G. Gustafson, A. Nilsson and C. Sjögren, Z. Phys. **C49** (1991) 79

[And98] B. Andersson, 'The Lund Model' (Cambridge University Press, 1998)

[And98a] J. André and T. Sjöstrand, Phys. Rev. **D57** (1998) 5767

[Ans90] F. Anselmo et al., in 'Large Hadron Collider Workshop', eds. G. Jarlskog and D. Rein, CERN 90-10 (Geneva,1990), Vol. II, p. 130

[App92] T. Appelquist and G. Triantaphyllou, Phys. Pev. Lett. **69** (1992) 2750

[Art74] X. Artru and G. Mennessier, Nucl. Phys. **B70** (1974) 93

[Art83] X. Artru, Phys. Rep. **97** (1983) 147

[Bab80] J.B. Babcock and R.E. Cutkosky, Nucl. Phys. **B176** (1980) 113;
J. Dorfan, Z. Phys. **C7** (1981) 349;
H.J. Daum, H. Meyer and J. Bürger, Z. Phys. **C8** (1981) 167;
K. Lanius, H.E. Roloff and H. Schiller, Z. Phys. **C8** (1981) 251;
M.C. Goddard, Rutherford preprint RL-81-069 (1981);
A. Bäcker, Z. Phys. **C12** (1982) 161

[Bae93] H. Baer, F.E. Paige, S.D. Protopopescu and X. Tata, in 'Workshop on Physics at Current Accelerators and Supercolliders', eds. J.L. Hewett, A.R. White and D. Zeppenfeld, ANL-HEP-CP-93-92 (Argonne, 1993), p. 703;
H. Baer, F.E. Paige, S.D. Protopopescu and X. Tata, hep-ph/0001086

[Bag82] J.A. Bagger and J.F. Gunion, Phys. Rev. **D25** (1982) 2287

[Bai81] V.N. Baier, E.A. Kuraev, V.S. Fadin and V.A. Khoze, Phys. Rep. **78** (1981) 293

[Bai83] R. Baier and R. Rückl, Z. Phys. **C19** (1983) 251

[Lip76] L.N. Lipatov, Sov. J. Nucl. Phys. **23** (1976) 338;
E.A. Kuraev, L.N. Lipatov and V.S. Fadin, Sov. Phys. JETP **45** (1977) 199;
I. Balitsky and L.N. Lipatov, Sov. J. Nucl. Phys. **28** (1978) 822
V.S. Fadin and L.N. Lipatov, Nucl.Phys. **B477** (1996) 767

[Bál01] C. Bálazs, J. Huston and I. Puljak, Phys. Rev. **D63** (2001) 014021

[Bam00] P. Bambade et al., in 'Reports of the Working Groups on Precision Calculations for LEP2 Physics', eds. S. Jadach, G. Passarino and R. Pittau, CERN 2000-009, p. 137

[Bar86a] A. Bartl, H. Fraas, W. Majerotto, Nucl. Phys. **B278** (1986) 1

[Bar86b] A. Bartl, H. Fraas, W. Majerotto, Z. Phys. **C30** (1986) 441

[Bar87] A. Bartl, H. Fraas, W. Majerotto, Z. Phys. **C34** (1987) 411

[Bar88] R.M. Barnett, H.E. Haber and D.E. Soper, Nucl. Phys. **B306** (1988) 697

[Bar90] T.L. Barklow, SLAC-PUB-5364 (1990)

[Bar90a] V.Barger, K. Cheung, T. Han and R.J.N. Phillips, Phys. Rev. **D42** (1990) 3052

[Bar94] D. Bardin, M. Bilenky, D. Lehner, A. Olchevski and T. Riemann, Nucl. Phys **B**, Proc. Suppl. **37B** (1994) 148;
D. Bardin, private communication

[Bar94a] E. Barberio and Z. Was, Computer Physics Commun. **79** (1994) 291

[Bar95] A. Bartl, W. Majerotto, and W. Porod, Z. Phys.**C68** (1995) 518

[Bas78] C. Basham, L. Brown, S. Ellis and S. Love, Phys. Rev. Lett. **41** (1978) 1585

[Bas83] A. Bassetto, M. Ciafaloni and G, Marchesini, Phys. Rep. **100** (1983) 202

[Bau90] U. Baur, M. Spira and P. M. Zerwas, Phys. Rev. **D42** (1990) 815

[Bee96] W. Beenakker et al., in 'Physics at LEP2', eds. G. Altarelli, T. Sjöstrand and F. Zwirner, CERN 96-01 (Geneva, 1996), Vol. 1, p. 79

[Bel00] A.S. Belyaev et al., talk at the ACAT2000 Workshop, Fermilab, October 16–20, 2000 [hep-ph/0101232]

[Ben84] H.-U. Bengtsson, Computer Physics Commun. **31** (1984) 323

[Ben84a] H.-U. Bengtsson and G. Ingelman, LU TP 84-3, Ref.TH.3820-CERN (1984)

[Ben85] H.-U. Bengtsson and G. Ingelman, Computer Physics Commun. **34** (1985) 251

[Ben85a] H.-U. Bengtsson, W.-S. Hou, A. Soni and D.H. Stork, Phys. Rev. Lett. **55** (1985) 2762

[Ben87] H.-U. Bengtsson and T. Sjöstrand, Computer Physics Commun. **46** (1987) 43

[Ben87a] M. Bengtsson and T. Sjöstrand, Phys. Lett. **B185** (1987) 435; Nucl. Phys. **B289** (1987) 810

[Ben87b] M. C. Bento and C. H. Llewellyn Smith, Nucl. Phys. **B289** (1987) 36

[Ben88] M. Bengtsson and T. Sjöstrand, Z. Phys. **C37** (1988) 465

[Ber81] E.L. Berger and D. Jones, Phys. Rev. **D23** (1981) 1521

[Ber82] F.A. Berends, R. Kleiss and S. Jadach, Nucl. Phys. **B202** (1982) 63; Computer Physics Commun. **29** (1983) 185

[Ber84] E. L. Berger, E. Braaten and R. D. Field, Nucl. Phys. **B239** (1984) 52

[Ber85] F.A. Berends and R. Kleiss, Nucl. Phys. **B260** (1985) 32

[Ber85a] L. Bergström and G. Hulth, Nucl. Phys. **B259** (1985) 137

[Ber89] F.A. Berends et al., in 'Z Physics at LEP 1', eds. G. Altarelli, R. Kleiss and C. Verzegnassi, CERN 89-08 (Geneva, 1989), Vol. 1, p. 89

[Ber01] M. Bertini, L. Lönnblad and T. Sjöstrand, Computer Physics Commun. **134** (2001) 365

[Bet89] S. Bethke, Z. Phys. **C43** (1989) 331

[Bet92] S. Bethke, Z. Kunszt, D.E. Soper and W.J. Stirling, Nucl. Phys. **B370** (1992) 310

[Bia86] A. Białas and R. Peschanski, Nucl. Phys. **B273** (1986) 703

[Bij01] J. Bijnens, P. Eerola, M. Maul, A. Månsson and T. Sjöstrand, Phys. Lett. **B503** (2001) 341

[Bjo70] J.D. Bjorken and S.J. Brodsky, Phys. Rev. **D1** (1970) 1416

[Bon73] G. Bonneau, M. Gourdin and F. Martin, Nucl. Phys.**B54** (1973) 573

[Boo01] E. Boos et al., in preparation, to appear in the proceedings of the Workshop on Physics at TeV Colliders, Les Houches, France, 21 May – 1 June 2001

[Bor93] F.M. Borzumati and G.A. Schuler, Z. Phys. **C58** (1993) 139

[Bow81] M.G. Bowler, Z. Phys. **C11** (1981) 169

[Bra64] S. Brandt, Ch. Peyrou, R. Sosnowski and A. Wroblewski, Phys. Lett. **12** (1964) 57;
E. Fahri, Phys. Rev. Lett. **39** (1977) 1587

[Bra79] S. Brandt and H.D. Dahmen, Z. Phys. **C1** (1979) 61

[Bru87] R. Brun and D. Lienart, 'HBOOK User Guide', CERN program library long write-up Y250 (1987);
R. Brun and N. Cremel Somon, 'HPLOT User Guide', CERN program library long write-up Y251 (1988)

[Bru89] R. Brun et al., GEANT 3, CERN report DD/EE/84-1 (1989)

[Bru96] P. Bruni, A. Edin and G. Ingelman, in preparation (draft ISSN 0418-9833)

[Bud75] V.M. Budnev, I.F. Ginzburg, G.V. Meledin and V.G. Serbo,
Phys. Rep **15** (1975) 181

[Cah84] R.N. Cahn and S. Dawson, Phys. Lett. **136B** (1984) 196;
R.N. Cahn, Nucl. Phys. **B255** (1985) 341;
G. Altarelli, B. Mele and F. Pitolli, Nucl. Phys. **B287** (1987) 205

[Can97] B. Cano-Coloma and M.A. Sanchis-Lozano, Nucl. Phys. **B508** (1997) 753;
A. Edin, G. Ingelman and J. Rathsman, Phys. Rev. **D56** (1997) 7317

[Car95] M. Carena, J.–R. Espinosa, M. Quiros and C.E.M. Wagner, Phys. Lett. **B355** (1995) 209;
M. Carena, M. Quiros and C.E.M. Wagner, Nucl. Phys. **B461** (1996) 407

[Car96] M. Carena et al., in 'Physics at LEP2', eds. G. Altarelli, T. Sjöstrand and F. Zwirner, CERN 96-01 (Geneva, 1996), Vol. 1, p. 351

[Car00] M. Carena et al., 'Report of the Tevatron Higgs Working Group',
FERMILAB-CONF-00-279-T [hep-ph/0010338];
M. Spira, talk at the Workshop on Physics at TeV Colliders, Les Houches, 21 May – 1 June 2001

[Cat91] S. Catani, Yu. L. Dokshitzer, M. Olsson, G. Turnock and B.R. Webber, Phys. Lett. **B269** (1991) 432

[CDF97] CDF Collaboration, F. Abe et al., Phys. Rev. Lett. **79** (1997) 584.

[Chi90] P. Chiappetta and M. Perrottet, in 'Large Hadron Collider Workshop', eds. G. Jarlskog and D. Rein, CERN 90-10 (Geneva, 1990), Vol. II, p. 806

[Cha85] M. Chanowitz and M.K. Gaillard, Nucl. Phys. **B261** (1985) 379

[Che75] M.-S. Chen and P. Zerwas, Phys. Rev. **D12** (1975) 187;
P. Zerwas, private communication (1991)

[Chu55] A.E. Chudakov, Izv. Akad. Nauk SSSR, Ser. Fiz. **19** (1955) 650

[Chý00] J. Chýla, Phys. Lett. **B488** (2000) 289

[Cia87] M. Ciafaloni, Nucl. Phys. **B296** (1987) 249;
S. Catani, F. Fiorani and G. Marchesini, Nucl. Phys. **B336** (1990) 18;
G. Marchesini and B.R. Webber, Nucl. Phys. **B349** (1991) 617

[Cla79] L. Clavelli, Phys. Lett. **B85** (1979) 111;
A.V. Smilga, Nucl. Phys. **B161** (1979) 449;
L. Clavelli and D. Wyler, Phys. Lett. **103B** (1981) 383

[Coc91] D. Cocolicchio, F. Feruglio, G.L. Fogli and J. Terron, Phys. Lett. **B255** (1991) 599;
F. Feruglio, private communication (1990)

[Col00] J. Collins, JHEP **05** (2000) 004;
Y. Chen, J. Collins and N. Tkachuk, hep-ph/0105291

[Com77] B.L. Combridge, J. Kripfganz and J. Ranft, Phys. Lett. **70B** (1977) 234;
R. Cutler and D. Sivers, Phys. Rev. **D17** (1978) 196

[Com79] B.L. Combridge, Nucl. Phys. **B151** (1979) 429

[Con71] V. Constantini, B. de Tollis and G. Pistoni, Nuovo Cim. **2A** (1971) 733

[Dan82] D. Danckaert, P. De Causmaecker, R. Gastmans, W. Troost and T.T. Wu, Phys. Lett. **B114** (1982) 203

[Daw85] S. Dawson, E. Eichten and C. Quigg, Phys. Rev. **D31** (1985) 1581

[DeR75] A. De Rújula, H. Georgi and S.L Glashow, Phys. Rev. **D12** (1975) 147

[Dic86] D.A. Dicus and S.S.D. Willenbrock, Phys. Rev. **D34** (1986) 155

[Dic88] D.A. Dicus and S.S.D. Willenbrock, Phys. Rev. **D37** (1988) 1801

[Din79] M. Dine and J. Sapirstein, Phys. Rev. Lett. **43** (1979) 668;
K.G. Chetyrkin et al., Phys. Lett. **B85** (1979) 277;
W. Celmaster and R.J. Gonsalves, Phys. Rev. Lett. **44** (1980) 560

[Din96] M. Dine, A.E. Nelson, Y. Nir, and Y. Shirman, Phys. Rev. **D53** (1996) 2658

[Dis01] J. Dischler and T. Sjöstrand, EPJdirect **C2** (2001) 1

[Dob91] A. Dobado, M.J. Herrero and J. Terron, Z. Phys. **C50** (1991) 205, *ibid.* 465

[Dok89] Yu.L. Dokshitzer, V.A. Khoze and S.I. Troyan, in 'Perturbative QCD', ed. A.H. Mueller (World Scientific, Singapore, 1989), p. 241

[Dok92] Yu.L. Dokshitzer, V.A. Khoze and T. Sjöstrand, Phys. Lett. **B274** (1992) 116

[Don92] A. Donnachie and P.V. Landshoff, Phys. Lett. **B296** (1992) 227

[Dre85] M. Drees and K. Grassie, Z. Phys. **C28** (1985) 451

[Dre89] M. Drees, J. Ellis and D. Zeppenfeld, Phys. Lett. **B223** (1989) 454

[Dre91] M. Drees and C.S. Kim, Z. Phys. **C53** (1991) 673.

[Dre95] M. Drees and S.P. Martin, in 'Electroweak symmetry breaking and new physics at the TeV scale', eds. T.L. Barklow et al., p. 146 [hep-ph/9504324]

[Dre00] H. Dreiner, P. Richardson and M. H. Seymour, JHEP **0004** (2000) 008 [hep-ph/9912407]

[Duk82] D.W. Duke and J.F. Owens, Phys. Rev. **D26** (1982) 1600

[Dun86] M.J. Duncan, G.L. Kane and W.W. Repko, Nucl. Phys. **B272** (1986) 517

[Edé97] P. Edén and G. Gustafson, Z. Phys. **C75** (1997) 41;
P. Edén, LUTP 96–29 [hep-ph/9610246]

[Edé00] P. Edén, JHEP **05** (2000) 029

[Eic84] E. Eichten, I. Hinchliffe, K. Lane and C. Quigg, Rev. Mod. Phys. **56** (1984) 579; Rev. Mod. Phys. **58** (1985) 1065

[Eic96] E. Eichten and K. Lane, Phys. Lett. **B388** (1996) 803;
E. Eichten, K. Lane and J. Womersley, Phys. Lett. **B405** (1997) 305

[Eij90] B. van Eijk and R. Kleiss, in 'Large Hadron Collider Workshop', eds. G. Jarlskog and D. Rein, CERN 90-10 (Geneva, 1990), Vol. II, p. 183

[Ell76] J. Ellis, M.K. Gaillard and G.G. Ross, Nucl. Phys. **B111** (1976) 253

[Ell79] J. Ellis and I. Karliner, Nucl. Phys. **B148** (1979) 141

[Ell81] R.K. Ellis, D.A. Ross and A.E. Terrano, Nucl. Phys. **B178** (1981) 421

[Ell86] R.K. Ellis and J.C. Sexton, Nucl. Phys. **B269** (1986) 445

[Ell88] R.K. Ellis, I. Hinchliffe, M. Soldate and J.J. van der Bij, Nucl. Phys. **B297** (1988) 221

[EMC87] EMC Collaboration, M. Arneodo et al., Z. Physik **C36** (1987) 527;
L. Apanasevich et al., Phys. Rev. **D59** (1999) 074007

[Fab82] K. Fabricius, G. Kramer, G. Schierholz and I. Schmitt, Z. Phys. **C11** (1982) 315

[Fad90] V. Fadin, V. Khoze and T. Sjöstrand, Z. Phys. **C48** (1990) 613

[Fer00] A. Ferrari et al., Phys. Rev. **D62** (2000) 013001;
A. Ferrari, private communication

[Fie78] R.D. Field and R.P. Feynman, Nucl. Phys. **B136** (1978) 1

[Fon81] M. Fontannaz, B. Pire and D. Schiff, Z. Phys. **C11** (1981) 211

[Fox79] G.C. Fox and S. Wolfram, Nucl. Phys. **B149** (1979) 413

[Fri93] S. Frixione, M.L. Mangano, P. Nason and G. Ridolfi, Phys. Lett. **B319** (1993) 339

[Fri97] C. Friberg, E. Norrbin and T. Sjöstrand, Phys. Lett. **B403** (1997) 329

[Fri00] C. Friberg and T. Sjöstrand, Eur. Phys. J. **C13** (2000) 151, JHEP **09** (2000) 010, Phys. Lett. **B492** (2000) 123

[Gab86] E. Gabrielli, Mod. Phys. Lett. **A1** (1986) 465

[Gae80] K.J.F. Gaemers and J.A.M. Vermaseren, Z. Phys. **C7** (1980) 81

[Gar98] L. Garren, `http://www-pat.fnal.gov/mcgen/lund/convert.pl`

[Gas87] R. Gastmans, W. Troost and T.T. Wu, Phys. Lett. **B184** (1987) 257

[Gin82] I.F. Ginzburg, G.L. Kotkin, V.G. Serbo and V.I. Telnov, JETP Lett. **34** (1982) 491, Nucl. Instrum. Meth. **205** (1983) 47

[Glo88] E.W.N. Glover, A.D. Martin and W.J. Stirling, Z. Phys. **C38** (1988) 473

[Glü92] M. Glück, E. Reya and A. Vogt, Z. Phys. **C53** (1992) 127

[Glü92a] M. Glück, E. Reya and A. Vogt, Z. Phys. **C53** (1992) 651

[Glü95] M. Glück, E. Reya and A. Vogt, Z. Phys. **C67** (1995) 433

[Glü99] M. Glück, E. Reya and I. Schienbein, Phys. Rev. **D60** (1999) 054019, Erratum Phys. Rev. **D62** (2000) 019902

[Gnu99] GNUPLOT website at `www.gnuplot.org`

[Got82] T.D. Gottschalk, Phys. Lett. **B109** (1982) 331;
T.D. Gottschalk and M.P. Shatz, Phys. Lett. **B150** (1985) 451, CALT-68-1172 (1984)

[Got86] T.D. Gottschalk, Nucl. Phys. **B277** (1986) 700

[Gri72] V.N. Gribov and L.N. Lipatov, Sov. J. Nucl. Phys. **15** (1972) 438, *ibid.* 75;
Yu. L. Dokshitzer, Sov. J. Phys. JETP **46** (1977) 641

[Gri83] L.V. Gribov, E.M. Levin and M.G. Ryskin, Phys. Rep. **100** (1983) 1

[Gro81] T.R. Grose and K.O. Mikaelian, Phys. Rev. **D23** (1981) 123

[Gül93] St. Güllenstern, P. Górnicki, L. Mankiewicz and A. Schäfer, Nucl. Phys. **A560** (1993) 494

[Gun86] J.F. Gunion and Z. Kunszt, Phys. Rev. **D33** (1986) 665;
errata as private communication from the authors

[Gun87] J.F. Gunion, H.E. Haber, F.E. Paige, W.-K. Tung and S.S.D. Willenbrock, Nucl. Phys. **B294** (1987) 621

[Gun88] J. Gunion and H. Haber, Phys. Rev. **D37** (1988) 2515

[Gun90] J.F. Gunion, H.E. Haber, G. Kane and S. Dawson, The Higgs Hunter's Guide (Addison-Wesley, 1990);
A. Djouadi, private communication (1991)

[Gus82] G. Gustafson, Z. Phys. **C15** (1982) 155

[Gus88] G. Gustafson, U. Pettersson and P. Zerwas, Phys. Lett. **B209** (1988) 90

[Gus94] G. Gustafson and J. Häkkinen, Z. Phys. **C64** (1994) 659.

[Gut84] F. Gutbrod, G. Kramer and G. Schierholz, Z. Phys. **C21** (1984) 235

[Gut87] F. Gutbrod, G. Kramer, G. Rudolph and G. Schierholz, Z. Phys. **C35** (1987) 543

[Hab85] H.E. Haber and G.L. Kane, Phys. Rep. **117** (1985) 75.

[Hag91] K. Hagiwara, H. Iwasaki, A. Miyamoto, H. Murayama and D. Zeppenfeld, Nucl. Phys. **B365** (1991) 544

[Hal78] F. Halzen and D. M. Scott, Phys. Rev. **D18** (1978) 3378

[HerBC] Herodotus of Halicarnassus, 'The Histories' (circa 430 BC),
translation to English e.g. by A. de Sélincourt (1954), available in Penguin Classics

[HER92] 'Physics at HERA', eds. W Buchmüller and G. Ingelman (DESY, Hamburg, 1992), Vol. 3

[HER99] 'Monte Carlo Generators for HERA Physics', eds. A.T. Doyle, G. Grindhammer, G. Ingelman and H. Jung, DESY-Proc-1999-02 (DESY, Hamburg, 1999)

[Hew88] J.L. Hewett and S. Pakvasa, Phys. Rev. **D37** (1988) 3165, and private communication from the authors

[Hin93] I. Hinchliffe and T. Kaeding, Phys. Rev. **D47** (1993) 279

[Hoo79] G. 't Hooft and M. Veltman, Nucl. Phys. **B153** (1979) 365

[Hoy79] P. Hoyer, P. Osland, H.G. Sander, T.F. Walsh and P.M. Zerwas, Nucl. Phys. **B161** (1979) 349

[Hui97] K. Huitu, J. Maalampi, A. Pietilä and M. Raidal, Nucl. Phys. **B487** (1997) 27 and private communication;
G. Barenboim, K. Huitu, J. Maalampi and M. Raidal, Phys. Lett. **B394** (1997) 132

[Ing80] G. Ingelman and T. Sjöstrand, LUTP 80-12 (1980);
G. Ingelman, A. Edin and J. Rathsman, Computer Physics Commun. **101** (1997) 108

[Ing85] G. Ingelman and P.E. Schlein, Phys. Lett. **152B** (1985) 256

[Ing87] G. Ingelman, Computer Physics Commun. **46** (1987) 217

[Ing87a] G. Ingelman et al., in 'Proceedings of the HERA Workshop', ed. R.D. Peccei (DESY, Hamburg, 1988), Vol. 1, p. 3

[Ing88] G. Ingelman and G.A. Schuler, Z. Phys. **C40** (1988) 299;
G. Ingelman and G.A. Schuler, in 'Physics at HERA', eds. W. Buchmüller and G. Ingelman (DESY, Hamburg, 1992), Vol. 3, p. 1346

[Iof78] B.L. Ioffe, Phys. Lett. **78B** (1978) 277

[JAD86] JADE Collaboration, W. Bartel et al., Z. Phys. **C33** (1986) 23;
S. Bethke, Habilitation thesis, LBL 50-208 (1987)

[JAD88] JADE Collaboration, S. Bethke et al., Phys. Lett. **B213** (1988) 235;
TASSO Collaboration, W. Braunschweig et al., Phys. Lett. **214B** (1988) 286

[Jad91] S. Jadach, Z. Was and J.H. Kühn, Computer Physics Commun. **64** (1991) 275;
M. Jezabek, Z. Was, S. Jadach and J.H. Kühn, Computer Physics Commun. **70** (1992) 69;
S. Jadach, Z. Was, R. Decker and J.H. Kühn, Computer Physics Commun. **76** (1993) 361

[Jam80] F. James, Rep. Prog. Phys. **43** (1980) 1145

[Jam90] F. James, Computer Physics Commun. **60** (1990) 329

[Jer81] J. Jersák, E. Laermann and P.M. Zerwas, Phys. Rev. **D25** (1982) 1218

[Jeż89] M. Jeżabek and J.H. Kühn, Nucl. Phys. **B314** (1989) 1

[Jol99] A. Joly, J. Gascon, P. Taras, Eur. Phys. J. **C6** (1999) 413.

[Jun97] H. Jung, private communication;
H. Kharraziha, private communication

[Kat83] M. Katuya, Phys. Lett. **124B** (1983) 421

[Kat98] S. Katsanevas and P. Morawitz, Computer Physics Commun. **112** (1998) 227

[Kho96] V.A. Khoze and T. Sjöstrand, Z. Phys. **C70** (1996) 625.

[Kle89] R. Kleiss et al., in 'Z physics at LEP 1', eds. G. Altarelli, R. Kleiss and C. Verzegnassi, CERN 89-08 (Geneva, 1989), Vol. 3, p. 143

[Kni89] B.A. Kniehl and J.H. Kühn, Phys. Lett. **B224** (1989) 229

[Kno93] I.G. Knowles and S.D. Protopopescu, in 'Workshop on Physics at Current Accelerators and Supercolliders', eds. J.L. Hewett, A.R. White and D. Zeppenfeld, ANL-HEP-CP-93-92 (Argonne, 1993), p. 651

[Kno96] I.G. Knowles et al., in 'Physics at LEP2', eds. G. Altarelli, T. Sjöstrand and F. Zwirner, CERN 96–01 (Geneva, 1996), Vol. 2, p. 103

[Kol78] K. Koller and T.F. Walsh, Nucl. Phys. **B140** (1978) 449

[Kol96] C. Kolda and S.P. Martin, Phys. Rev. **D53** (1996) 3871

[Kör85] J.G. Körner and G. Schuler, Z. Phys. **C26** (1985) 559

[Kra88] G. Kramer and B. Lampe, Z. Phys. **C39** (1988) 101; Fortschr. Phys. **37** (1989) 161

[Krz72] A. Krzywicki and B. Petersson, Phys. Rev. **D6** (1972) 924;
J. Finkelstein and R.D. Peccei, Phys. Rev. **D6** (1972) 2606;
F. Niedermayer, Nucl. Phys. **B79** (1974) 355;
A. Casher, J. Kogut and L. Susskind, Phys. Rev. **D10** (1974) 732

[Küh89] J.H. Kühn et al., in 'Z Physics at LEP 1', eds. G. Altarelli, R. Kleiss and C. Verzegnassi, CERN 89-08 (Geneva, 1989), Vol. 1, p. 267

[Kun81] Z. Kunszt, Phys. Lett. **B99** (1981) 429; Phys. Lett. **B107** (1981) 123

[Kun84] Z. Kunszt, Nucl. Phys. **B247** (1984) 339

[Kun87] Z. Kunszt et al., in 'Proceedings of the Workshop on Physics at Future Accelerators', ed. J.H. Mulvey, CERN 87-08 (1987), Vol. I, p. 123, and private communication

[Lae80] E. Laermann, K.H. Streng and P.M. Zerwas, Z. Phys. **C3** (1980) 289; Erratum Z. Phys. **C52** (1991) 352

[Lai95] CTEQ Collaboration, H.L. Lai et al., Phys. Rev. **D51** (1995) 4763

[Lai00] CTEQ Collaboration, H.L. Lai et al., Eur. Phys. J. **C12** (2000) 375

[Lan91] K. Lane, private communication (1991)

[Lan99] K. Lane, Phys. Rev. **D60** (1999) 075007;
S. Mrenna, Phys. Lett. **B461** (1999) 352

[LEP90] OPAL Collaboration, M.Z. Akrawy et al., Z. Phys **C47** (1990) 505;
L3 Collaboration, B. Adeva et al., Z. Phys. **C55** (1992) 39;
ALEPH Collaboration, D. Buskulic et al., Z. Phys. **C55** (1992) 209

[Lev90] E.M. Levin and M.G. Ryskin, Phys. Rep. **189** (1990) 267

[LHC00] 'Proceedings of the Workshop on Standard Model Physics (and more) at the LHC', eds. G. Altarelli and M.L. Mangano, CERN 2000–004 (Geneva, 2000)

[Lin97] O. Linossier and R. Zitoun, internal ATLAS note and private communication;
V. Barger et al., Phys. Rev. **D49** (1994) 79

[Lön95] L. Lönnblad and T. Sjöstrand, Phys. Lett. **B351** (1995) 293, Eur. Phys. J. **C2** (1998) 165

[Lön96] L. Lönnblad et al., in 'Physics at LEP2', eds. G. Altarelli, T. Sjöstrand and F. Zwirner, CERN 96–01 (Geneva, 1996), Vol. 2, p. 187

[Lön99] L. Lönnblad, Computer Physics Commun. **118** (1999) 213

[Lör89] B. Lörstad, Int. J. of Mod. Phys. **A4** (1989) 2861

[Lus91] M. Lusignoli and M. Masetti, Z. Physik **C51** (1991) 549

[Lüs94] M. Lüscher, Computer Physics Commun. **79** (1994) 100;
F. James, Computer Physics Commun. **79** (1994) 111

[Mag89] N. Magnussen, Ph.D. Thesis, University of Wuppertal WUB-DI 88-4 and DESY F22-89-01 (1989);
G. Kramer and N. Magnussen, Z. Phys. **C49** (1991) 301

[Mah98] G. Mahlon and S. Parke, Phys. Rev. **D58** (1998) 054015

[Man00] M.L. Mangano, in International Europhysics Conference on High Energy Physics, eds. K. Huitu et al. (IOP Publishing, Bristol, 2000), p. 33

[MAR79] MARK J Collaboration, D.P. Barber et al., Phys. Rev. Lett. **43** (1979) 830

[Mar88] G. Marchesini and B.R. Webber, Nucl. Phys. **B310** (1988) 571;
G. Marchesini, B.R. Webber, M.H. Seymour, G. Abbiendi, L. Stanco and I.G. Knowles, Computer Physics Commun. **67** (1992) 465

[Mar90] G. Marsaglia, A. Zaman and W.-W. Tsang, Stat. Prob. Lett. **9** (1990) 35

[Mar94] S.P. Martin and M.T. Vaughn, Phys. Rev. **D50** (1994) 2282

[Miu99] G. Miu and T. Sjöstrand, Phys. Lett. **B449** (1999) 313

[Mon79] I. Montvay, Phys. Lett. **B84** (1979) 331

[Mor89] D.A. Morris, Nucl. Phys. **B313** (1989) 634

[Mor98] S. Moretti, L. Lönnblad and and T. Sjöstrand, JHEP **08** (1998) 001

[Mre97] S. Mrenna, Computer Physics Commun. **101** (1997) 232

[Mre99] S. Mrenna, UCD-99-4 [hep-ph/9902471]

[Mue81] A.H. Mueller, Phys. Lett. **104B** (1981) 161;
B.I. Ermolaev, V.S. Fadin, JETP Lett. **33** (1981) 269

[Nil87] B. Nilsson-Almqvist and E. Stenlund, Computer Physics Commun. **43** (1987) 387;
H. Pi, Computer Physics Commun. **71** (1992) 173

[Nor97] E. Norrbin and T. Sjöstrand, Phys. Rev. **D55** (1997) R5;
V.A. Khoze and T. Sjöstrand, Eur. Phys. J. **C6** (1999) 271, EPJdirect **C1** (2000) 1.

[Nor98] E. Norrbin and T. Sjöstrand, Phys. Lett. **B442** (1998) 407, Eur. Phys. J. **C17** (2000) 137

[Nor01] E. Norrbin and T. Sjöstrand, Nucl. Phys. **B603** (2001) 297

[Ohl97] T. Ohl, Computer Physics Commun.101 (1997) 269

[Ols80] H.A. Olsen, P. Osland and I. Øverbø, Nucl. Phys. **B171** (1980) 209

[OPA91] OPAL Collaboration, M.Z. Akrawy et al., Z. Phys. **C49** (1991) 375

[OPA92] OPAL Collaboration, P.D. Acton et al., Phys. Lett. **B276** (1992) 547

[Owe84] J.F. Owens, Phys. Rev. **D30** (1984) 943

[Par78] G. Parisi, Phys. Lett. **74B** (1978) 65;
J.F. Donoghue, F.E. Low and S.Y. Pi, Phys. Rev. **D20** (1979) 2759

[PDG86] Particle Data Group, M. Aguilar-Benitez et al., Phys. Lett. **B170** (1986) 1

[PDG88] Particle Data Group, G. P. Yost et al., Phys. Lett. **B204** (1988) 1

[PDG92] Particle Data Group, K. Hikasa et al., Phys. Rev. **D45** (1992) S1

[PDG96] Particle Data Group, R.M. Barnett et al., Phys. Rev. **D54** (1996) 1

[PDG00] Particle Data Group, D.E. Groom et al., Eur. Phys. J. **C15** (2000) 1

[Pet83] C. Peterson, D. Schlatter, I. Schmitt and P. Zerwas, Phys. Rev. **D27** (1983) 105

[Pet88] U. Pettersson, LU TP 88-5 (1988);
L. Lönnblad and U. Pettersson, LU TP 88-15 (1988);
L. Lönnblad, Computer Physics Commun. **71** (1992) 15

[Pie97] D.M. Pierce, J.A. Bagger, K. Matchev and R. Zhang, Nucl. Phys. **B491** (1997) 3

[Plo93] H. Plothow-Besch, Computer Physics Commun. **75** (1993) 396, Int. J. Mod. Phys. **A10** (1995) 2901, `http://consult.cern.ch/writeup/pdflib/`

[Puk99] A. Pukhov et al., preprint INP MSU 98-41/542 [hep-ph/9908288]

[Ran99] L. Randall and R. Sundrum, Phys. Rev. Lett. **83** (1999) 3370;
B.C. Allanach, K. Odagiri, M.A. Parker and B.R. Webber, JHEP **0009** (2000) 019

[Riz81] T. Rizzo and G. Senjanovic, Phys. Rev. **D24** (1981) 704

[Sam91] M.A. Samuel, G. Li, N. Sinha, R. Sinha and M.K. Sundaresan, Phys. Rev. Lett. **67** (1991) 9; ERRATUM *ibid.* 2920

[Sch80] G. Schierholz and D.H. Schiller, DESY 80/88 (1980);
J.G. Körner and D.H. Schiller, DESY 81-043 (1981);
K. Koller, D.H. Schiller and D. Wähner, Z. Phys. **C12** (1982) 273

[Sch92] G.A. Schuler and J. Terron, in 'Physics at HERA', eds. W. Buchmüller and G. Ingelman (DESY, Hamburg, 1992), Vol. 1, p. 599

[Sch93] G.A. Schuler and T. Sjöstrand, Phys. Lett. **B300** (1993) 169

[Sch93a] G.A. Schuler and T. Sjöstrand, Nucl. Phys. **B407** (1993) 539

[Sch94] G.A. Schuler and T. Sjöstrand, Phys. Rev. **D49** (1994) 2257

[Sch94a] G.A. Schuler and T. Sjöstrand, in 'Workshop on Two-Photon Physics from DAPHNE to LEP200 and Beyond', eds. F. Kapusta and J. Parisi (World Scientific, Singapore, 1994), p. 163

[Sch95] G.A. Schuler and T. Sjöstrand, Z. Phys. **C68** (1995) 607.

[Sch96] G.A. Schuler and T. Sjöstrand, Phys. Lett. **B376** (1996) 193.

[Sch97] G.A. Schuler and T. Sjöstrand, Z. Phys. **C73** (1997) 677

[Sch98] G.A. Schuler, Computer Physics Commun. **108** (1998) 279

[Sey95] M.H. Seymour, Nucl. Phys. **B436** (1995) 163;
D.J. Miller and M.H. Seymour, Phys. Lett. **B435** (1998) 213

[Sey95a] M.H. Seymour, Phys. Lett. **B354** (1995) 409

[Sjö78] T. Sjöstrand, B. Söderberg, LU TP 78-18 (1978)

[Sjö79] T. Sjöstrand, LU TP 79-8 (1979)

[Sjö80] T. Sjöstrand, LU TP 80-3 (1980)

[Sjö82] T. Sjöstrand, Computer Physics Commun. **27** (1982) 243

[Sjö83] T. Sjöstrand, Computer Physics Commun. **28** (1983) 229

[Sjö84] T. Sjöstrand, Phys. Lett. **142B** (1984) 420, Nucl. Phys. **B248** (1984) 469

[Sjö84a] T. Sjöstrand, Z. Phys. **C26** (1984) 93;
M. Bengtsson, T. Sjöstrand and M. van Zijl, Phys. Lett. **B179** (1986) 164

[Sjö85] T. Sjöstrand, Phys. Lett. **157B** (1985) 321;
M. Bengtsson, T. Sjöstrand and M. van Zijl, Z. Phys. **C32** (1986) 67

[Sjö86] T. Sjöstrand, Computer Physics Commun. **39** (1986) 347

[Sjö87] T. Sjöstrand and M. Bengtsson, Computer Physics Commun. **43** (1987) 367

[Sjö87a] T. Sjöstrand and M. van Zijl, Phys. Rev. **D36** (1987) 2019

[Sjö88] T. Sjöstrand, Int. J. Mod. Phys. **A3** (1988) 751

[Sjö89] T. Sjöstrand et al., in 'Z physics at LEP 1', eds. G. Altarelli, R. Kleiss and C. Verzegnassi, CERN 89-08 (Geneva, 1989), Vol. 3, p. 143

[Sjö92] T. Sjöstrand, in '1991 CERN School of Computing', ed. C. Verkerk, CERN 92-02 (Geneva, 1992), p. 227

[Sjö92a] T. Sjöstrand and P.M. Zerwas, in 'e^+e^- Collisions at 500 GeV: The Physics Potential', ed P.M. Zerwas, DESY 92-123 (Hamburg, 1992), Part A, p. 463;
T. Sjöstrand, in 'Proceedings of the 1992 Workshops on High-Energy Physics with Colliding Beams', ed. J. Rogers, SLAC Report-428 (Stanford, 1993), Vol. 2, p. 445;
V.A. Khoze and T. Sjöstrand, Phys. Lett. **B328** (1994) 466

[Sjö92b] T. Sjöstrand, in 'Physics at HERA', eds. W Buchmüller and G. Ingelman (DESY, Hamburg, 1992), Vol. 3, p. 1405

[Sjö92c] T. Sjöstrand, in 'Workshop on Photon Radiation from Quarks', ed. S. Cartwright, CERN 92-04 (Geneva, 1992), p. 89 and p. 103

[Sjö92d] T. Sjöstrand, CERN-TH.6488/92 (1992)

[Sjö94] T. Sjöstrand, Computer Physics Commun. **82** (1994) 74

[Sjö94a] T. Sjöstrand and V.A. Khoze, Z. Phys. **C62** (1994) 281, Phys. Rev. Lett. **72** (1994) 28

[Sjö01] T. Sjöstrand, P. Edén, C. Friberg, L. Lönnblad, G. Miu, S. Mrenna and E. Norrbin, Computer Physics Commun. **135** (2001) 238

[Ska01] P. Z. Skands, Master's Thesis, Niels Bohr Inst., Copenhagen University (2001)

[Ste81] P.M. Stevenson, Phys. Rev. **D23** (1981) 2916

[Sud56] V.V. Sudakov, Zh.E.T.F. **30** (1956) 87 (Sov. Phys. J.E.T.P. **30** (1956) 65)

[UA183] UA1 Collaboration, G. Arnison et al., Phys. Lett. **123B** (1983) 115;
UA1 Collaboration, C. Albajar et al., Nucl. Phys. **B309** (1988) 405

[UA584] UA5 Collaboration, G.J. Alner et al., Phys. Lett. **138B** (1984) 304;
UA5 Collaboration, R.E. Ansorge et al., Z. Phys. **C43** (1989) 357

[Ver81] J.A.M. Vermaseren, K.J.F. Gaemers and S.J. Oldham, Nucl. Phys. **B187** (1981) 301

[Web86] B.R. Webber, Ann. Rev. Nucl. Part. Sci. **36** (1986) 253

[Wu79] S.L. Wu and G. Zobernig, Z. Phys. **C2** (1979) 107

[Wud86] J. Wudka, Phys. Lett. **167B** (1986) 337

[Zaj87] W.A. Zajc, Phys. Rev. **D35** (1987) 3396

[Zhu83] R.-y. Zhu, Ph. D. Thesis (M.I.T.), MIT-LNS Report RX-1033 (1983); Caltech Report CALT-68-1306; in Proceedings of the 1984 DPF conference, Santa Fe, p. 229; in Proceedings of 1985 DPF conference, Oregon, p. 552

Subprocess Summary Table

This index is intended to give a quick reference to the different physics processes implemented in the program. Further details are to be found elsewhere in the manual, especially in section 8. A trailing '+' on a few SUSY processes indicates inclusion of charge-conjugate modes as well.

No.	Subprocess
Hard QCD processes:	
11	$f_i f_j \to f_i f_j$
12	$f_i \bar{f}_i \to f_k \bar{f}_k$
13	$f_i \bar{f}_i \to gg$
28	$f_i g \to f_i g$
53	$gg \to f_k \bar{f}_k$
68	$gg \to gg$
Soft QCD processes:	
91	elastic scattering
92	single diffraction (XB)
93	single diffraction (AX)
94	double diffraction
95	low-$p_\perp$ production
Open heavy flavour: (also fourth generation)	
81	$f_i \bar{f}_i \to Q_k \overline{Q}_k$
82	$gg \to Q_k \overline{Q}_k$
83	$q_i f_j \to Q_k f_l$
84	$g\gamma \to Q_k \overline{Q}_k$
85	$\gamma\gamma \to F_k \overline{F}_k$
Closed heavy flavour:	
86	$gg \to J/\psi g$
87	$gg \to \chi_{0c} g$
88	$gg \to \chi_{1c} g$
89	$gg \to \chi_{2c} g$
104	$gg \to \chi_{0c}$
105	$gg \to \chi_{2c}$
106	$gg \to J/\psi\gamma$
107	$g\gamma \to J/\psi g$
108	$\gamma\gamma \to J/\psi\gamma$

No.	Subprocess
W/Z production:	
1	$f_i \bar{f}_i \to \gamma^*/Z^0$
2	$f_i \bar{f}_j \to W^\pm$
22	$f_i \bar{f}_i \to Z^0 Z^0$
23	$f_i \bar{f}_j \to Z^0 W^\pm$
25	$f_i \bar{f}_i \to W^+ W^-$
15	$f_i \bar{f}_i \to g Z^0$
16	$f_i \bar{f}_j \to g W^\pm$
30	$f_i g \to f_i Z^0$
31	$f_i g \to f_k W^\pm$
19	$f_i \bar{f}_i \to \gamma Z^0$
20	$f_i \bar{f}_j \to \gamma W^\pm$
35	$f_i \gamma \to f_i Z^0$
36	$f_i \gamma \to f_k W^\pm$
69	$\gamma\gamma \to W^+ W^-$
70	$\gamma W^\pm \to Z^0 W^\pm$
Prompt photons:	
14	$f_i \bar{f}_i \to g\gamma$
18	$f_i \bar{f}_i \to \gamma\gamma$
29	$f_i g \to f_i \gamma$
114	$gg \to \gamma\gamma$
115	$gg \to g\gamma$
Deeply Inel. Scatt.:	
10	$f_i f_j \to f_k f_l$
99	$\gamma^* q \to q$
Photon-induced:	
33	$f_i \gamma \to f_i g$
34	$f_i \gamma \to f_i \gamma$
54	$g\gamma \to f_k \bar{f}_k$
58	$\gamma\gamma \to f_k \bar{f}_k$
131	$f_i \gamma^*_T \to f_i g$

No.	Subprocess
132	$f_i \gamma^*_L \to f_i g$
133	$f_i \gamma^*_T \to f_i \gamma$
134	$f_i \gamma^*_L \to f_i \gamma$
135	$g \gamma^*_T \to f_i \bar{f}_i$
136	$g \gamma^*_L \to f_i \bar{f}_i$
137	$\gamma^*_T \gamma^*_T \to f_i \bar{f}_i$
138	$\gamma^*_T \gamma^*_L \to f_i \bar{f}_i$
139	$\gamma^*_L \gamma^*_T \to f_i \bar{f}_i$
140	$\gamma^*_L \gamma^*_L \to f_i \bar{f}_i$
80	$q_i \gamma \to q_k \pi^\pm$
Light SM Higgs:	
3	$f_i \bar{f}_i \to h^0$
24	$f_i \bar{f}_i \to Z^0 h^0$
26	$f_i \bar{f}_j \to W^\pm h^0$
102	$gg \to h^0$
103	$\gamma\gamma \to h^0$
110	$f_i \bar{f}_i \to \gamma h^0$
111	$f_i \bar{f}_i \to g h^0$
112	$f_i g \to f_i h^0$
113	$gg \to g h^0$
121	$gg \to Q_k \overline{Q}_k h^0$
122	$q_i \bar{q}_i \to Q_k \overline{Q}_k h^0$
123	$f_i f_j \to f_i f_j h^0$
124	$f_i f_j \to f_k f_l h^0$
Heavy SM Higgs:	
5	$Z^0 Z^0 \to h^0$
8	$W^+ W^- \to h^0$
71	$Z^0_L Z^0_L \to Z^0_L Z^0_L$
72	$Z^0_L Z^0_L \to W^+_L W^-_L$
73	$Z^0_L W^\pm_L \to Z^0_L W^\pm_L$
76	$W^+_L W^-_L \to Z^0_L Z^0_L$
77	$W^\pm_L W^\pm_L \to W^\pm_L W^\pm_L$

No.	Subprocess
BSM Neutral Higgses:	
151	$f_i\bar{f}_i \to H^0$
152	$gg \to H^0$
153	$\gamma\gamma \to H^0$
171	$f_i\bar{f}_i \to Z^0H^0$
172	$f_i\bar{f}_j \to W^\pm H^0$
173	$f_if_j \to f_if_jH^0$
174	$f_if_j \to f_kf_lH^0$
181	$gg \to Q_k\overline{Q}_kH^0$
182	$q_i\overline{q}_i \to Q_k\overline{Q}_kH^0$
183	$f_i\bar{f}_i \to gH^0$
184	$f_ig \to f_iH^0$
185	$gg \to gH^0$
156	$f_i\bar{f}_i \to A^0$
157	$gg \to A^0$
158	$\gamma\gamma \to A^0$
176	$f_i\bar{f}_i \to Z^0A^0$
177	$f_i\bar{f}_j \to W^\pm A^0$
178	$f_if_j \to f_if_jA^0$
179	$f_if_j \to f_kf_lA^0$
186	$gg \to Q_k\overline{Q}_kA^0$
187	$q_i\overline{q}_i \to Q_k\overline{Q}_kA^0$
188	$f_i\bar{f}_i \to gA^0$
189	$f_ig \to f_iA^0$
190	$gg \to gA^0$
Charged Higgs:	
143	$f_i\bar{f}_j \to H^+$
161	$f_ig \to f_kH^+$
Higgs pairs:	
297	$f_i\bar{f}_j \to H^\pm h^0$
298	$f_i\bar{f}_j \to H^\pm H^0$
299	$f_i\bar{f}_i \to A^0h^0$
300	$f_i\bar{f}_i \to A^0H^0$
301	$f_i\bar{f}_i \to H^+H^-$

No.	Subprocess
New gauge bosons:	
141	$f_i\bar{f}_i \to \gamma/Z^0/Z'^0$
142	$f_i\bar{f}_j \to W'^+$
144	$f_i\bar{f}_j \to R$
Leptoquarks:	
145	$q_i\ell_j \to L_Q$
162	$qg \to \ell L_Q$
163	$gg \to L_Q\overline{L}_Q$
164	$q_i\overline{q}_i \to L_Q\overline{L}_Q$
Technicolor:	
149	$gg \to \eta_{tc}$
191	$f_i\bar{f}_i \to \rho^0_{tc}$
192	$f_i\bar{f}_j \to \rho^+_{tc}$
193	$f_i\bar{f}_i \to \omega^0_{tc}$
194	$f_i\bar{f}_i \to f_k\bar{f}_k$
195	$f_i\bar{f}_j \to f_k\bar{f}_l$
361	$f_i\bar{f}_i \to W^+_L W^-_L$
362	$f_i\bar{f}_i \to W^\pm_L \pi^\mp_{tc}$
363	$f_i\bar{f}_i \to \pi^+_{tc}\pi^-_{tc}$
364	$f_i\bar{f}_i \to \gamma\pi^0_{tc}$
365	$f_i\bar{f}_i \to \gamma\pi'^0_{tc}$
366	$f_i\bar{f}_i \to Z^0\pi^0_{tc}$
367	$f_i\bar{f}_i \to Z^0\pi'^0_{tc}$
368	$f_i\bar{f}_i \to W^\pm\pi^\mp_{tc}$
370	$f_i\bar{f}_j \to W^\pm_L Z^0_L$
371	$f_i\bar{f}_j \to W^\pm_L \pi^0_{tc}$
372	$f_i\bar{f}_j \to \pi^\pm_{tc} Z^0_L$
373	$f_i\bar{f}_j \to \pi^\pm_{tc}\pi^0_{tc}$
374	$f_i\bar{f}_j \to \gamma\pi^\pm_{tc}$
375	$f_i\bar{f}_j \to Z^0\pi^\pm_{tc}$
376	$f_i\bar{f}_j \to W^\pm\pi^0_{tc}$
377	$f_i\bar{f}_j \to W^\pm\pi'^0_{tc}$

No.	Subprocess
Compositeness:	
146	$e\gamma \to e^*$
147	$dg \to d^*$
148	$ug \to u^*$
167	$q_iq_j \to d^*q_k$
168	$q_iq_j \to u^*q_k$
169	$q_i\overline{q}_i \to e^\pm e^{*\mp}$
165	$f_i\bar{f}_i(\to \gamma^*/Z^0) \to f_k\bar{f}_k$
166	$f_i\bar{f}_j(\to W^\pm) \to f_k\bar{f}_l$
Left–right symmetry:	
341	$\ell_i\ell_j \to H^{\pm\pm}_L$
342	$\ell_i\ell_j \to H^{\pm\pm}_R$
343	$\ell^\pm_i\gamma \to H^{\pm\pm}_L e^\mp$
344	$\ell^\pm_i\gamma \to H^{\pm\pm}_R e^\mp$
345	$\ell^\pm_i\gamma \to H^{\pm\pm}_L \mu^\mp$
346	$\ell^\pm_i\gamma \to H^{\pm\pm}_R \mu^\mp$
347	$\ell^\pm_i\gamma \to H^{\pm\pm}_L \tau^\mp$
348	$\ell^\pm_i\gamma \to H^{\pm\pm}_R \tau^\mp$
349	$f_i\bar{f}_i \to H^{++}_L H^{--}_L$
350	$f_i\bar{f}_i \to H^{++}_R H^{--}_R$
351	$f_if_j \to f_kf_lH^{\pm\pm}_L$
352	$f_if_j \to f_kf_lH^{\pm\pm}_R$
353	$f_i\bar{f}_i \to Z^0_R$
354	$f_i\bar{f}_j \to W^\pm_R$
Extra Dimensions:	
391	$f\bar{f} \to G^*$
392	$gg \to G^*$
393	$q\overline{q} \to gG^*$
394	$qg \to qG^*$
395	$gg \to gG^*$

No.	Subprocess
SUSY:	
201	$f_i\bar{f}_i \to \tilde{e}_L\tilde{e}_L^*$
202	$f_i\bar{f}_i \to \tilde{e}_R\tilde{e}_R^*$
203	$f_i\bar{f}_i \to \tilde{e}_L\tilde{e}_R^*+$
204	$f_i\bar{f}_i \to \tilde{\mu}_L\tilde{\mu}_L^*$
205	$f_i\bar{f}_i \to \tilde{\mu}_R\tilde{\mu}_R^*$
206	$f_i\bar{f}_i \to \tilde{\mu}_L\tilde{\mu}_R^*+$
207	$f_i\bar{f}_i \to \tilde{\tau}_1\tilde{\tau}_1^*$
208	$f_i\bar{f}_i \to \tilde{\tau}_2\tilde{\tau}_2^*$
209	$f_i\bar{f}_i \to \tilde{\tau}_1\tilde{\tau}_2^*+$
210	$f_i\bar{f}_j \to \tilde{\ell}_L\tilde{\nu}_\ell^*+$
211	$f_i\bar{f}_j \to \tilde{\tau}_1\tilde{\nu}_\tau^*+$
212	$f_i\bar{f}_j \to \tilde{\tau}_2\tilde{\nu}_\tau^*+$
213	$f_i\bar{f}_i \to \tilde{\nu}_\ell\tilde{\nu}_\ell{}^*$
214	$f_i\bar{f}_i \to \tilde{\nu}_\tau\tilde{\nu}_\tau^*$
216	$f_i\bar{f}_i \to \tilde{\chi}_1\tilde{\chi}_1$
217	$f_i\bar{f}_i \to \tilde{\chi}_2\tilde{\chi}_2$
218	$f_i\bar{f}_i \to \tilde{\chi}_3\tilde{\chi}_3$
219	$f_i\bar{f}_i \to \tilde{\chi}_4\tilde{\chi}_4$
220	$f_i\bar{f}_i \to \tilde{\chi}_1\tilde{\chi}_2$
221	$f_i\bar{f}_i \to \tilde{\chi}_1\tilde{\chi}_3$
222	$f_i\bar{f}_i \to \tilde{\chi}_1\tilde{\chi}_4$
223	$f_i\bar{f}_i \to \tilde{\chi}_2\tilde{\chi}_3$
224	$f_i\bar{f}_i \to \tilde{\chi}_2\tilde{\chi}_4$
225	$f_i\bar{f}_i \to \tilde{\chi}_3\tilde{\chi}_4$
226	$f_i\bar{f}_i \to \tilde{\chi}_1^\pm\tilde{\chi}_1^\mp$
227	$f_i\bar{f}_i \to \tilde{\chi}_2^\pm\tilde{\chi}_2^\mp$
228	$f_i\bar{f}_i \to \tilde{\chi}_1^\pm\tilde{\chi}_2^\mp$
229	$f_i\bar{f}_j \to \tilde{\chi}_1\tilde{\chi}_1^\pm$

No.	Subprocess
230	$f_i\bar{f}_j \to \tilde{\chi}_2\tilde{\chi}_1^\pm$
231	$f_i\bar{f}_j \to \tilde{\chi}_3\tilde{\chi}_1^\pm$
232	$f_i\bar{f}_j \to \tilde{\chi}_4\tilde{\chi}_1^\pm$
233	$f_i\bar{f}_j \to \tilde{\chi}_1\tilde{\chi}_2^\pm$
234	$f_i\bar{f}_j \to \tilde{\chi}_2\tilde{\chi}_2^\pm$
235	$f_i\bar{f}_j \to \tilde{\chi}_3\tilde{\chi}_2^\pm$
236	$f_i\bar{f}_j \to \tilde{\chi}_4\tilde{\chi}_2^\pm$
237	$f_i\bar{f}_i \to \tilde{g}\tilde{\chi}_1$
238	$f_i\bar{f}_i \to \tilde{g}\tilde{\chi}_2$
239	$f_i\bar{f}_i \to \tilde{g}\tilde{\chi}_3$
240	$f_i\bar{f}_i \to \tilde{g}\tilde{\chi}_4$
241	$f_i\bar{f}_j \to \tilde{g}\tilde{\chi}_1^\pm$
242	$f_i\bar{f}_j \to \tilde{g}\tilde{\chi}_2^\pm$
243	$f_i\bar{f}_i \to \tilde{g}\tilde{g}$
244	$gg \to \tilde{g}\tilde{g}$
246	$f_ig \to \tilde{q}_{iL}\tilde{\chi}_1$
247	$f_ig \to \tilde{q}_{iR}\tilde{\chi}_1$
248	$f_ig \to \tilde{q}_{iL}\tilde{\chi}_2$
249	$f_ig \to \tilde{q}_{iR}\tilde{\chi}_2$
250	$f_ig \to \tilde{q}_{iL}\tilde{\chi}_3$
251	$f_ig \to \tilde{q}_{iR}\tilde{\chi}_3$
252	$f_ig \to \tilde{q}_{iL}\tilde{\chi}_4$
253	$f_ig \to \tilde{q}_{iR}\tilde{\chi}_4$
254	$f_ig \to \tilde{q}_{jL}\tilde{\chi}_1^\pm$
256	$f_ig \to \tilde{q}_{jL}\tilde{\chi}_2^\pm$
258	$f_ig \to \tilde{q}_{iL}\tilde{g}$
259	$f_ig \to \tilde{q}_{iR}\tilde{g}$
261	$f_i\bar{f}_i \to \tilde{t}_1\tilde{t}_1^*$
262	$f_i\bar{f}_i \to \tilde{t}_2\tilde{t}_2^*$

No.	Subprocess
263	$f_i\bar{f}_i \to \tilde{t}_1\tilde{t}_2^*+$
264	$gg \to \tilde{t}_1\tilde{t}_1^*$
265	$gg \to \tilde{t}_2\tilde{t}_2^*$
271	$f_if_j \to \tilde{q}_{iL}\tilde{q}_{jL}$
272	$f_if_j \to \tilde{q}_{iR}\tilde{q}_{jR}$
273	$f_if_j \to \tilde{q}_{iL}\tilde{q}_{jR}+$
274	$f_i\bar{f}_j \to \tilde{q}_{iL}\tilde{q}_{jL}^*$
275	$f_i\bar{f}_j \to \tilde{q}_{iR}\tilde{q}_{jR}^*$
276	$f_i\bar{f}_j \to \tilde{q}_{iL}\tilde{q}_{jR}^*+$
277	$f_i\bar{f}_i \to \tilde{q}_{jL}\tilde{q}_{jL}^*$
278	$f_i\bar{f}_i \to \tilde{q}_{jR}\tilde{q}_{jR}^*$
279	$gg \to \tilde{q}_{iL}\tilde{q}_{i\,L}^*$
280	$gg \to \tilde{q}_{iR}\tilde{q}_{i\,R}^*$
281	$bq_i \to \tilde{b}_1\tilde{q}_{iL}$
282	$bq_i \to \tilde{b}_2\tilde{q}_{iR}$
283	$bq_i \to \tilde{b}_1\tilde{q}_{iR} + \tilde{b}_2\tilde{q}_{iL}$
284	$b\bar{q}_i \to \tilde{b}_1\tilde{q}_{i\,L}^*$
285	$b\bar{q}_i \to \tilde{b}_2\tilde{q}_{i\,R}^*$
286	$b\bar{q}_i \to \tilde{b}_1\tilde{q}_{i\,R}^* + \tilde{b}_2\tilde{q}_{i\,L}^*$
287	$q_i\bar{q}_i \to \tilde{b}_1\tilde{b}_1^*$
288	$q_i\bar{q}_i \to \tilde{b}_2\tilde{b}_2^*$
289	$gg \to \tilde{b}_1\tilde{b}_1^*$
290	$gg \to \tilde{b}_2\tilde{b}_2^*$
291	$bb \to \tilde{b}_1\tilde{b}_1$
292	$bb \to \tilde{b}_2\tilde{b}_2$
293	$bb \to \tilde{b}_1\tilde{b}_2$
294	$bg \to \tilde{b}_1\tilde{g}$
295	$bg \to \tilde{b}_2\tilde{g}$
296	$b\bar{b} \to \tilde{b}_1\tilde{b}_2^*+$

Index of Subprograms and Common Block Variables

This index is not intended to be complete, but gives the page where the main description begins of a subroutine, function, block data, common block, variable or array. For common block variables also the name of the common block is given. When some components of an array are described in a separate place, a special reference (indented with respect to the main one) is given for these components.

AQCDUP in HEPEUP 217
AQEDUP in HEPEUP 217
BRAT in PYDAT3 352
CHAF in PYDAT4 352
CKIN in PYSUBS 163
COEF in PYINT2 243
EBMUP in HEPRUP 211
HEPEUP common block 217
HEPEVT common block 61
HEPRUP common block 211
ICOL in PYINT2 244
ICOLUP in HEPEUP 219
IDBMUP in HEPRUP 211
IDPRUP in HEPEUP 217
IDUP in HEPEUP 218
IDWTUP in HEPRUP 212
IMSS in PYMSSM 193
ISET in PYINT2 242
ISIG in PYINT3 244
ISTUP in HEPEUP 218
K in PYJETS 56
KCHG in PYDAT2 347
KFDP in PYDAT3 352
KFIN in PYSUBS 162
KFPR in PYINT2 243
LPRUP in HEPRUP 216
MAXNUP in HEPEUP 217
MAXPUP in HEPRUP 211
MDCY in PYDAT3 349
MDME in PYDAT3 349
MINT in PYINT1 235
MOTHUP in HEPEUP 219
MRPY in PYDATR 45
MSEL in PYSUBS 161
MSTI in PYPARS 197
MSTJ in PYDAT1, main 335
MSTJ(38) - MSTJ(50) 272
MSTJ(101) - MSTJ(121) 77
MSTP in PYPARS, main 166
MSTP(22) 275
MSTP(61) - MSTP(71) 276
MSTP(81) - MSTP(94) 291
MSTP(131) - MSTP(134) 293
MSTU in PYDAT1, main 332
MSTU(1) and some more 363
MSTU(41) - MSTU(63) 380
MSTU(101) - MSTU(118) 190
MSTU(161) - MSTU(162) 381
MSUB in PYSUBS 162
MWID in PYINT4 244
N in PYJETS 56
NGEN in PYINT5 245
NPRUP in HEPRUP 216
NUP in HEPEUP 217
P in PYJETS 57
PARF in PYDAT2 347
PARI in PYPARS 200
PARJ in PYDAT1, main 339
PARJ(80) - PARJ(90) 275
PARJ(121) - PARJ(171) 80
PARJ(180) - PARJ(195) 193
PARP in PYPARS, main 185
PARP(61) - PARP(72) 277
PARP(81) - PARP(100) 294
PARP(131) 295
PARU in PYDAT1, main 335
PARU(41) - PARU(63) 381
PARU(101) - PARU(195) 190
PDFGUP in HEPRUP 211
PDFSUP in HEPRUP 211
PMAS in PYDAT2 347
PROC in PYINT6 246
PUP in HEPEUP 220
PY1ENT subroutine 327
PY2ENT subroutine 327
PY3ENT subroutine 328
PY4ENT subroutine 328
PY2FRM subroutine 229
PY4FRM subroutine 229
PY6FRM subroutine 231
PY4JET subroutine 232
PYADSH function 271
PYALEM function 189
PYALPS function 189
PYANGL function 331
PYBINS common block 385

PYBOEI subroutine 330
PYBOOK subroutine 383
PYCELL subroutine 375
PYCHGE function 330
PYCLUS subroutine 374
PYCOMP function 331
PYDAT1 common block 332
PYDAT2 common block 347
PYDAT3 common block 349
PYDAT4 common block 352
PYDATA block data 235
PYDATR common block 45
PYDCYK subroutine 345
PYDECY subroutine 330
PYDIFF subroutine 233
PYDISG subroutine 233
PYDOCU subroutine 234
PYDUMP subroutine 384
PYEDIT subroutine 358
PYEEVT subroutine 75
PYERRM subroutine 331
PYEVNT subroutine 158
PYEVWT subroutine 204
PYEXEC subroutine 330
PYFACT subroutine 383
PYFILL subroutine 383
PYFOWO subroutine 376
PYFRAM subroutine 159
PYGAGA subroutine 233
PYGAMM function 235
PYGANO function 234
PYGBEH function 234
PYGDIR function 234
PYGGAM function 234
PYGIVE subroutine 329
PYGVMD function 234
PYHEPC subroutine 63
PYHFTH function 234
PYHIST subroutine 384
PYI3AU subroutine 235
PYINBM subroutine 233
PYINDF subroutine 330
PYINIT subroutine 156
PYINKI subroutine 233
PYINPR subroutine 233
PYINRE subroutine 233
PYINT1 common block 235
PYINT2 common block 242
PYINT3 common block 244
PYINT4 common block 244
PYINT5 common block 245
PYINT6 common block 246
PYINT7 common block 246
PYINT8 common block 247
PYINT9 common block 247
PYJETS common block 55
PYJMAS subroutine 376
PYJOIN subroutine 328
PYK function 361
PYKCUT subroutine 159
PYKFDI subroutine 330
PYKFIN subroutine 345
PYKLIM subroutine 234
PYKMAP subroutine 234
PYLIST subroutine 360
PYLOGO subroutine 331
PYMAEL function 271
PYMASS function 330
PYMAXI subroutine 233
PYMEMX subroutine 271
PYMEWT subroutine 271
PYMRUN function 189
PYMSRV common block 196
PYMSSM common block 193
PYMULT subroutine 233
PYNAME subroutine 330
PYNMES subroutine 345
PYNULL subroutine 384
PYONIA subroutine 76
PYOPER subroutine 383
PYOFSH subroutine 234
PYP function 362
PYPARS common block 166, 197
PYPDEL subroutine 234
PYPDFL subroutine 234
PYPDFU subroutine 234
PYPDGA subroutine 234
PYPDPI subroutine 234
PYPDPR subroutine 234
PYPILE subroutine 233
PYPLOT subroutine 384
PYPREP subroutine 330
PYPTDI subroutine 330
PYQQBH subroutine 235
PYR function 44
PYRGET subroutine 45
PYRSET subroutine 45
PYRADK subroutine 76
PYRAND subroutine 233
PYRECO subroutine 235
PYREMN subroutine 233
PYRESD subroutine 233

PYROBO subroutine 358
PYSAVE subroutine 233
PYSCAT subroutine 233
PYSHOW subroutine 270
PYSIGH subroutine 234
PYSPEN function 235
PYSPHE subroutine 373
PYSPLI subroutine 234
PYSSMT common block 196
PYSSPA subroutine 271
PYSTAT subroutine 158
PYSTRF subroutine 330
PYSUBS common block 161
PYTABU subroutine 376
PYTAUD subroutine 331
PYTEST subroutine 25
PYTHRU subroutine 373
PYTIME subroutine 331
PYUPDA subroutine 346
PYWAUX subroutine 235
PYWIDT subroutine 234
PYX3JT subroutine 76
PYX4JT subroutine 76
PYXDIF subroutine 76
PYXJET subroutine 76
PYXKFL subroutine 76
PYXTEE subroutine 76
PYXTOT subroutine 233
PYZDIS subroutine 330
RMSS in PYMSSM 195
RRPY in PYDATR 46
RVLAM in PYMSRV 196
RVLAMB in PYMSRV 196
RVLAMP in PYMSRV 196
SCALUP in HEPEUP 217
SFMIX in PYSSMT 196
SIGH in PYINT3 244
SIGT in PYINT7 246
SMW in PYSSMT 196
SMZ in PYSSMT 196
SPINUP in HEPEUP 221
UPEVNT subroutine 216
UPINIT subroutine 210
V in PYJETS 57
VCKM in PYDAT2 349
VINT in PYINT1 238
VTIMUP in HEPEUP 220
VXPANH in PYINT9 247
VXPANL in PYINT9 247
VXPDGM in PYINT9 247
VXPVMD in PYINT9 247
WIDS in PYINT4 245
XERRUP in HEPRUP 216
XMAXUP in HEPRUP 216
XPANH in PYINT8 247
XPANL in PYINT8 247
XPBEH in PYINT8 247
XPDIR in PYINT8 247
XPVMD in PYINT8 247
XSEC in PYINT5 245
XSECUP in HEPRUP 216
XSFX in PYINT3 244
XWGTUP in HEPEUP 217
UMIX in PYSSMT 196
UMIXI in PYSSMT 196
VMIX in PYSSMT 196
VMIXI in PYSSMT 196
ZMIX in PYSSMT 196
ZMIXI in PYSSMT 196