

```
*****
*****
**
**                                     June 1990
**
**                               A Manual to
**
**                               The Lund Monte Carlo for Hadronic Processes
**
**                               PYTHIA version 5.4
**
**                               Hans-Uno Bengtsson
**                               Department of Theoretical Physics, University of Lund
**                               Solvegatan 14A, S-223 62 Lund, Sweden
**                               INTERNET address HANSUNO@THEP.LU.SE
**                               BITNET address THEPHUB@SELDC52
**                               Tel. +46 - 10 48 16
**                               Department of Physics, UCLA
**                               405 Hilgard Avenue, Los Angeles, CA 90024, USA
**                               BITNET address GOLLUM@UCLAHEP
**                               Tel. +213 - 825 - 5672
**
**                               Torbjorn Sjostrand
**                               CERN/TH, CH-1211 Geneva 23
**                               BITNET/EARN address TORSJO@CERNVM
**                               Tel. +22 - 767 28 20
**
**                               Copyright Hans-Uno Bengtsson and Torbjorn Sjostrand
**
*****
```

```
*****
*****
*
*   Table of Contents
*
*   1.   Introductory Material
*   1.1. Program Objective
*   1.2. Update History
*   1.3. Major Changes from PYTHIA 4.8 to 5.3
*   1.4. Major Changes from PYTHIA 5.3 to 5.4
*   1.5. Installation of Program
*   1.6. Programming Philosophy
*
*   2.   The Program Components
*   2.1. The Main Subroutines
*   2.2. The Physics Processes
*   2.3. Comments on Physics Processes
*   2.4. Switches for Event Type and Kinematics Selection
*   2.5. The General Switches and Parameters
*   2.6. General Event Information
*   2.7. The Event Record
*   2.8. Other Routines and Commonblocks
*   2.9. The JETSET Routines
*   2.10. On Cross-Sections
*   2.11. Examples
*
*   Acknowledgements
*
```

```

*           References
*
*****
* Legend:
* >=      larger than or equal to
* <=      smaller than or equal to
* /=      not equal to
* ->     goes to (rightarrow)
* ^       what follows next is to be read as an upper index
* _       what follows next is to be read as a lower index
* (D=..)  default value for commonblock parameter
* (R)     commonblock variable which user may read but not change
* (I)     commonblock variable for purely internal use
*****

```

1. Introductory Material

PYTHIA is a program intended for the study of high-pT physics in hadronic interactions, but also covers the domain of low-pT interactions as an integral part of the total cross-section. In its present form it includes hard scattering matrix elements, structure functions and initial and final state parton showers. Fragmentation is performed using the ordinary Lund fragmentation model, JETSET version 7.3, but an important task for PYTHIA is to set up the correct string configuration, particularly nontrivial for the low-pT target remnants.

1.1. Program Objective

Leaving aside for the moment a discussion of to what extent we can trust the various pieces that make up a full-fledged Monte Carlo program like PYTHIA, i.e., assuming that PYTHIA indeed generates events as they would occur if the underlying theory and assumptions were true, three important tasks can be singled out as the main objectives for PYTHIA:

- 1) to test the underlying theory by comparison with experiments at present colliders;
- 2) to help designing techniques and strategies for probing the standard physics at present and future colliders;
- 3) to help disentangle possible signals for new physics at present colliders by giving accurate estimates of standard model backgrounds.

Over the years, PYTHIA has indeed been put to all of these uses; in particular it has been extensively used to look at the possibilities of detection of predicted standard model particles like the top quark and the Higgs boson at the Superconducting Super Collider (SSC).

1.2. Update History

The PYTHIA program has undergone a continued rapid expansion, in terms of the number of subprocesses included, and also in terms of the physics aspects covered. This is a continuous process, with the

official numbered versions little more than snapshots of this process. Some versions have never been given a larger distribution, but have instead been handed to a few people for specific purposes. For the record, we below give a list of the versions, with some brief notes.

no	date	publ.	main new or improved features
1	Dec82	[Ben84]	synthesis of predecessors COMPTON, HIGHPT and KASSANDRA
2	-----		
3.1	-----		
3.2	-----		
3.3	Feb84	[Ben84a]	scale-breaking structure functions
3.4	Sep84	[Ben85]	more efficient kinematics selection
4.1	Dec84		initial and final state parton showers, W and Z
4.2	Jun85		multiple interactions
4.3	Aug85		W W, W Z, Z Z, R
4.4	Nov85		gamma W, gamma Z, gamma gamma
4.5	Jan86		H0 production, diffractive and elastic events
4.6	May86		angular correlation in resonance pair decays
4.7	May86		Z'0, H+
4.8	Jan87	[Ben87]	variable impact parameter in multiple interactions
4.9	May86		g H+
5.1	May86		massive matrix elements for heavy quarks
5.2	Jun87		intermediate boson scattering
5.3	Oct89		new particle and subprocess codes, new commonblock structure, new kinematics selection, some lepton-lepton and lepton-hadron interactions, new subprocesses
5.4	Jun90	[this]	s-dependent widths, resonances not on mass-shell, new processes, new structure functions

Versions preceding 4.8 by now should be considered obsolete. Versions 4.9, 5.1 and 5.2 are all minor expansions on 4.8, and have been given only limited distribution. Version 5.3 is therefore the first major revision since 4.8. Since the current version is (almost) backwards compatible with 5.3, and since several errors have been found in the original distribution of 5.3, users are recommended to switch from 5.3 to 5.4.

1.3. Major Changes from PYTHIA 4.8 to 5.3

Version 5.3 of PYTHIA, the Lund Monte Carlo for Hadronic Processes, represented a major break in continuity from a programming point of view, with no backwards compatibility to PYTHIA 4.8, the former standard version. Many major rewritings have been prompted by the adoption of the new particle numbering scheme developed under the aegis of the Particle Data Group [PDG88]. This scheme is also the standard adopted for JETSET 7. The internal scheme for subprocess classification has also been expanded to cover the future inclusion of further new processes. Also a number of other changes have been made, to systematize current features and leave space for new ones.

Here is a more extensive list of major changes in PYTHIA 5.3.

- New KF particle codes have been introduced and are consistently used

in the program.

- The program has, also in other respects, been updated and extended to run with JETSET 7.2. In particular, this applies to the commonblock LUJETS, which has been expanded to contain more event information.
 - The ISUB subprocess classification scheme has been reorganized and expanded.
 - A number of new subprocesses have been included: heavy quark production with massive matrix elements, gauge boson pair scattering, etc.
 - Many of the subprocesses can also be simulated in lepton-lepton interactions (and are prepared for lepton-hadron ones).
 - The possibility of event kinematics preselection has been added.
 - Switches for event flavour preselection have been reorganized and coordinated with JETSET.
 - Generation of kinematical variables has been changed and systematized.
 - It is possible to generate pileup events, where one single bunch crossing gives rise to several hadron-hadron interactions.
-

1.4. Major Changes from PYTHIA 5.3 to 5.4

The updates from version 5.3 to 5.4 are all minor, and just about any program that ran with version 5.3 will also work with PYTHIA 5.4. The following changes might give compatibility problems from programs based on PYTHIA 5.3:

- The routine PYTHIA has been renamed PYEVNT (to avoid confusion between the program-as-a-whole and one specific subroutine); a dummy routine PYTHIA which calls PYEVNT is provided, however.
- Subprocesses 142 and 143 have been moved to 143 and 144, respectively. The related MSEL codes 22 and 23 have also been moved to 23 and 24, respectively.
- The non-standard decay channel $Z \rightarrow H^+ + H^-$ has been removed, instead the decay $Z' \rightarrow H^+ + H^-$ has been introduced to fill the same function.
- The program should be run with JETSET version 7.3 rather than 7.2 (to take into account the new decay channels and the running of α_{em}). Note that many decay channels appear with new numbers.
- Several errors have been corrected, where previously the program gave wrong results.

In addition, the following changes have been made, where no compatibility problems are involved:

- A subprocess 142 has been added; in general the Z'/W' sector has been much expanded, and couplings to ordinary fermions and gauge bosons appear as free parameters (defined in JETSET).
- Four new structure function parametrization by Morfin and Tung come with the program, and two new by Gluck, Reya and Vogt.
- α_{em} is now taken to be running; if the user so decides it can be frozen, however.
- s-channel resonances are generated with s-hat-dependent widths; this should give an improved description of line shapes.
- In the decays $H \rightarrow Z Z$ or $W W$, the contribution to the total Higgs width is included also for Higgs so light that either or both products are off mass-shell.
- The couplings of H to quarks is taken with running quark masses,

- both for production $q + q_b \rightarrow H^0$ and decay $H^0 \rightarrow q + q_b$; see MSTP(37) for possibility to use fix masses.
- In a number of 2 \rightarrow 2 processes it is now possible to select specific mass ranges for resonances in the final state, and in particular to generate events below the nominal threshold mass.
 - Also for other decays of a resonance into two new resonances it is possible to set mass ranges on the decay products. This feature is not fully developed yet, however, so some limitations exist.
 - Process 22, $f + f_b \rightarrow Z^0 + Z^0$, has been expanded so that the two Z^0 by default represent the correct mixture of γ^*/Z^0 , with the possibility to switch to Z^0 only or γ^* only, according to MSTP(43) value.
 - The process $f + q \rightarrow f' + Q$ for single heavy flavour production by W exchange in the t channel has been implemented.
 - Some processes may now be used also in leptonproduction. This in particular includes ZZ, ZW and WW interactions, like in H production. PYTEST has been expanded to cover this possibility.
 - The table produced with PYSTAT(2) also gives the flavour (KF) codes and current masses of resonances, and the decay channel numbers in JETSET (IDC) for the channels listed. By default, fourth generation channels are not listed.
 - A number of internal administrative changes, which should not be visible to the outside user.
-

1.5. Installation of program

The program is written entirely in Fortran 77, and should run on any machine with such a compiler. The only external program required is JETSET 7.3. Comments on compiler optimization level, random number generators and machine precision problems are the same as given in the JETSET 7.3 manual. (Those wishing to use the IBM AUTODBL compiler option should note that a dummy variable MSEDUM should be inserted after MSEL in commonblock PYSUBS.)

SAVE statements have been included in accordance with the Fortran standard. Since most ordinary machines take SAVE for granted, this part is not particularly well tried out, however. Users on machines without automatic SAVE are therefore warned to be on the lookout for any variables which may have been missed.

A test program, PYTEST, is included in the PYTHIA package. It is disguised as a subroutine, so that the user has to run a main program

```
CALL PYTEST(0)
END
```

This program will generate some events of different types. If PYTHIA has not been properly installed, this program is likely to crash, or at least generate erroneous events. If everything works properly, as far as the program can judge, a final message to that effect is produced, else various error messages may appear. If PYTEST(1) is called instead of PYTEST(0), the same program is run through, but with more complete initialization and cross-section information, and with a listing of one event of each type generated. Finally, PYTEST(2) will in addition give an extensive listing of the initial search for cross-section coefficients and maxima; normally there is no reason to

use this latter option.

1.6. Programming Philosophy

The Monte Carlo program is built as a slave system, i.e. the user supplies the main program, and 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.

It should be noted that, while the physics content is obviously at the center of attention, the Lund Monte Carlo also contains a more extensive setup of auxiliary service routines than any other physics event generator. The hope is that this will provide a comfortable working environment, where not only events are generated, but users also linger on to perform a lot of the subsequent studies. (As for the relatively small attention given to physics in this manual, the reason is that the physics is documented separately in a series of papers, but the program pieces only here.)

The general rule is that all routines have names six characters long, beginning with PY. Also commonblocks have names starting with PY.

Before events can be generated, an initialization call is necessary, unlike the case of JETSET. Default values and other data are stored in BLOCK DATA PYDATA. Thus this subprogram, as well as BLOCK DATA LUDATA in JETSET, must be linked, which does not occur automatically with all loaders.

Apart from writing 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 the user to see to it that this unit is open for write.

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

A number of error checks are performed during execution. Serious errors, in particular those that may be found already at initialization time, lead to the program being aborted. Less serious errors may lead to the treatment of a particular event being cut short; see the JETSET manual. Consequences are unpredictable when using integer valued input variables with values not defined, or real-valued variables outside the physically sensible range. Users beware!

2. The Program Components

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 analyzed, using information stored in some commonblocks, 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.

In the presentation in this section, the ordering above is not strictly adhered to. In particular, information less important for an efficient use of PYTHIA has been delegated closer to the end.

- In subsection 2.1 the subroutines to be called by the user are introduced, particularly PYINIT, PYEVNT and PYSTAT.
- The ISUB classification code for subprocesses included in PYTHIA is tabulated in 2.2. Some comments on the physics of these subprocesses, and what special possibilities are open, are given in 2.3.
- The following two subsections, 2.4 and 2.5, deal with variables that can (with a few exceptions) be set only in the initialization phase, i.e. before the PYINIT call, if the default values should not be desirable. In 2.4 is collected the switches for the type of process to generate and kinematical constraints. Subsection 2.5 covers all the switches and parameters which regulate the details of the generation, such as choice of structure functions or Q^2 scale, on/off switches for parton showers or fragmentation, etc.
- The information available on each new event is covered in 2.6 and 2.7, in the former the general type of process and some kinematical variables, in the latter the detailed list of all particles generated.
- In 2.8 the further subroutines, functions and commonblocks in PYTHIA are listed, with brief information about their purpose.
- The fragmentation routines of JETSET 7.3 are amply documented elsewhere [Sjo90], but in 2.9 a brief reminder is given, with special emphasis on aspects of relevance when running PYTHIA.
- A few comments on cross-section calculations in the program are collected in 2.10.
- Finally, subsection 2.11 contains some examples on how to run the program.

2.1. The Main Subroutines

There are two routines that users 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 is frequently useful. The two other routines described here, PYFRAM

and PYKCUT, are of more specialized interest.

SUBROUTINE PYINIT(FRAME,BEAM,TARGET,WIN)

Purpose: to initialize the generation procedure.

FRAME : a character variable used to specify the frame of the experiment. Uppercase and lowercase letters may be freely mixed.

- = 'CMS' : colliding beam experiment in CM 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.
- = 'USER' : 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 LUJETS.

BEAM, TARGET : character variables to specify beam and target particles. Uppercase and lowercase letters may be freely mixed. An antiparticle may be denoted either by '~' or 'bar' at the end of the name. It is also possible to leave out the underscore ('_') directly after 'nu' in neutrino names, and the charge for proton and neutron.

- = 'p+' : proton.
- = 'p~- ' : antiproton.
- = 'n0' : neutron.
- = 'n~0' : antineutron.
- = 'pi+' : positive pion.
- = 'pi-' : negative pion.
- = 'e-' : electron.
- = 'e+' : positron.
- = 'nu_e' : electron neutrino.
- = 'nu_e~' : electron antineutrino.
- = 'mu-' : muon.
- = 'mu+' : antimuon.
- = 'nu_mu' : muon neutrino.
- = 'nu_mu~' : muon antineutrino.

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='USER' : dummy (information is taken from the P vectors, see above).

SUBROUTINE 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.)

Note: Previously this routine was called PYTHIA; it has now been renamed PYEVNT to avoid confusion between a subroutine and the program-as-a-whole. The dummy routine PYTHIA will be removed in version 5.5, but currently hands on control to PYEVNT (after warning the user the first time it is called).

SUBROUTINE 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 information desired.

- = 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 the user in PYKCUT.
- = 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 JETSET decay tables, the partial decay width, branching ratio and effective branching ratio (in the event some channels have been excluded by the user).
- = 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 the user in the current run, and a table of the cuts on variables used in the actual generation as derived from these user-defined cuts.
- = 5 : prints a table with all the values of the status codes MSTP(I) and the parameters PARP(I) used in the current run.

SUBROUTINE PYFRAM(IFRAME)

Purpose: to transform event between different frames, if so desired.

IFRAME : specification of frame the event is to be boosted to.

- = 1 : frame specified by user in the PYINIT call.
- = 2 : CM frame of incoming particles.

SUBROUTINE PYKCUT(MCUT)

Purpose: to enable a user to reject a given set of kinematic variables at an early stage of the generation procedure (before evaluation 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 the user requires it by setting MSTP(131) = 1, and never if "minimum bias" type events (class d of section 2.2) are to be generated as well.

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 commonblock (see section 2.8) 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 list of information given includes subprocess type ISUB, E_CM, s-hat, t-hat, u-hat, p_T-hat, x_1, x_2, x_F, tau, y*, tau', cos(theta-hat), and a few more. Some of these may not be relevant for the process under study, and are then set to zero.

2.2. The Physics Processes

A wide selection of fundamental $2 \rightarrow 1$ and $2 \rightarrow 2$ tree processes of the standard model (electroweak and strong) has been included in PYTHIA,

and slots are provided for those not yet implemented. In addition, a few "minimum bias" type processes (like elastic scattering), loop graphs, box graphs, 2 -> 3 tree graphs and some 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 -> 1 process (e.g. $q \bar{q} \rightarrow \gamma^* \rightarrow e^+ e^-$), but a 2 -> 2 cross-section may well have contributions from s-channel diagrams ($g g \rightarrow g g$ obtains contributions from $g g \rightarrow g^* \rightarrow g g$). Also, in the program, 2 -> 1 and 2 -> 2 graphs may sometimes be folded with two 1 -> 2 splittings to form effective 2 -> 3 or 2 -> 4 processes ($W^+ W^- \rightarrow H^0$ is folded with $q \rightarrow q' W$ to give $q \bar{q}' \rightarrow q'' \bar{q}''' H^0$).

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 this section, while its detailed use will be made clear in sections 2.4 and 2.6. Only processes marked with a + sign in the first column have been implemented in the program to date. The others are still given here, so that the user will easier understand how the classification works.

In the following f_i represents a fundamental fermion of flavour i , i.e. either of $d, u, s, c, b, t, l, h, e^-, \nu_e, \mu^-, \nu_\mu, \tau^-, \nu_\tau, \chi^-$ or ν_χ . A corresponding antifermion is denoted $f_{i\sim}$. In several cases, some classes of fermions are explicitly excluded, since they do not couple to the g or γ (no $e^+ e^- \rightarrow g g$, e.g.). Flavours appearing already in the initial state are denoted i and j , whereas new flavours in the final state are denoted k and l . 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 practice, the full γ^*/Z^0 structure is only included in subprocesses 1 and 22; for the other processes currently a Z^0 does not contain the γ^* piece. 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 an on-mass-shell photon.

In the last column below, references are given to works from which formulae have been taken. Sometimes these references are to the original work on the subject, sometimes only to the place where the formulae are given in the most convenient or accessible form. 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 etc.

Subprocess	References
a) 2 -> 1, tree	
+ 1 $f_i f_{i\sim} \rightarrow Z^0$	EHL84
+ 2 $f_i f_{j\sim} \rightarrow W^+$	EHL84
+ 3 $f_i f_{i\sim} \rightarrow H^0$	EHL84
4 $\gamma W^+ \rightarrow W^+$	
+ 5 $Z^0 Z^0 \rightarrow H^0$	EHL84

	6	$Z^0 W^+ \rightarrow W^+$	
	7	$W^+ W^- \rightarrow Z^0$	
+	8	$W^+ W^- \rightarrow H^0$	EHL84
b)	2	$\rightarrow 2$, tree	
+	11	$f_i f_j(\sim) \rightarrow f_i f_j(\sim)$	Cut78, Ben84
+	12	$f_i f_{i\sim} \rightarrow f_k f_{k\sim}$	Cut78, Ben84
+	13	$f_i f_{i\sim} \rightarrow g g$	Cut78, Ben84
+	14	$f_i f_{i\sim} \rightarrow g \gamma$	Hal78, Ben84
+	15	$f_i f_{i\sim} \rightarrow g Z^0$	EHL84
+	16	$f_i f_{j\sim} \rightarrow g W^+$	EHL84
	17	$f_i f_{i\sim} \rightarrow g H^0$	
+	18	$f_i f_{i\sim} \rightarrow \gamma \gamma$	EHL84
+	19	$f_i f_{i\sim} \rightarrow \gamma Z^0$	EHL84
+	20	$f_i f_{j\sim} \rightarrow \gamma W^+$	EHL84
	21	$f_i f_{i\sim} \rightarrow \gamma H^0$	
+	22	$f_i f_{i\sim} \rightarrow Z^0 Z^0$	EHL84, Gun86
+	23	$f_i f_{j\sim} \rightarrow Z^0 W^+$	EHL84, Gun86
+	24	$f_i f_{i\sim} \rightarrow Z^0 H^0$	EHL84
+	25	$f_i f_{i\sim} \rightarrow W^+ W^-$	EHL84, Gun86
+	26	$f_i f_{j\sim} \rightarrow W^+ H^0$	EHL84
	27	$f_i f_{i\sim} \rightarrow H^0 H^0$	
+	28	$f_i g \rightarrow f_i g$	Cut78, Ben84
+	29	$f_i g \rightarrow f_i \gamma$	Hal78, Ben84
+	30	$f_i g \rightarrow f_i Z^0$	EHL84
+	31	$f_i g \rightarrow f_k W^+$	EHL84
	32	$f_i g \rightarrow f_i H^0$	
	33	$f_i \gamma \rightarrow f_i g$	
	34	$f_i \gamma \rightarrow f_i \gamma$	
	35	$f_i \gamma \rightarrow f_i Z^0$	
	36	$f_i \gamma \rightarrow f_k W^+$	
	37	$f_i \gamma \rightarrow f_i H^0$	
	38	$f_i Z^0 \rightarrow f_i g$	
	39	$f_i Z^0 \rightarrow f_i \gamma$	
	40	$f_i Z^0 \rightarrow f_i Z^0$	
	41	$f_i Z^0 \rightarrow f_k W^+$	
	42	$f_i Z^0 \rightarrow f_i H^0$	
	43	$f_i W^+ \rightarrow f_k g$	
	44	$f_i W^+ \rightarrow f_k \gamma$	
	45	$f_i W^+ \rightarrow f_k Z^0$	
	46	$f_i W^+ \rightarrow f_k W^+$	
	47	$f_i W^+ \rightarrow f_k H^0$	
	48	$f_i H^0 \rightarrow f_i g$	
	49	$f_i H^0 \rightarrow f_i \gamma$	
	50	$f_i H^0 \rightarrow f_i Z^0$	
	51	$f_i H^0 \rightarrow f_k W^+$	
	52	$f_i H^0 \rightarrow f_i H^0$	
+	53	$g g \rightarrow f_k f_{k\sim}$	Cut78, Ben84
	54	$g \gamma \rightarrow f_k f_{k\sim}$	
	55	$g Z^0 \rightarrow f_k f_{k\sim}$	
	56	$g W^+ \rightarrow f_k f_{l\sim}$	
	57	$g H^0 \rightarrow f_k f_{k\sim}$	
	58	$\gamma \gamma \rightarrow f_k f_{k\sim}$	
	59	$\gamma Z^0 \rightarrow f_k f_{k\sim}$	
	60	$\gamma W^+ \rightarrow f_k f_{l\sim}$	
	61	$\gamma H^0 \rightarrow f_k f_{k\sim}$	

	62	$Z^0 Z^0 \rightarrow f_k f_{k\sim}$	
	63	$Z^0 W^+ \rightarrow f_k f_{l\sim}$	
	64	$Z^0 H^0 \rightarrow f_k f_{k\sim}$	
	65	$W^+ W^- \rightarrow f_k f_{k\sim}$	
	66	$W^+ H^0 \rightarrow f_k f_{l\sim}$	
	67	$H^0 H^0 \rightarrow f_k f_{k\sim}$	
+	68	$g g \rightarrow g g$	Cut78, Ben84
	69	$\gamma \gamma \rightarrow W^+ W^-$	
	70	$\gamma W^+ \rightarrow \gamma W^+$	
+	71	$Z^0 Z^0 \rightarrow Z^0 Z^0$	
+	72	$Z^0 Z^0 \rightarrow W^+ W^-$	
+	73	$Z^0 W^+ \rightarrow Z^0 W^+$	BLS87
	74	$Z^0 H^0 \rightarrow Z^0 H^0$	
	75	$W^+ W^- \rightarrow \gamma \gamma$	
+	76	$W^+ W^- \rightarrow Z^0 Z^0$	BLS87
+	77	$W^+ W^{+(-)} \rightarrow W^+ W^{+(-)}$	Dun86
	78	$W^+ H^0 \rightarrow W^+ H^0$	
	79	$H^0 H^0 \rightarrow H^0 H^0$	

c) $2 \rightarrow 2$, tree, massive final quarks

+	81	$f_i f_{i\sim} \rightarrow Q_i Q_{i\sim}$	Com77
+	82	$g g \rightarrow Q_i Q_{i\sim}$	Com77
+	83	$q_i f_j \rightarrow Q_k f_l$	Zer90

d) "minimum bias"

+	91	elastic scattering	Blo85
+	92	single diffraction	Gou83, Blo85
+	93	double diffraction	Gou83, Blo85
	94	central diffraction	
+	95	low-pT production	Sjo87

e) $2 \rightarrow 1$, loop

	101	$g g \rightarrow Z^0$	
+	102	$g g \rightarrow H^0$	EHL84

f) $2 \rightarrow 2$, box

+	111	$f_i f_{i\sim} \rightarrow g H^0$	Ell88
+	112	$f_i g \rightarrow f_i H^0$	Ell88
+	113	$g g \rightarrow g H^0$	Ell88
+	114	$g g \rightarrow \gamma \gamma$	Con71, Dic88
+	115	$g g \rightarrow g \gamma$	Con71, Dic88
	116	$g g \rightarrow \gamma Z^0$	
	117	$g g \rightarrow Z^0 Z^0$	
	118	$g g \rightarrow W^+ W^-$	

g) $2 \rightarrow 3$, tree

	121	$g g \rightarrow f_k f_{k\sim} H^0$	
--	-----	-------------------------------------	--

h) non-standard model, $2 \rightarrow 1$

+	141	$f_i f_{i\sim} \rightarrow Z'^0$	Alt89
+	142	$f_i f_{j\sim} \rightarrow W'^+$	Alt89
+	143	$f_i f_{j\sim} \rightarrow H^+$	
+	144	$f_i f_{j\sim} \rightarrow R$	Ben85a

i) non-standard model, $2 \rightarrow 2$

+ 161 f_i g -> f_k H+

For many of the subprocesses, further notes and qualifications may be of interest. These are given in the following section. Also note that some groups of subprocesses are available with the MSEL switch, see section 2.4.

2.3. Comments on Physics Processes

This section contains some useful comments on the processes included in the program, grouped by physics interest rather than sequentially by ISUB or 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, like 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 the following sections, 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.

2.3.1 QCD Jets

MSEL = 1,2
ISUB = 11,12,13,28,53,68

The basic cross-sections are taken from [Cut78]. However, 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 $1/N^2$. By default, these interference terms are excluded; however, they can be introduced by changing MSTP(34). In this case, the interference terms are distributed on the various colour flows according to the pole structure of the terms.

As an example, consider subprocess 28, $q + g \rightarrow q + g$. The total cross-section for this process, obtained by summing and squaring the Feynman s-, t-, and u-channel graphs, is [Cut78]:

$$2*(1 - u*s/t^2) - 4/9*(s/u + u/s) - 1.$$

(The hats on s, t and u have been suppressed, and an overall factor α_S^2/s^2 ignored.) Using the identity of the Mandelstam variables for the massless case:

$$s + t + u = 0,$$

this can be rewritten as

$$(s^2 + u^2)/t^2 - 4/9*(s/u - u/s).$$

On the other hand, the cross-sections for the two possible colour flows of this subprocess are [Ben84]:

$$A: 4/9*(2*u^2/t^2 - u/s)$$

$$B: 4/9*(2*s^2/t^2 - s/u),$$

the sum of which is:

$$8/9*(s^2 + u^2)/t^2 - 4/9*(s/u - u/s).$$

The difference between this expression and that of [Cut78], corresponding to the interference between the two colour flow configurations, is then

$$1/9*(s^2 + u^2)/t^2,$$

and can be naturally divided between colour flow A and B:

$$A: 1/9*u^2/t^2$$

$$B: 1/9*s^2/t^2.$$

This procedure is 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 mass-shell). As a consequence, some kind of regularization for $p_T \rightarrow 0$ is required. Normally the user is expected to set the desired p_{Tmin} value in CKIN(3).

The new flavour produced in the annihilation processes (ISUB = 12,53) is determined by the flavours allowed for gluon splitting into quark-antiquark; see switch MDME.

For production of massive quarks, see below.

2.3.2 Heavy Flavours

MSEL=4,5,6,7,8,35,36,37,38

ISUB = 81,82,83

The cross-sections are taken from [Com77] for ISUB = 81,82 and from [Zer90] for ISUB = 83. The former two processes are pure QCD ones, and normally dominate. The last process proceeds via t channel W exchange, and 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 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 $p_T \rightarrow 0$. It is therefore not necessary to introduce any special cuts.

The flavour produced is determined by the heaviest flavour allowed for gluon splitting into quark-antiquark; see switch MDME. When one of the MSEL options is used, MDME is automatically set by the program. Note that only one heavy flavour at a time is allowed; if more than one is turned on, only the heaviest will be produced (as opposed to the case for ISUB = 12,53 above, where more than one flavour is allowed simultaneously).

The lowest order processes above just represent one source of heavy flavour production. Heavy quarks can also be present in the structure functions at the Q^2 scale of the hard interaction, leading to processes like $Q + g \rightarrow Q + g$, so-called flavour excitation, or be created by gluon splittings $g \rightarrow Q + \bar{Q}$ in initial or final state shower evolution. In fact, as the CM energy is increased, these other

processes gain in importance relative to the lowest order production graphs above. As an example, only 10% of the b production at LHC energies come from the lowest order graphs. The figure is even smaller for charm, while it is at or above 50% for top. At LHC/SSC energies, the specialized treatment described in this subsection 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 with b/c 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_T event 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_T flow of the jets, and therefore do not survive normal experimental cuts.

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.

The cross-section for a heavy flavour pair close to threshold can be modified according to the formulae of [Fad89], see MSTP(35). This affects the total rate and also kinematical distributions.

2.3.3 Minimum Bias

MSEL = 1,2
ISUB = 91,92,93,95

The total and elastic cross-sections are given by the parametrizations in [Blo85]. Several different parametrizations are available; these can be selected with MSTP(31). For the single and double diffraction cross-sections, the ansatz in [Gou83] is used. The remaining part of the cross-section is automatically assigned to low-p_T events.

The simulation of the that variable in elastic or diffractive scattering is still fairly primitive. Combined with the imprecise kinematics treatment of small scattering angles, this means that the program should not be used for studies of the scattered (undiffracted) hadron. In diffractive scattering, the structure of the hadronic system selected may be regulated with MSTP(101). No high-p_T jet production in diffractive events is included so far.

The subprocess 95, low-p_T events, is somewhat unique, in that no meaningful physical borderline to high-p_T events can be defined. Even if the QCD 2 → 2 high-p_T processes are formally switched off, some of the events generated will be classified as belonging to this group, with a p_T spectrum of interactions to match the "minimum bias" event sample. Only with the option

MSTP(82) = 0 will subprocess 95 yield strictly low- p_T events, events which will then probably not be compatible with any experimental event sample. A number of options exists for the detailed structure of low- p_T events, see in particular MSTP(81) and MSTP(82). Details of the model(s) used for these events may be found in [Sjo87].

2.3.4 Prompt Photons

MSEL = 10
ISUB = 14,18,29,114,115

Processes ISUB = 14,29 give the main source of single gamma production, with ISUB = 115 giving an additional contribution which in some kinematics regions may become important. For gamma pair production, the process ISUB = 18 is often overshadowed in importance by ISUB = 114. Cross-sections for the Born term graphs 14, 18 and 29 are found e.g. in [EHL84], while the box graphs 114 and 115 are given in [Con71,Dic88].

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 timelike 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 with MSTJ(41) (see JETSET manual). Photon radiation off the spacelike incoming quark legs is not yet included, but should be of lesser importance for production at reasonably large p_T values.

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 shat 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 e.g. not give 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 $\text{abs}(\cos(\theta\text{-hat}))$ is close to unity. It is then also necessary to change MSTP(38) to 0.

2.3.5 Single Z/W Production

MSEL = 11,12,13,14,(21)
ISUB = 1,2,15,16,30,31,(141)

This group consists of 2 -> 1 processes, single resonance production,

and $2 \rightarrow 2$ processes, resonance + jets. With initial state parton showers turned on, the $2 \rightarrow 1$ processes also generate resonance + jets; in order to avoid double counting, the corresponding $2 \rightarrow 2$ processes should not be turned on simultaneously. The basic rule is to use the $2 \rightarrow 1$ processes for inclusive generation of Z/W, whereas the $2 \rightarrow 2$ processes should be used for study of the subsample with high transverse momentum.

The Z^0 of subprocess 1 includes the full interference structure Z^0/γ^* ; via MSTP(43) the user can select to produce only γ^* , only Z^0 , or the full Z^0/γ^* . 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, the former also includes the possibility of a decay into charged Higgses. The Z^0 of subprocesses 15 and 30 are currently pure Z^0 only.

For the $2 \rightarrow 1$ processes, the Breit-Wigner includes an s-hat-dependent width, which should provide an improved description of line shapes.

2.3.6 Neutral Higgs Production

MSEL = 16

ISUB = 3,5,8,24,26,(71,72,73,76,77),102,111,112,113

So far, only production of the standard model neutral Higgs is included. Many different processes can be involved, as seen from the list above. The proper choice depends on the actual Higgs mass, and (occasionally) on the desired search strategy.

For a Higgs which can still be handled not unreasonably well in a narrow width approximation, i.e. with a mass below 700 GeV, say, the production processes that are involved are ISUB = 3,5,8,102, as obtained for MSEL = 16. The subprocess $t \bar{t} \rightarrow H^0$ (a subset of the more general subprocess 3, but the only subset of importance for heavy Higgs production at hadronic colliders) is by now known to overestimate the cross-section for heavy Higgs production as compared to a more careful calculation based on the subprocess $g g \rightarrow t \bar{t} H^0$ (not yet implemented). This, however, is in general not a problem, since heavy Higgs production is anyway dominated by subprocesses 5, 8 and 102.

The subprocesses 5 and 8, $V V \rightarrow H^0$, which contribute to the processes $V V \rightarrow V' V'$ (V and V' intermediate vector bosons) show bad high energy behaviour, which can be cured only by the inclusion of all $V V \rightarrow V' V'$ 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 (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 subprocesses 71, 72, 76 and 77, an option is included (see MSTP(46)) whereby the user 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. $Z0Z0$ or $W+W-$) set via MDME. The difference is that the Breit-Wigner in subprocesses 5 and 8 contain an s-hat-dependent width, whereas the width in subprocesses 71 - 77 is calculated at nominal Higgs mass; also, higher order corrections to the widths are treated more accurately in subprocesses 5 and 8. Further, processes 71 - 77 assume on-mass-shell incoming W/Z, with associated kinematics factors, while processes 5 and 8 have W/Z correctly spacelike. All this leads to differences in the cross-sections by up to a factor 1.5. For subprocesses 71 - 77, also read comments in next subsection.

A subprocess like 113, with a Higgs recoiling against a gluon jet, is also effectively generated by initial state corrections to subprocess 102; thus, in order to avoid double counting, just as for the case of $Z0/W+$ above, these subprocesses should not be switched on simultaneously, but 3, 5, 8, and 102 be used for inclusive production of Higgs, and 111 - 113 for the study of the Higgs subsample with high transverse momentum.

Finally, the Higgs can also be produced in association with a $Z0/W$, ISUB = 24,26. These processes have a lower cross-section than single Higgs production, but are still of interest at the lower Higgs mass range because of a potentially better signal to background ratio.

The branching ratios of the Higgs are very strongly dependent on the actual mass. In principle, the program is set up to calculate these correctly at initialization. However, higher order corrections may at times be important and not fully unambiguous; see e.g. MSTP(37).

2.3.7 $\gamma/Z/W$ Pairs

MSEL = 15

ISUB = 19,20,22,23,25,71,72,73,76,77

This heading contains two different types of processes: fermion-antifermion annihilation into $\gamma/Z/W$ pairs, and scattering of intermediate vector bosons. Obviously other sources of $\gamma/Z/W$ pairs also may exist, like the Higgs or new gauge bosons.

The first set contains the standard $\gamma/Z/W$ produced by $f + \bar{f}$ ($f + \bar{f}'$) annihilation. We remind that the notation γ means a massless γ , while Z in principle stands for the full $\gamma^*/Z0$ structure. In fact, currently the $Z0$ appearing in subprocesses 19 and 23 are pure $Z0$ only, while the $Z0$ of process 22 contains the full interference structure, modifiable with MSTP(43). Currently subprocess 22 does not contain all possible contributions, but only those where both $\gamma^*/Z0$ are produced off the quark line, i.e. excluding e.g. emission of a γ^* off a $Z0$ decay product. This approximation is

fully valid for two on-mass-shell Z^0 's, but should give the dominant contribution also in more general situations. We also remind the user that the mass ranges of the two daughters may be set with the CKIN(41) - CKIN(44) parameters; this is particularly convenient e.g. to pick one Z almost on mass-shell and the other not.

The second set is of interest in that only the inclusion of the full set of $VV \rightarrow V'V'$ graphs will cure the bad high energy behaviour of $VV \rightarrow H^0$ (see 2.3.6). There is an option (see MSTP(46)) that selects only the s-channel Higgs exchange from subprocesses 71,72,76 and 77; with this option, these subprocesses will be essentially equivalent to subprocesses 5 and 8 if the Higgs is allowed to decay into Z or W (see, however, further comments in the Higgs subsection).

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.

WARNING 1: Subprocess 73, $Z^0 + W^{+/-} \rightarrow Z^0 + W^{+/-}$, presently gives a nonsensical cross-section and should not be used.

WARNING 2: For subprocess 77, the option for like-sign W scattering presently gives a non-sensical cross-section and should not be used; the default is set for W^+W^- scattering.

2.3.8 New Gauge Bosons

MSEL = 21,22,24
ISUB = 141,142,144

The Z' of subprocess 141 contains the full $\gamma^*/Z^0/Z'^0$ interference structure; the choice between different combinations is made via MSTP(44). The couplings of Z' to quarks and leptons can be set via PARU(121) - PARU(128), the coupling to W via PARU(129), and to H^+ via PARU(143). The coupling of the Z to H^+ is set via PARU(142). By a suitable setting of these parameters, it is possible to simulate several different physics scenarios. The default values agree with the 'extended gauge model' of [Alt89]. Further description of the coupling parameters are given in the JETSET 7.3 manual.

The W' of subprocess 142 so far does not contain interference with the standard model W - 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), the coupling to Z + W via PARU(135). Further comments as for Z' ; in particular, default couplings again agree with the 'extended gauge model' of [Alt89].

The R boson (particle code 40) 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 $s + d\bar{b} \rightarrow \mu^- + e^+$. Experimental limits on flavour changing neutral currents forces such a boson to be fairly heavy. The model implemented is the one described in [Ben85a].

2.3.8 Charged Higgs Production

MSEL = 23

ISUB = (141),143,161

The basic subprocess for charged Higgs production is ISUB = 143. Subprocess 161 gives the high- p_T tail of this distribution, so the two should not be used simultaneously (cf. Z only vs. Z + jet). The $\tan^2(\beta)$ parameter relevant for H^\pm coupling in the two Higgs doublet scenario is set via PARU(141).

Note in particular, that subprocess 141 is one possibility for H^\pm production, in addition to subprocesses 143 and 161. Note also, that it is only via subprocess 141 that a Z can be made to decay into charged Higgses. The coupling of the Z to $H^+ + H^-$ is regulated by PARU(142), and that of the Z' by PARU(143).

2.4. Switches for Event Type and Kinematics Selection

By default, only QCD $2 \rightarrow 2$ processes are generated by PYTHIA, composed of hard interactions above $p_{T\text{-hat_min}} = \text{PARP}(81) = 1.6$ GeV, with low- p_T processes added on so as to give the full (parametrized) inelastic, non-diffractive cross-section. With the help of the commonblock PYSUBS, it is possible to select for the generation of another process, or a 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 like resonance production (Z^0 , W^\pm , H^0 , Z'^0 , H^\pm or R^0) or QCD production of new flavours (ISUB = 12, 53, 81, 82), switches in the JETSET program may be used to this end; see section 2.9.

The CKIN array may be used to impose specific kinematics cuts. The user 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 $2 \rightarrow 1$ resonance production is performed in terms of the $m\text{-hat}$ 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 $2 \rightarrow 2$ processes, $\cos(\theta\text{-hat})$ is added as a third generation variable, and so additionally the range CKIN(27) - CKIN(28) may be restricted without any danger.

In addition to the variables found in PYSUBS, also those in the PYPARS commonblock may be used to select exactly what one wants to have simulated. These possibilities will be described in the following subsection.

The notation used above and in the following is that 'hat' denotes internal variables in the hard scattering subsystem, while '*' is for variables in the CM frame of the event-as-a-whole. 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 cutoffs that can be imposed on the generation process are therefore smeared, both by QCD radiation and by fragmentation. In a study of events within a given window of experimentally defined variables, it is up to the user 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 the user.

COMMON/PYSUBS/MSEL,MSUB(200),KFIN(2,-40:40),CKIN(200)

Purpose: to allow the user to run the program with any desired subset of processes, or restrict flavours or kinematics.

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 : QCD high- p_T processes switched on (ISUB = 11, 12, 13, 28, 53, 68); additionally low- p_T production if CKIN(3) < PARP(81) or PARP(82), depending on MSTP(82) (ISUB = 95). The CKIN cuts are here not used. If both incoming partons are leptons, the above default is not meaningful, and instead Z or W production (ISUB = 1 or 2) is used as default.
- = 2 : all QCD processes, including low- p_T , single and double diffractive and elastic scattering, are included (ISUB = 11, 12, 13, 28, 53, 68, 91, 92, 93, 95). The CKIN cuts are here not used. As with MSEL = 1, Z or W production is assumed if both incoming partons are leptons.
- = 4 : charm (cc~) production with massive matrix elements (ISUB = 81, 82).
- = 5 : bottom (bb~) production with massive matrix elements (ISUB = 81, 82).
- = 6 : top (tt~) production with massive matrix elements (ISUB = 81, 82).
- = 7 : low (ll~) production with massive matrix elements (ISUB = 81, 82).
- = 8 : high (hh~) production with massive matrix elements (ISUB = 81, 82).
- = 10 : prompt photons (ISUB = 14, 18, 29).
- = 11 : $Z\theta$ production (ISUB = 1).
- = 12 : $W^{+/-}$ production (ISUB = 2).
- = 13 : $Z\theta$ + jet production (ISUB = 15, 30).
- = 14 : $W^{+/-}$ + jet production (ISUB = 16, 31).
- = 15 : pair production of different combinations of gamma, $Z\theta$ and $W^{+/-}$ (except gamma + gamma; see MSEL = 10) (ISUB = 19, 20, 22, 23, 25).
- = 16 : $H\theta$ production (ISUB = 3, 5, 8, 102).
- = 17 : $H\theta$ + $Z\theta$ or $H\theta$ + $W^{+/-}$ (ISUB = 24, 26).
- = 21 : $Z'\theta$ production (ISUB = 141).
- = 22 : $W'^{+/-}$ production (ISUB = 142).
- = 23 : $H'^{+/-}$ production (ISUB = 143).
- = 24 : $R\theta$ production (ISUB = 144).
- = 35: single bottom production by W exchange (ISUB = 83).

- = 36: single top production by W exchange (ISUB = 83).
- = 37: single low production by W exchange (ISUB = 83).
- = 38: single high production by W exchange (ISUB = 83).

MSUB : (D=200*0) array to be set when MSEL = 0 (for MSEL >= 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 subsection 2.2 (with comments as given there).

MSUB(ISUB) = 0 : the subprocess is excluded.

MSUB(ISUB) = 1 : the subprocess is included.

Note: when MSEL >= 1 the MSUB values set by the user are never changed by PYTHIA. If the user wants to combine several different 'subruns', each with its own PYINIT call, into one single run, it is up to him 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 selectively to switch on and off contributions to the cross-sections from the different incoming partons/particles at the hard interaction. In combination with the JETSET resonance decay switches, this also allows the user 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 2.7, or the JETSET manual.

KFIN(I,J) = 0 : the parton/particle is forbidden.

KFIN(I,J) = 1 : the parton/particle is allowed.

By default, everything is on, except for J = 0, which does not have a physical meaning.

CKIN(1), CKIN(2) : (D=2.,-1.) range of allowed $m\text{-hat} = \sqrt{s\text{-hat}}$ values. IF CKIN(2) < 0., the upper limit is inactive.

CKIN(3), CKIN(4) : (D=0.,-1.) range of allowed $p_{T\text{-hat}}$ values for hard 2 -> 2 processes, with transverse momentum $p_{T\text{-hat}}$ defined in the rest frame of the hard interaction. If CKIN(4) < 0., the upper limit is inactive. For processes which are singular in the limit $p_{T\text{-hat}} \rightarrow 0$ (see CKIN(6)), CKIN(5) provides an additional constraint. The CKIN(3) and CKIN(4) limits can also be used in 2 -> 1 -> 2 processes. Here, however, the product masses are not known and hence assumed vanishing in the event selection. The actual $p_{T\text{-hat}}$ range for massive products is thus shifted downwards with respect to the nominal one.

CKIN(5) : (D=1.) lower cutoff on $p_{T\text{-hat}}$ values, in addition to the CKIN(3) cut above, for processes which are singular in the limit $p_{T\text{-hat}} \rightarrow 0$ (see CKIN(6)).

CKIN(6) : (D=1.) hard 2 -> 2 processes, which do not proceed only via an intermediate resonance (i.e. are 2 -> 1 -> 2 processes), are classified as singular in the limit $p_{T\text{-hat}} \rightarrow 0$ if either or both of the two final state products has a mass $m < \text{CKIN}(6)$.

CKIN(7), CKIN(8) : (D=-10.,10.) range of allowed scattering subsystem rapidities y^* in the CM frame of the event, where $y^* = (1/2) * \ln(x_1/x_2)$.

CKIN(9), CKIN(10) : (D=-10.,10.) range of allowed (true) rapidities for the product with largest rapidity in a 2 -> 2 or a 2 -> 1 -> 2 process, defined in the CM frame of the event,

i.e. $\max(y^*_3, y^*_4)$.

CKIN(11), CKIN(12) : (D=-10.,10.) range of allowed (true) rapidities for the product with smallest rapidity in a 2 -> 2 or a 2 -> 1 -> 2 process, defined in the CM frame of the event, i.e. $\min(y^*_3, y^*_4)$. Consistency thus requires CKIN(11) <= CKIN(9) and CKIN(12) <= CKIN(10).

CKIN(13), CKIN(14) : (D=-10.,10.) range of allowed pseudorapidities for the product with largest pseudorapidity in a 2 -> 2 or a 2 -> 1 -> 2 process, defined in the CM frame of the event, i.e. $\max(\eta^*_3, \eta^*_4)$.

CKIN(15), CKIN(16) : (D=-10.,10.) range of allowed pseudorapidities for the product with smallest pseudorapidity in a 2 -> 2 or a 2 -> 1 -> 2 process, defined in the CM frame of the event, i.e. $\min(\eta^*_3, \eta^*_4)$. Consistency thus requires CKIN(15) <= 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 -> 2 or a 2 -> 1 -> 2 process, defined in the CM 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 -> 2 or a 2 -> 1 -> 2 process, defined in the CM 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 $\cos(\theta\text{-hat})$ values in a hard 2 -> 2 scattering, where $\theta\text{-hat}$ is the scattering angle in the rest frame of the hard interaction.

CKIN(31), CKIN(32) : (D=2.,-1.) range of allowed $m'\text{-hat}$ values, where $m'\text{-hat}$ is the mass of the complete three- or four-body final state in 2 -> 3 or 2 -> 4 processes (while $m\text{-hat}$, 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(41) - CKIN(44) : (D=12.,-1.,12.,-1.) range of allowed mass values of the two (or one) resonances produced in a 'true' 2 -> 2 process, i.e. one not (only) proceeding through a single s-channel resonance (2 -> 1 -> 2). 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 mass-shell without applying any cuts. The exact interpretation of the CKIN variables depends on the flavours of the produced two resonances. For two resonances like Z + W (produced from $f + f' \rightarrow Z + W$), 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 given in the production process specification, see section 2.2. For two resonances like Z + Z, which are identical, or $W^+ + W^-$,

which are each other's antiparticles, one instead has
 $CKIN(41) < \min(m1, m2) < CKIN(42)$, and
 $CKIN(43) < \max(m1, m2) < CKIN(44)$.

In addition, whatever limits are set on $CKIN(1)$ and, in particular, $CKIN(2)$ obviously affects 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 $CKIN(42) < CKIN(41)$ or $CKIN(44) < CKIN(43)$ it means that the $CKIN(42)$ or $CKIN(44)$ limit is inactive.

Note 3: If limits are active and the resonances are identical, it is up to the user to ensure that $CKIN(41) \leq CKIN(43)$ and $CKIN(42) \leq CKIN(44)$.

Note 4: For identical resonances, it is not possible to preselect which of the resonances is the lighter one; if e.g. one Z is to decay to leptons and the other to quarks, there is no mechanism to guarantee that the lepton pair has a smaller mass than the quark one.

Note 5: The $CKIN$ values are applied to all relevant $2 \rightarrow 2$ 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, users are recommended to take proper care - if a Z is only allowed to decay into $b + \bar{b}$, e.g., setting its mass range to be $2 - 8$ GeV is obviously not a good idea.

Note 6: In principle, the machinery should work for any $2 \rightarrow 2$ process with resonances in the final state, but so far it has only been checked for processes 22 - 26, so also from this point some caution is urged.

$CKIN(45) - CKIN(48)$: ($D=12., -1., 12., -1.$) range of allowed mass values of the two (or one) secondary resonances produced in $2 \rightarrow 1 \rightarrow 2$ process (like $g + g \rightarrow H_0 \rightarrow Z_0 + Z_0$) or even a $2 \rightarrow 2 \rightarrow 4$ (or 3) process (like $q + \bar{q} \rightarrow Z_0 + H_0 \rightarrow Z_0 + W^+ + W^-$). 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 mass-shell. The description closely parallels the one given for $CKIN(41) - CKIN(44)$. Thus, for two resonances which are not identical or each other's antiparticles, one has

$CKIN(45) < m1 < CKIN(46)$, and
 $CKIN(47) < m2 < CKIN(48)$,

where $m1$ and $m2$ are the actually generated masses of the two resonances, and 1 and 2 are defined by the order in which they given in the decay channel specification in the program (see e.g. output from $PYSTAT(2)$ or $LULIST(12)$). For two resonances which are identical or each other's antiparticles, one instead has
 $CKIN(45) < \min(m1, m2) < CKIN(46)$, and
 $CKIN(47) < \max(m1, m2) < CKIN(48)$.

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 $H \rightarrow Z^0 + Z^0$ and $H \rightarrow W^+ + W^-$ have been fully implemented, such that an arbitrary mass range below the naive 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 makes an on-mass-shell decay impossible. These limitations will be relaxed in future versions.

2.5. The General Switches and Parameters

In addition to the event information described in section 2.6, the PYPARS commonblock contains the status code and parameters which 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.

COMMON/PYPARS/MSTP(200),PARP(200),MSTI(200),PARI(200)

Purpose: to give access to status code and parameters which regulate the performance of the program. 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(1) : (D=3) maximum number of generations. Automatically set ≤ 4 .

MSTP(2) : (D=1) calculation of α_{strong} at hard interaction, in the routine ULALPS.

= 0 : α_{strong} is fixed at value PARU(111).

= 1 : first order running α_{strong} .

= 2 : second order running α_{strong} .

MSTP(3) : (D=2) selection of Λ value in α_{strong} for MSTP(2) ≥ 1 .

= 1 : Λ is given by PARP(1) for hard interactions, by PARP(61) for spacelike showers and by PARJ(81) for timelike ones. This Λ is assumed 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 structure function parametrizations, i.e. $\Lambda = \text{PARP}(1)$ for user-defined structure functions, = 0.20 GeV for EHLQ1, = 0.29 GeV for EHLQ2, = 0.20 GeV for D01, = 0.40 GeV for D02, = 0.187 GeV for MT1, = 0.212 GeV for MT2, = 0.191 GeV for MT3, = 0.155 GeV for MT4, = 0.22 GeV for GRV1, = 0.16 GeV for GRV2, = 0.16 GeV for DFLM1, = 0.26 GeV for DFLM2 and = 0.36 GeV for DFLM3, respectively (cf. MSTP(51)). All the Λ values above 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 Q^2 argument still remains (see MSTP(32), MSTP(64) and MSTJ(44)). This Λ value is used also for MSTP(52) = 0, but the sensible choice here would be to use MSTP(2) = 0 and have no initial or final state radiation.

MSTP(31) : (D=1) parametrization of total and elastic cross-sections, nuclear slope parameter B and curvature C [Blo85].

= 1 : Block-Cahn fit 1 for cross-section, fit 1 for slope parameter.

= 2 : Block-Cahn fit 2 for cross-section, fit 1 for slope parameter.

- = 3 : Block-Cahn fit 3 for cross-section, fit 1 for slope parameter.
 - = 4 : Block-Cahn fit 6 for cross-section, fit 2 for slope parameter.
 - = 5 : Block-Cahn fit 8 for cross-section, fit 2 for slope parameter.
- Note: sets 1-3 for cross-section and set 1 for slope parameter are fits excluding recent measurements from $S_{pp \sim S}$, whereas sets 4-5 for cross-section and set 2 for slope parameter are fits including the $S_{pp \sim S}$ measurements.
- MSTP(32) : (D=2) Q^2 -definition in hard scattering for 2 \rightarrow 2 processes; for resonance production Q^2 is always chosen to be m_{res}^2 , where m_{res} is the mass of the resonance.
- = 1 : $Q^2 = 2 \cdot \hat{s} \cdot \hat{t} \cdot \hat{u} / (\hat{s}^2 + \hat{t}^2 + \hat{u}^2)$.
 - = 2 : $Q^2 = (m_{T1}^2 + m_{T2}^2) / 2$.
 - = 3 : $Q^2 = \min(-\hat{t}, -\hat{u})$.
 - = 4 : $Q^2 = \hat{s}$.
- 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_{strong} Q^2 argument, $\alpha_{strong} = \alpha_{strong}(PARP(33) \cdot Q^2)$.
- MSTP(34) : (D=0) use of interference term in matrix elements for QCD processes.
- = 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 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 and 82. The nonstandard options are mainly intended for top, but can be used, with less theoretical reliability, also for charm and bottom. See [Fad89] for details.
- = 0 : naive lowest order matrix element behaviour.
 - = 1 : enhancement or suppression close to threshold, according to the colour structure of the process, see [Fad89] for the expressions actually used. The α_{strong} value appearing in the threshold factor (which is not the same as the α_{strong} of the lowest order 2 \rightarrow 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_{strong} value appearing in the threshold factor is taken to be running, with argument $Q^2 = m_Q \sqrt{(m_{\hat{}} - 2m_Q)^2 + \Gamma_Q^2}$. Here m_Q is the nominal heavy quark mass, Γ_Q is the width of the heavy quark mass distribution, and $m_{\hat{}}$ is the invariant mass of the heavy quark pair. The Γ_Q value has to be stored by the user in PARP(35). The regularization of α_{strong} at low Q^2 is given by MSTP(36).
- MSTP(36) : (D=2) regularization of α_{strong} in the limit $Q^2 \rightarrow 0$ 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

($q + qb \rightarrow H0$) and decay ($H0 \rightarrow q + qb$) couplings.
 = 0 : not included, i.e. fix 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 (i.e. Higgs mass) scale given by PARP(37) times the quark mass itself.

MSTP(38) : (D=5) handling of quark loop masses in the box graphs $g + g \rightarrow \gamma + \gamma$ and $g + g \rightarrow g + \gamma$.
 = 0 : the program will for each flavour automatically choose the massless approximation for light quarks and the full massive formulae for heavy quarks, with the dividing line between light and heavy quarks dependent on the actual shat scale.
 >= 1 : 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.

WARNING: for =0, numerical instabilities may arise for scattering at small angles. Users are therefore recommended in this case to set CKIN(27) and CKIN(28) so as to exclude the range of scattering angles that are not of interest anyway.

MSTP(41) : (D=1) master switch for all resonance decays ($Z0$, $W+-$, $H0$, $Z'0$, $W'+-$, $H+-$, R).
 = 0 : off.
 = 1 : on.
 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 JETSET. However, since the MDCY values are overwritten in the PYINIT call, individual resonances should be switched off after the PYINIT call.

MSTP(42) : (D=1) mass treatment in $2 \rightarrow 2$ 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 mass-shell.
 = 1 : mass generated according to a Breit-Wigner.

MSTP(43) : (D=3) treatment of $Z0/\gamma^*$ interference in matrix elements. So far only implemented in subprocesses 1 and 22; in other processes what is called a $Z0$ is really a $Z0$ only, without the γ^* piece.
 = 1 : only γ^* included.
 = 2 : only $Z0$ included.
 = 3 : complete $Z0/\gamma^*$ structure (with interference) included.

MSTP(44) : (D=7) treatment of $Z'0/Z0/\gamma^*$ interference in matrix elements.
 = 1 : only γ^* included.
 = 2 : only $Z0$ included.
 = 3 : only $Z'0$ included.
 = 4 : only $Z0/\gamma^*$ (with interference) included.
 = 5 : only $Z'0/\gamma^*$ (with interference) included.
 = 6 : only $Z'0/Z0$ (with interference) included.
 = 7 : complete $Z'0/Z0/\gamma^*$ structure (with interference) included.

MSTP(45) : (D=2) treatment of $WW \rightarrow WW$ structure (ISUB = 77).
 = 1 : only $W+W+ \rightarrow W+W+$ and $W-W- \rightarrow W-W-$ included.

= 2 : only $W+W- \rightarrow W+W-$ included.
 = 3 : all charge combinations $WW \rightarrow WW$ included.
 MSTP(46) : (D=1) treatment of $VV \rightarrow V'V'$ structures (ISUB = 71 - 77).
 = 0 : only s-channel Higgs exchange included (where existing).
 = 1 : all graphs contributing to present $VV \rightarrow V'V'$ process included.
 Note: with the option MSTP(46) = 0, subprocesses 71 - 72 and 76 - 77 will essentially be equivalent to subprocesses 5 and 8, respectively, with the proper decay channels (i.e. only $Z0Z0$ or $W+W-$) set via MDME. The difference is that the Breit-Wigner in subprocesses 5 and 8 contain an s-hat-dependent width, whereas in subprocesses 71 - 72 and 76 - 77 the width is calculated at nominal $H0$ mass; also, subprocesses 5 and 8 treat higher order corrections to the widths more accurately.

MSTP(47) : (D=1) (C) angular orientation of decay products of resonances (Z, W, H, Z', W', R), either when produced singly or in pairs (also from a $H0$ decay), or in combination with a single quark, gluon or photon.
 = 0 : independent decay of each resonance, isotropic in CM frame of the resonance.
 = 1 : correlated decay angular distributions according to proper matrix elements, to the extent these are known.

MSTP(51) : (D=1) choice of structure functions; see also MSTP(52).
 Note: since all parametrizations have some region of applicability, the structure functions are assumed frozed below the lowest Q^2 and above the highest Q^2 covered by the parametrizations. The extrapolation to low x is covered by PARP(51).
 = 0 : user-supplied function, to be evolved in Tung program.
 = 1 : EHLQ set 1 (1986 updated version) for p, Owens set 1 for pi.
 = 2 : EHLQ set 2 (1986 updated version) for p, Owens set 2 for pi.
 = 3 : Duke-Owens set 1 for p, Owens set 1 for pi.
 = 4 : Duke-Owens set 2 for p, Owens set 2 for pi.
 = 5 : Morfin-Tung set 1 for p, Owens set 1 for pi.
 = 6 : Morfin-Tung set 2 for p, Owens set 1 for pi.
 = 7 : Morfin-Tung set 3 for p, Owens set 1 for pi.
 = 8 : Morfin-Tung set 4 for p, Owens set 1 for pi.
 = 9 : Gluck-Reya-Vogt L0 set for p, Owens set 1 for pi.
 = 10 : Gluck-Reya-Vogt H0 set for p, Owens set 1 for pi.
 = 11 : Diemoz-Ferroni-Longo-Martinelli set 1 ($\Lambda = 0.16$ GeV) for p, Owens set 1 for pi. Note that the DFLM parametrizations must be provided as an external file, and the interface in PYSTFE enabled (see the code for instructions).
 = 12 : Diemoz-Ferroni-Longo-Martinelli set 2 ($\Lambda = 0.26$ GeV) for p, Owens set 1 for pi. Comment as for =11.
 = 13 : Diemoz-Ferroni-Longo-Martinelli set 3 ($\Lambda = 0.36$ GeV) for p, Owens set 2 for pi. Comment as for =11.

MSTP(52) : (D=1) choice of Q^2 dependence in structure functions.
 = 0 : structure functions are evaluated at nominal lower cutoff value Q_0^2 , i.e. are made Q^2 -independent.
 = 1 : the parametrized Q^2 dependence is used.
 = 2 : the Tung program is called on to perform evolution of structure functions selected by MSTP(51). The results are not saved on file, but are used in current run.
 = 3 : as =2, but results are stored on file number MSTP(53).
 = 4 : evolution results from the Tung program are read in from file number MSTP(53) and are subsequently used.

MSTP(53) : (D=20) file number used for storing or reading evolved structure functions obtained with the Tung evolution program.
It is up to the user to see to it that this number is associated with a file, and that this file is properly opened for read/write.

MSTP(54) : (D=min(6,2*MSTP(1))) maximum number of quark flavours used in structure functions, and thus also for initial state spacelike showers. If some distributions (notably t) are absent in the parametrization selected in MSTP(51), these are obviously automatically excluded.

MSTP(61) : (D=1) (C) master switch for initial state QCD radiation.
= 0 : off.
= 1 : on.

MSTP(62) : (D=2) (C) level of coherence imposed on the spacelike 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 timelike) partons are both strictly ordered, increasing towards the hard interaction.

MSTP(63) : (D=2) (C) structure of associated timelike showers, i.e. showers initiated by emission off the incoming spacelike partons.
= 0 : no associated showers are allowed, i.e. emitted partons are put on mass-shell.
= 1 : a shower may evolve, with maximum allowed timelike virtuality set by the phase space only.
= 2 : a shower may evolve, with maximum allowed timelike virtuality set by phase space or by PARP(71) times the Q^2 value of the spacelike parton created in the same vertex, whichever is the stronger constraint.
= 3 : a shower may evolve, with maximum allowed timelike virtuality set by phase space or by the requirement of angular ordering, whichever is the stronger constraint. (Not yet implemented!)

MSTP(64) : (D=2) (C) choice of α_{strong} and Q^2 scale in spacelike parton showers.
= 0 : α_{strong} is taken to be fix at the value PARU(111).
= 1 : first order running α_{strong} with argument PARP(63)* Q^2 .
= 2 : first order running α_{strong} with argument
PARP(64)* $k_T^2 = \text{PARP}(64)*(1-z)*Q^2$.

MSTP(65) : (D=1) (C) treatment of soft gluon emission in spacelike 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(71) : (D=1) (C) master switch for final state QCD radiation.
= 0 : off.
= 1 : on.
Note: additional switches (e.g. for conventional/coherent showers) are available in JETSET, see MSTJ(41) - MSTJ(45), PARJ(81) and PARJ(82).

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_T values below $p_{T\text{min}}$, it also affects the choice

of structure for the one hard/semihard interaction.

- = 0 : simple two-string model without any hard interactions.
- = 1 : multiple interactions assuming the same probability in all events, with an abrupt p_{Tmin} cutoff at $PARP(81)$.
- = 2 : multiple interactions assuming the same probability in all events, with a continuous turnoff of the cross-section at $p_{T0} = PARP(82)$.
- = 3 : multiple interactions assuming a varying impact parameter and a hadronic matter overlap consistent with a Gaussian matter distribution, with a continuous turnoff of the cross-section at $p_{T0} = PARP(82)$.
- = 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 turnoff of the cross-section at $p_{T0} = PARP(82)$.

Note 1: For $MSTP(82) \geq 2$ and $CKIN(3) > PARP(82)$, 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, in particular, $PARP(82)$ values are sensitive to the choice of structure functions, Λ_{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 $PARP(81)$ value is consistent with the other default values give, i.e. EHLQ set 1 structure functions etc. When options $MSTP(82) = 2 - 4$ are used, the default $PARP(82)$ value is to be used in conjunction with $MSTP(2) = 2$ and $MSTP(33) = 3$. These switches should be set by the user.

$MSTP(83)$: ($D=100$) number of Monte Carlo generated phase space points per bin (whereof there are 20) in the initialization (in $PYINMU$) of multiple interactions for $MSTP(82) \geq 2$.

$MSTP(91)$: ($D=1$) (C) primordial k_T distribution.

- = 0 : no primordial k_T .
- = 1 : Gaussian, width given in $PARP(91)$.
- = 2 : exponential, width given in $PARP(92)$.

$MSTP(92)$: ($D=4$) (C) energy partitioning in hadron remnant. The energy fraction χ taken by one of the two objects, with conventions as described for $PARP(94) - PARP(97)$, is chosen according to the different distributions below. Here $c_{min} = 2*m_{quark}/\sqrt{s} = 0.6 \text{ GeV}/\sqrt{s}$.

- = 1 : 1 for meson, $2*(1-\chi)$ for baryon, i.e. simple counting rules.
- = 2 : $(k+1)*(1-\chi)^k$, with k as given in $PARP(94) - PARP(97)$.
- = 3 : as =2 for remnant splitting into hadron plus jet, but proportional to $(1-\chi)^k/(\chi^2+c_{min}^2)^{0.25}$ for remnant splitting into two jets, with k given by $PARP(94)$ or $PARP(96)$.
- = 4 : as =2 for remnant splitting into hadron plus jet, but proportional to $(1-\chi)^k/(\chi^2+c_{min}^2)^{0.5}$ for remnant splitting into two jets, with k given by $PARP(94)$ or $PARP(96)$.
- = 5 : as =2 for remnant splitting into hadron plus jet, but

proportional to $(1-\chi)^k / (\chi^2 + c_{\min}^2)^{(\text{PARP}(98)/2)}$ for remnant splitting into two jets, with k given by $\text{PARP}(94)$ or $\text{PARP}(96)$.

- MSTP(101) : (D=1) (C) structure of diffractive system.
= 1 : forward moving diquark + interacting quark.
= 2 : forward moving diquark + quark joined via interacting gluon ('hairpin' configuration).
- MSTP(111) : (D=1) (C) master switch for fragmentation and decay, as obtained with a LUEXEC call.
= 0 : off.
= 1 : on.
= -1 : only choose kinematical variables for hard scattering, i.e. no jets are defined. This is useful e.g. 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.
= 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 fragmentation, to compensate in parts for some numerical problems appearing at high energies.
= 0 : not performed.
= 1 : performed.
- MSTP(121) : (D=0) calculation of kinematics selection coefficients and differential cross-section maxima for subprocesses included (by user or default).
= 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 $\text{PARP}(121)$ (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 user in $\text{COEF}(\text{ISUB}, J)$ ($J = 1 - 20$) in $/\text{PYINT2}/$ and maximum of the corresponding differential cross-section times Jacobians in $\text{XSEC}(\text{ISUB}, 1)$ in $/\text{PYINT5}/$. This is to be done for each included subprocess ISUB before initialization, with the sum of all $\text{XSEC}(\text{ISUB}, 1)$ values, except for $\text{ISUB} = 95$, stored in $\text{XSEC}(0, 1)$.
- MSTP(122) : (D=1) initialization and differential cross-section maximization printout (see also MSTP(127)).
= 0 : none.
= 1 : short message.
= 2 : detailed message, including full maximization.
- MSTP(123) : (D=2) reaction to violation of maximum differential cross-section.
= 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 : CM frame of incoming particles.

MSTP(125) : (D=1) (C) documentation of partonic process, see section 'The event record' 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=20) number of lines in the beginning of event record that are reserved for event history information; see section 2.7. This value should never be reduced, but may be increased at a later date if more complicated processes are included.

MSTP(127) : (D=1) writing of header (version number and last date of change) on output file.
 = 0 : not done.
 = 1 : header is written at first PYINIT call, at which time MSTP(127) is set =0.

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(131) : (D=0) master switch for pileup 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 pileup events. The first event may be set up completely arbitrarily, using the switches in the PYSUBS commonblock, 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 pileup option.
 = 1 : low-p_T processes (ISUB = 95) only.
 = 2 : low-p_T + double diffractive processes (ISUB = 95 and 93).
 = 3 : low-p_T + double diffractive + single diffractive processes (ISUB = 95, 93 and 92).
 = 4 : low-p_T + double diffractive + single diffractive + elastic processes, together corresponding to the full hadron-hadron cross-section (ISUB = 95, 93, 92 and 91).

MSTP(133) : (D=0) multiplicity distribution of pileup events.
 = 0 : selected by user, before each PYEVNT call, by giving the MSTP(134) value.
 = 1 : a Poissonian multiplicity distribution in the total number of pileup 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 $\bar{n} = (\text{sum of cross-section for allowed processes}) * \text{PARP}(131)$. Since bunch crossings which do not give any events at all (probability $\exp(-\bar{n})$) are not simulated, the actual average number per PYEVNT call is $\langle n \rangle = \bar{n} / (1 - \exp(-\bar{n}))$.

= 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 σ_{hard} is the cross-section for this rare process (or the sum of the cross-sections of several rare processes) and σ_{soft} the cross-section for the processes allowed by MSTP(132), then define $\bar{n} = \sigma_{\text{soft}} * \text{PARP}(131)$ and $f = \sigma_{\text{hard}} / \sigma_{\text{soft}}$. The probability that a bunch crossing will give i events is then $\text{Prob}(i) = \exp(-\bar{n}) * \bar{n}^i / i! * f * i$ i.e. the naive 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 $\bar{n} < 120$, i.e. that an average beam crossing does not contain more than 120 pileup 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 pileup events, to be generated in the next PYEVNT call. May be reset for each new event, but must be in the range $1 \leq \text{MSTP}(134) \leq 200$.

MSTP(141) : (D=0) calling of PYKCUT in event generation chain, for inclusion of user-specified cuts.
= 0 : not called.
= 1 : called.

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 Lambda-QCD used in running alpha-strong for hard scattering (see MSTP(3)).

PARP(2) : (D=10. GeV) lowest CM energy for the event-as-a-whole that the program will accept to simulate.

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 `alpha_strong` at the hard interaction for `MSTP(33) = 3`. The effective K-factor thus obtained is in accordance with the results in [Ell86].

PARP(35) : (D=0.20) fix `alpha_strong` value used in heavy flavour threshold factor when `MSTP(35) = 1`.

PARP(36) : (D=0. GeV) the width `Gamma_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 naive fix quark mass in Higgs couplings is assumed defined; specifically the running quark mass is assumed to coincide with fix one at `PARP(37)` times the fix quark mass, i.e. $m_{\text{running}}(\text{PARP}(37)*m_{\text{fix}}) = m_{\text{fix}}$.

PARP(41) : (D=0.010 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 more narrow resonances are put on mass-shell.

PARP(42) : (D=2. GeV) minimum mass of resonances assumed allowed when evaluating total width of H^0 to $Z^0 + Z^0$ or $W^+ + W^-$ for cases when the H^0 is so light that (at least) one Z/W is forced to be off mass-shell. Also generally used as safety check on minimum mass of resonance.

PARP(43) : (D=0.10) precision parameter used in numerical integration of width into channel with at least one daughter off mass-shell.

PARP(51) : (D=1.) if structure functions for light flavours have to be extrapolated to x values lower than covered by the parametrizations, an $x^{(-\text{PARP}(51))}$ behaviour is assumed in that region.

PARP(52) : (D=2.26 GeV) initial Q -value used in Tung's evolution of structure functions.

PARP(53) : (D=1.E+04 GeV) maximum Q -value used in Tung's evolution of structure functions.

PARP(54) : (D=1.E-04) minimum x -value used in Tung's evolution of structure functions.

PARP(61) : (D=0.25 GeV) (C) Λ value used in spacelike parton shower (see `MSTP(64)`). This value may be overwritten, see `MSTP(3)`.

PARP(62) : (D=1. GeV) (C) effective cutoff Q or k_T value (see `MSTP(64)`), below which spacelike parton showers are not evolved.

PARP(63) : (D=0.25) (C) in spacelike shower evolution the virtuality Q^2 of a parton is multiplied by `PARP(63)` for use as a scale in `alpha_strong` and structure functions when `MSTP(64) = 1`.

PARP(64) : (D=1.) (C) in spacelike parton shower evolution the transverse momentum-squared evolution scale k_T^2 is multiplied by `PARP(64)` for use as a scale in `alpha_strong` and structure functions when `MSTP(64) = 2`.

PARP(65) : (D=2. GeV) (C) effective minimum energy (in CM frame) of timelike or on-shell parton emitted in spacelike shower; see also `PARP(66)`.

PARP(66) : (D=0.001) (C) effective lower cutoff on $1-z$ in spacelike showers, in addition to the cut implied by `PARP(65)`.

PARP(67) : (D=4.) (C) the Q^2 scale of the hard scattering (see `MSTP(32)`) is multiplied by `PARP(67)` to define the maximum parton virtuality allowed in spacelike showers. This does not apply to

s-channel resonances, where the maximum virtuality is set by m^2 .

PARP(71) : (D=4.) (C) the Q^2 scale of the hard scattering (see MSTP(32)) is multiplied by PARP(71) to define the maximum parton virtuality allowed in timelike showers. This does not apply to s-channel resonances, where the maximum virtuality is set by m^2 .

PARP(81) : (D=1.6 GeV/c) effective minimum transverse momentum p_{T_min} for multiple interactions with MSTP(82) = 1.

PARP(82) : (D=2.00 GeV) regularization scale p_{T_0} of the transverse momentum spectrum for multiple interactions with MSTP(82) ≥ 2 . Note that this default value is somewhat changed compared to the recommended PYTHIA 4.8 value of 1.96 due to a minor change of the assumed alpha-strong behaviour at small p_T .

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 $(1 - \text{PARP}(83)) \cdot \exp(-r^2) + (\text{PARP}(83) / \text{PARP}(84)^3) \cdot \exp(-r^2 / \text{PARP}(84)^2)$ 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 naive cross-section is replaced by a qq -one; this is used only for MSTP(82) ≥ 2 . The probability is parametrized as $\text{prob} = \text{PARP}(87) \cdot (1 - (p_{T^2} / (p_{T^2} + \text{PARP}(88) \cdot (\text{PARP}(82))^2))^2)$.

PARP(91) : (D=0.44 GeV/c) (C) width of Gaussian primordial k_T distribution for MSTP(91) = 1.

PARP(92) : (D=0.44 GeV/c) (C) width of exponential primordial k_T distribution for MSTP(91) = 2.

PARP(93) : (D=2. GeV/c) (C) upper cutoff for primordial k_T distribution.

PARP(94) : (D=1.) (C) for MSTP(92) ≥ 2 this gives the value of the parameter k for the case when a pion remnant is split into two fragments (which is which is chosen at random).

PARP(95) : (D=0.) (C) for MSTP(92) ≥ 2 this gives the value of the parameter k for the case when a pion 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(92) ≥ 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 $1/\chi^{\text{PARP}(98)}$ behaviour in the splitting distribution.

PARP(101) : (D=-0.02 GeV²) minimum value of t-hat in (diffractive and) elastic scattering.

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.

PARP(121) : (D=1.) the maxima obtained at initial maximization are multiplied by this factor if PARP(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 the 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 remaining fraction distributed according to the optimization results of PYMAXI.

PARP(131) : (D=0.01 mb⁽⁻¹⁾) in the pileup events scenario (see MSTP(131) - MSTP(133)), PARP(131) gives the assumed luminosity per bunch-bunch crossing, i.e. if a subprocess has a cross-section sigma, the average number of events of this type per bunch-bunch crossing is nbar = sigma * 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 pileup events, the initialization procedure will crash if nbar is above 120.

2.6. General Event Information

When an event is generated with PYEVNT, some information on this event is stored in the MSTI and PARI arrays of the LUDATA commonblock (often copied directly from the internal MINT and VINT variables). Further information is stored in the complete event record; see section 2.7.

Part of the information is only relevant for some subprocesses; by default everything irrelevant is set 0. Kindly note that, like the CKIN constraints described in section 2.4, kinematical variables normally (where it is not explicitly stated otherwise) refer to the naive 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 the accumulated statistics during the run.

MSTI(1) : specifies the general type of subprocess that has occurred, according to the ISUB code given in section 2.2.

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 -> 2 processes. With i = MSTI(15), j = MSTI(16) and k = MSTI(2), the QCD possibilities are, in the classification scheme of [Ben84]:

ISUB = 11, i = j, q_i q_i -> q_i q_i;
 k = 1 : colour configuration A.

k = 2 : colour configuration B.
 ISUB = 11, $i \neq j$, $q_i q_j \rightarrow q_i q_j$;
 k = 1 : only possibility.
 ISUB = 12, $q_i q_{\sim i} \rightarrow q_l q_{\sim l}$;
 k = 1 : only possibility.
 ISUB = 13, $q_i q_{\sim i} \rightarrow g g$;
 k = 1 : colour configuration A.
 k = 2 : colour configuration B.
 ISUB = 28, $q_i g \rightarrow q_i g$;
 k = 1 : colour configuration A.
 k = 2 : colour configuration B.
 ISUB = 53, $g g \rightarrow q_l q_{\sim l}$;
 k = 1 : colour configuration A.
 k = 2 : colour configuration B.
 ISUB = 68, $g g \rightarrow g g$;
 k = 1 : colour configuration A.
 k = 2 : colour configuration B.
 k = 3 : colour configuration C.
 ISUB = 83, $f q \rightarrow f' Q$ (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) : number of partons produced in the hard interactions, i.e.
 the number n of the 2 \rightarrow n matrix elements used; is sometimes 3 or
 4 when a basic 2 \rightarrow 1 or 2 \rightarrow 2 process has been folded with
 two 1 \rightarrow 2 initial branchings (like $q q' \rightarrow q'' q''' H^0$).

MSTI(4) : number of documentation lines in beginning of common block
 LUJETS that are given with $K(I,1) = 21$; 0 for $MSTP(125) = 0$.

MSTI(5) : number of events generated to date in current run.

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 \rightarrow 2 or 2 \rightarrow 1 \rightarrow 2 processes
 (else = 0).

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.

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 semihard scatterings that occurred in
 current event in the multiple interaction scenario; is = 0 for a
 low- p_T event.

MSTI(41) : the number of pileup events generated in latest PYEVNT
 call (including the first, 'hard' event).

MSTI(42) - MSTI(50) : ISUB codes for the events 2 - 10 generated in

the pileup 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 pileup events. If MSTI(41) is above 10, some ISUB codes will not appear anywhere.

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 event have been generated. The value stored here is based on all events up till the latest one generated.

PARI(2) : is the ratio PARI(1)/MSTI(5), i.e. the ratio of total integrated cross-section and number of events generated. Histograms generated with unit weight for events have to be multiplied by this factor, at the end of the run, to convert results to mb.

PARI(11) : E_CM, i.e. total CM energy.

PARI(12) : s, i.e. total CM energy-squared.

PARI(13) : m-hat = sqrt(s-hat), i.e. mass of the hard scattering subsystem.

PARI(14) : s-hat of the hard subprocess (2 -> 2 or 2 -> 1).

PARI(15) : t-hat of the hard subprocess (2 -> 2 or 2 -> 1 -> 2).

PARI(16) : u-hat of the hard subprocess (2 -> 2 or 2 -> 1 -> 2).

PARI(17) : p_T-hat of the hard subprocess (2 -> 2 or 2 -> 1 -> 2), evaluated in the rest frame of the hard interaction.

PARI(18) : p_T-hat^2 of the hard subprocess; see PARI(17).

PARI(19) : m'-hat, the mass of the complete three- or four-body final state in 2 -> 3 or 2 -> 4 processes (while m-hat, given in PARI(13), here corresponds to the one- or two-body central system).

Kinematically $m\text{-hat} \leq m'\text{-hat} \leq E\text{-CM}$.

PARI(20) : s'-hat = m'-hat^2; see PARI(19).

PARI(21) : Q of the hard subprocess, as used in structure functions and alpha_strong. The exact definition is process-dependent, see MSTP(32).

PARI(22) : Q^2 of the hard subprocess; see PARI(21).

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 structure functions.

PARI(35) : Feynman-x, $x_F = x_1 - x_2 = \text{PARI}(33) - \text{PARI}(34)$.

PARI(36) : tau = s-hat/s = x_1*x_2 = PARI(33)*PARI(34).

PARI(37) : y* = (1/2) * log(x_1/x_2), i.e. rapidity of the hard interaction subsystem in the CM frame of the event-as-a-whole.

PARI(38) : tau' = s'-hat/s = PARI(20)/PARI(12).

PARI(41) : cos(theta-hat), where theta-hat is the scattering angle of a 2 -> 2 (or 2 -> 1 -> 2) interaction, defined in the rest frame of the hard scattering subsystem.

PARI(42) : x_T, i.e. scaled transverse momentum of the hard scattering subprocess, $x_T = 2 * p_T\text{-hat}/E\text{-CM}$.

PARI(43), PARI(44) : x_L3 and x_L4, i.e. longitudinal momentum fractions of the two scattered partons, in the range $-1 < x_L < 1$, in the CM 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 CM 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 CM 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 CM 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 CM 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 CM 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 CM 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. $2 * \text{PARI}(17)$.

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_T .

PARI(67) : sum of transverse momenta of partons generated at hard interactions, excluding the hardest one (see PARI(65)), and also excluding initial and final state radiation. Is nonvanishing 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. Is nonvanishing only in the multiple interaction scenario.

PARI(69) : sum of transverse momenta of all partons generated in hard interactions ($\text{PARI}(66) + \text{PARI}(68)$) 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 side 2, excluding those of the hardest interaction. Is nonvanishing 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 nonvanishing only in the multiple interaction scenario.

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 n_{bar} of pileup events, defined as $(\text{cross-section for processes allowed by MSTP}(132)) * \text{PARP}(131)$. Only relevant for $\text{MSTP}(133) = 1$ or 2 .

PARI(92) : average multiplicity of pileup events as actually simulated, i.e. with multiplicity = 0 events removed and the high-end tail truncated. Only relevant for $\text{MSTP}(133) = 1$ or 2 .

PARI(93) : for $\text{MSTP}(133) = 1$ it is the probability that a beam crossing will produce a pileup event at all, i.e. that there will be at least one hadron-hadron interaction; for $\text{MSTP}(133) = 2$ the probability that a beam crossing will produce a pileup event with one hadron-hadron interaction of the desired rare type.

2.7. The Event Record

When an event is generated by a PYEVNT call, it is stored in the common block LUJETS. Here each parton and particle is represented by one line of information, giving a status code, parton/particle code, line of mother, lines of first and last daughter (or colour flow information in hard interaction and parton showers), momentum, energy, mass, production vertex and time, and invariant lifetime (if unstable). The LUJETS common block is used extensively both by the PYTHIA and the JETSET routines; indeed it provides the bridge that allows the general utility routines in JETSET to be used also for PYTHIA events. A detailed description of LUJETS is found in the JETSET 7.3 manual, a companion file to the program. 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 JETSET manual. Therefore we here give a brief summary on the general structure of the LUJETS common block, followed by details specific to PYTHIA.

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

N : number of entries, i.e. number of lines used in the K, P and V matrices, in the current event.

K(I,1) : status code KS for the parton/particle stored in the line.

= 0 : empty line.

= 1 - 10 : undecayed particle or unfragmented parton.

= 11 - 20 : decayed particle or fragmented or showered parton.

= 21 : documentation lines for compressed event summary (see below).

K(I,2) : KF flavour code for partons and particles, following the 1988 Particle Data Group numbering conventions [PDG88]. Some of the most frequently used ones are

1	d	11	e-	21	g	31	
2	u	12	nu_e	22	gamma	32	Z'0
3	s	13	mu-	23	Z0	33	
4	c	14	nu_mu	24	W+	34	W'+
5	b	15	tau-	25	H0	35	
6	t	16	nu_tau	26		36	
7	l	17	chi-	27		37	H+
8	h	18	nu_chi	28		38	
9		19		29		39	
10		20		30		40	R0
1103	dd_1	2103	ud_1	3101	sd_0	3101	su_0
2101	ud_0	2203	uu_1	3103	sd_1	3103	su_1
211	pi+	213	rho+	2112	n	1114	Delta-
311	K0	313	K*0	2212	p	2114	Delta0
321	K+	323	K*+	3122	Lambda0	2214	Delta+
411	D+	413	D*+	3112	Sigma-	2224	Delta++
421	D0	423	D*0	3212	Sigma0	3114	Sigma*-
431	D_s+	433	D*_s+	3222	Sigma+	3214	Sigma*0
130	K_L0			3312	Xi-	3224	Sigma*+
310	K_S0			3322	Xi0	3334	Omega-

A negative KF code, where existing, always corresponds to the antiparticle of the one listed above.

To designate diffractive states, three non-standard codes are made available by JETSET.

210 pi_diff+ 2110 n_diffr 2210 p_diffr
 K(I,3) : line number of parent parton or jet, where known, else 0.
 K(I,4) : normally line number of the first daughter; for K(I,1) = 3, 13
 or 14 instead special colour flow information.
 K(I,5) : normally line number of the last daughter; for K(I,1) = 3, 13
 or 14 instead special colour flow information.

 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/c²). Spacelike partons are
 given with P(I,5) = -sqrt(-m²).

 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 = 3.33*10⁻¹² s).
 V(I,5) : proper lifetime of the particle/parton (in mm/c = 3.33*10⁻¹²
 s); is 0 for a stable entry.

In most instances, only the actual partons and particles 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 $q \bar{q} \rightarrow g \gamma$ or $u \bar{u} \rightarrow Z^0 \rightarrow e^+ e^-$) 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 in 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 hadrons. When listed with LULIST, these two lines will be separated from subsequent lines by a sequence of ===== signs, to improve readability. For diffractive and elastic events, the two outgoing states in lines 3 and 4 completes the list. Otherwise, lines 3 and 4 contain the two partons that initiate the two initial state parton showers, and 5 and 6 the endproducts 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 coincide. For a simple 2 -> 2 hard scattering, lines 7 and 8 give the two outgoing partons/particles from the hard interaction, before any final state radiation. For 2 -> 2 processes proceeding via an intermediate resonance such as Z⁰/gamma*, W^{+/-}, H⁰ or R, the resonance is found in 7 and the two outgoing partons/particles in 8 and 9. In some cases one or both of these may be a resonance in its own right, 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 on 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. A special case is provided by $W^+ W^-$ or $Z^0 Z^0$ fusion to a 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 spacelike, what is actually listed as the mass for them is $-\sqrt{-m^2}$. 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 $p + p \rightarrow p_{\text{diff}} + p$, 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 p~
=====				
3	21	27	1	outgoing p_diff
4	21	-2212	2	outgoing p~
=====				

The typical QCD $2 \rightarrow 2$ 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 p~
=====				
3	21	2	1	u picked from incoming p
4	21	-1	2	d~ picked from incoming p~
5	21	21	3	u evolved to g at hard scattering
6	21	-1	4	still d~ at hard scattering
7	21	21	0	outgoing g from hard scattering
8	21	-1	0	outgoing d~ from hard scattering
=====				

Note that, where well defined, the $K(I,3)$ code does contain information on 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. $g g \rightarrow g g$.

As a final example, $W^+ W^-$ fusion to a H^0 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 c~
7	21	24	5	spacelike W^+ emitted by u quark
8	21	-24	6	spacelike W^- emitted by 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	c~ turned into s~ by emission of W^-
12	21	23	9	first Z^0 coming from decay of H^0

13	21	23	9	second Z0 coming from decay of H0
14	21	12	12	electron neutrino from first Z0 decay
15	21	-12	12	electron antineutrino from first Z0 decay
16	21	5	13	b quark from second Z0 decay
17	21	-5	13	b~ quark from second Z0 decay

=====

After these lines with 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. 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. The extra pairs of partons that are generated by multiple interactions do not point back to anything, i.e. they have $K(I,3) = 0$.

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 can not 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 20 (can be changed in $MSTP(126)$). From line 21 and onwards each parton with $K(I,1) = 3, 13$ or 14 appear with special colour flow information in the $K(I,4)$ and $K(I,5)$ positions; see the JETSET 7.3 manual. For an ordinary $2 \rightarrow 2$ scattering, the two incoming partons at the hard scattering are stored in line 21 and 22, and the two outgoing in 23 and 24. The colour flow between these partons has to be chosen according to the proper relative probabilities in cases when many alternatives are possible, see [Ben84]. If there is initial state radiation, the two partons in line 21 and 22 are copied down to line 25 and 26, from which the initial state showers are reconstructed backwards step by step. The branching history may be read by noting that, for a branching $a \rightarrow b + c$, 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 timelike partons c may initiate timelike 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 $c \rightarrow d + e$, with d and e pointing towards c . The mass of timelike partons is properly stored in $P(I,5)$, for spacelike partons instead $-\sqrt{-m^2}$ is stored. 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.

2.8. Other Routines and Commonblocks

The subroutines and commonblocks that a user will come in direct contact with have already been described. A number of other routines and commonblocks exist, and are here briefly listed for the sake of completeness.

SUBROUTINE PYINKI

Purpose: to initialize the kinematics given by the two incoming hadrons/leptons.

SUBROUTINE PYINRE

Purpose: to initialize the widths and effective widths of resonances.

SUBROUTINE PYXTOT

Purpose: to give the parametrized total, double diffractive, single diffractive and elastic cross-sections for different energies and colliding hadrons.

SUBROUTINE PYMAXI

Purpose: 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

Purpose: to determine the number of pileup events, i.e. events appearing in the same beam-beam crossing.

SUBROUTINE PYRAND

Purpose: to generate the quantities characterizing a hard scattering on the parton level, according to the relevant matrix elements.

SUBROUTINE PYSCAT

Purpose: to find outgoing flavours and to set up the kinematics and colour flow of the hard scattering.

SUBROUTINE PYSSPA(IP1,IP2)

Purpose: to generate the spacelike showers of the initial state radiation.

SUBROUTINE PYRESO

Purpose: to allow Z^0 , $W^{+/-}$, H^0 , Z'^0 , $H^{+/-}$ and R^0 resonances to decay, including chains of successive decays and parton showers.

SUBROUTINE PYMULT

Purpose: to generate semihard interactions according to the multiple interaction formalism.

SUBROUTINE PYREMN(IPU1,IPU2)

Purpose: to add on target remnants and include primordial k_T .

SUBROUTINE PYDIFF

Purpose: to handle diffractive and elastic scattering events.

SUBROUTINE PYDOCU

Purpose: 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

Purpose: to calculate widths and effective widths of resonances.

SUBROUTINE PYKLIM

Purpose: to calculate allowed kinematical limits.

SUBROUTINE PYKMAP

Purpose: to calculate the value of a kinematical variable when this is selected according to one of the simple pieces.

Purpose: to give the differential cross-section (multiplied by the relevant Jacobians) for a given subprocess and kinematical setup.

SUBROUTINE PYSTFU(KF,X,Q2,XPQ)

Purpose: to give gluon, quark and antiquark structure functions for given x and Q^2 values for p , p^- , n , n^- , π^+ or π^- .

SUBROUTINE PYSPLI

Purpose: to give hadron remnant or remnants left when the reacting parton is kicked out.

FUNCTION PYGAMM(X)

Purpose: to give the value of the ordinary gamma function $\Gamma(x)$ (used in some structure function parametrizations).

SUBROUTINE PYWAUX(IAUX,EPS,WRE,WIM)

SUBROUTINE PYI3AU(EPS,RAT,Y3RE,Y3IM)

Purpose: to calculate the value of some auxiliary functions appearing in the cross-section expressions in PYSIGH.

FUNCTION PYSPEN(XREIN,XIMIN,IREIM)

Purpose: to calculate the real and imaginary part of the Spence function.

BLOCK DATA PYDATA

Purpose: to give sensible default values to all status codes and parameters.

SUBROUTINE PYSTFE(KF,X,Q2,XPQ)

Purpose: to provide a dummy routine, which can be used as an interface to an external set of structure functions, according to the wishes of the user. The routine actually provided with PYTHIA contains an interface to the Tung evolution package [Tun87] and the Diemoz-Ferroni-Longo-Martinelli structure function parametrizations [DFL88]. Some lines have been commented out, so as to avoid problems with unresolved external references when either of these packages are not linked (in addition, for DFLM one also needs to

link the CERN library KERNLIB package). How to enable the commented sections is clearly marked in the subroutine code. The rules for input (KF, X and Q2) and output (XPQ) variables must be adhered to.

KF : hadron flavour code, 2212 for proton and 211 for pi+. PYSTFU, from which PYSTFE is called, will automatically perform isospin conjugation for neutron and charge conjugation for antiproton, antineutron and pi-.

X : x value.

Q2 : Q² value.

XPQ(-6:6) : array of structure function values for different flavours, multiplied by x. Thus $XPQ(i) = x * f_i(x, Q^2)$. Code follows normal KF one, except that gluon structure function is put at $i = 0$:

XPQ(0) = xg(x,Q²),
 XPQ(1) = xd(x,Q²), XPQ(-1) = xd~(x,Q²),
 XPQ(2) = xu(x,Q²), XPQ(-2) = xu~(x,Q²),
 XPQ(3) = xs(x,Q²), XPQ(-3) = xs~(x,Q²),
 XPQ(4) = xc(x,Q²), XPQ(-4) = xc~(x,Q²),
 XPQ(5) = xb(x,Q²), XPQ(-5) = xb~(x,Q²),
 XPQ(6) = xt(x,Q²), XPQ(-6) = xt~(x,Q²).

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 the user.

MINT(1) : specifies the general type of subprocess that has occurred, according to the ISUB code given in section 2.2.

MINT(2) : whenever MINT(1) (together with MINT(15) and MINT(16)) are not enough to specify the type of process uniquely, MINT(2) provides an ordering of the different possibilities, see MSTI(2).

MINT(3) : number of partons produced in the hard interactions, i.e. the number n of the 2 -> n matrix elements used; is sometimes 3 or 4 when a basic 2 -> 1 or 2 -> 2 process has been folded with two 1 -> 2 initial branchings (like $q q' \rightarrow q'' q''' H_0$).

MINT(4) : number of documentation lines in beginning of common block LUJETS that are given with $K(I,1) = 21$; 0 for $MSTP(125) = 0$.

MINT(5) : number of events generated to date in current run.

MINT(6) : current frame of event, cf. MSTP(124).

MINT(7), MINT(8) : line number for documentation of outgoing partons/particles from hard scattering for 2 -> 2 or 2 -> 1 -> 2 processes (else = 0).

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.

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(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.

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 semihard scatterings that occurred in
 current event in the multiple interaction scenario; is = 0 for a
 low-p_T event.

MINT(41), MINT(42) : type of incoming beam or target particle;
 1 for lepton and 2 for hadron.

MINT(43) : combination of incoming beam and target particles.
 = 1 : lepton on lepton.
 = 2 : lepton on hadron.
 = 3 : hadron on lepton.
 = 4 : hadron on hadron.

MINT(44) : total number of subprocesses switched on.

MINT(45) : number of subprocesses that are switched on, apart from
 elastic scattering and single, double and central diffractive.

MINT(46) : the heaviest new flavour switched on for QCD processes,
 specifically the flavour to be generated for ISUB = 81, 82 or
 83.

MINT(51) : internal flag that event failed cuts.
 = 0 : no problem.
 = 1 : event failed; new one to be generated.

MINT(52) : internal counter for number of lines used (in /LUJETS/)
 before multiple interactions are considered.

MINT(53) : internal counter for number of lines used (in /LUJETS/)
 before beam remnants are considered.

MINT(61) : internal switch for the mode of operation of resonance
 width calculations in PYWIDT for gamma*/Z0 or gamma*/Z0/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 width.
 = 0 : use widths into Z + Z* or W + W* calculated before.
 = 1 : evaluate widths into Z + Z* or W + W* for current Higgs mass.

MINT(71) : switch whether current process is singular for p_T -> 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 which 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 pileup events selected.

MINT(82) : sequence number of currently considered pileup event.

MINT(83) : number of lines in the event record already filled by
 previously considered pileup 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.

VINT(1) : ECM, CM energy.

VINT(2) : s (=ECM²) mass-square of complete system.

VINT(3) : mass of beam particle.

VINT(4) : mass of target particle.

VINT(5) : momentum of beam (and target) particle in CM frame.
 VINT(6) - VINT(10) : theta, phi and beta for rotation and boost
 from CM frame to user-specified frame.
 VINT(11) : tau_min.
 VINT(12) : y*_min.
 VINT(13) : cos(theta-hat)_min for cos(theta-hat) <= 0.
 VINT(14) : cos(theta-hat)_min for cos(theta-hat) >= 0.
 VINT(15) : x_T^2_min.
 VINT(16) : tau'_min.
 VINT(21) : tau.
 VINT(22) : y*.
 VINT(23) : cos(theta-hat).
 VINT(24) : phi* (azimuthal angle).
 VINT(25) : x_T^2.
 VINT(26) : tau'.
 VINT(31) : tau_max.
 VINT(32) : y*_max.
 VINT(33) : cos(theta-hat)_max for cos(theta-hat) <= 0.
 VINT(34) : cos(theta-hat)_max for cos(theta-hat) >= 0.
 VINT(35) : x_T^2_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 structure functions.
 VINT(43) : m-hat = sqrt(s-hat), mass of hard scattering subsystem.
 VINT(44) : s-hat of the hard subprocess (2 -> 2 or 2 -> 1).
 VINT(45) : t-hat of the hard subprocess (2 -> 2 or 2 -> 1 -> 2).
 VINT(46) : u-hat of the hard subprocess (2 -> 2 or 2 -> 1 -> 2).
 VINT(47) : p_T-hat of the hard subprocess (2 -> 2 or 2 -> 1 -> 2),
 i.e. transverse momentum evaluated in the rest frame of the
 scattering.
 VINT(48) : p_T-hat^2 of the hard subprocess; see VINT(47).
 VINT(49) : m'-hat, the mass of the complete three- or four-body final
 state in 2 -> 3 or 2 -> 4 processes.
 VINT(50) : s'-hat = m'-hat^2; see VINT(49).
 VINT(51) : Q of the hard subprocess, as used in structure functions
 and alpha_strong. The exact definition is process-dependent, see
 MSTP(32).
 VINT(52) : Q^2 of the hard subprocess; see VINT(51).
 VINT(61), VINT(62) : nominal m^2 values, i.e. without initial state
 radiation effects, for the two partons entering the hard
 interaction.
 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.
 VINT(65) : p-hat_init, i.e. common nominal absolute momentum of the two
 partons entering the hard interaction, in their rest frame.
 VINT(66) : p-hat_fin, i.e. common nominal absolute momentum of the two
 partons leaving the hard interaction, in their rest frame.
 VINT(71) : p_T_min of process, i.e. CKIN(3) or CKIN(5), depending on
 which is larger, and whether the process is singular in p_T -> 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 $2 \rightarrow 2$ 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(101) : total cross-section.

VINT(102) : elastic cross-section.

VINT(103) : single diffractive cross-section.

VINT(104) : double diffractive cross-section.

VINT(105) : central diffractive cross-section.

VINT(106) : total non-diffractive, inelastic cross-section.

VINT(108) : ratio of maximum differential cross-section observed to maximum differential cross-section assumed in generation; cf. MSTP(123).

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, $\gamma^*-Z'^0$ 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(121) : nuclear slope parameter B in t -hat distribution for (diffractive and) elastic scattering.

VINT(122) : curvature parameter C in t -hat distribution for (diffractive and) elastic scattering.

VINT(131) : total cross-section (in mb) for subprocesses allowed in the pileup events scenario according to the MSTP(132) value.

VINT(132) : $nbar = VINT(131) * PARP(131)$, cf. PARI(91).

VINT(133) : $\langle n \rangle = (\sum_i Prob_i * i) / (\sum_i Prob_i)$ as actually simulated, i.e. $1 \leq i \leq 100$ (or smaller), cf. PARI(92).

VINT(134) : is $\exp(-nbar) * \sum_i nbar^i / i!$ for MSTP(133) = 1 and $\exp(-nbar) * \sum_i nbar^i / (i-1)!$ 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) : $1 - (\text{sum of } x\text{-values})$ for all scatterings; used for rescaling each new x-value in the multiple interaction structure 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 .

VINT(147) : average hadronic matter overlap; used for MSTP(82) ≥ 2 .

VINT(148) : enhancement factor for current event in multiple interaction probability, defined as the actual overlap divided by the average one; used for MSTP(82) ≥ 2 .

VINT(149) : x_T^2 cutoff or turnover for multiple interactions. For MSTP(82) ≤ 1 it is $4 * p_{Tmin}^2 / s$, for MSTP(82) ≥ 2 it is $4 * p_{T0}^2 / s$.

VINT(150) : probability to keep given event in multiple interaction scenario, as given by the "Sudakov" form factor.

VINT(151), VINT(152) : sum of x values for all multiple interaction partons.

VINT(153) : current differential cross-section value obtained from

PYSIGH; used in multiple interactions only.

VINT(161) - VINT(200) : sum of Cabibbo-Kobayashi-Maskawa matrix element-squared 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.

COMMON/PYINT2/ISET(200),KFPR(200,2),COEF(200,20),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 cumulative cross-sections needed for multiple interaction generation for MSTP(82) >= 2. These variables must not be changed by the user.

ISET(ISUB) : gives the type of kinematical variable selection scheme used for subprocess ISUB.

- = 0 : elastic, diffractive and low-p_T processes.
- = 1 : 2 -> 1 processes (irrespective of subsequent decays).
- = 2 : 2 -> 2 processes (i.e. the bulk of processes).
- = 3 : 2 -> 3 processes (like $q q' \rightarrow q'' q''' H^0$).
- = 4 : 2 -> 4 processes (like $q q' \rightarrow q'' q''' W^+ W^-$).
- = -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 , W^+ , H^0 , etc.).

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 : tau selected according $1/\tau$.
- J = 2 : tau selected according to $1/\tau^2$.
- J = 3 : tau selected according to $1/(\tau*(\tau+\tau_r))$, where $\tau_r = m^2/s$ is tau value of resonance; only used for resonance production.
- J = 4 : tau selected according to Breit-Wigner of form $1/((\tau-\tau_r)^2+\text{gam}_r^2)$, where $\tau_r = m^2/s$ is tau value of resonance and $\text{gam}_r = m*\Gamma/s$ is its scaled mass times width; only used for resonance production.
- J = 5 : tau selected according to $1/(\tau*(\tau+\tau'_r))$, where $\tau'_r = m'^2/s$ is tau value of second resonance; only used for simultaneous production of two resonances.
- J = 6 : tau selected according to second Breit-Wigner of form $1/((\tau-\tau'_r)^2+\text{gam}'_r^2)$, where $\tau'_r = m'^2/s$ is tau value of second resonance and $\text{gam}'_r = m'*\Gamma'/s$ is its scaled mass times width; is used only for simultaneous production

of two resonances, like $\gamma^*/Z^0/Z'^0$.

J = 7 : y^* selected according to $y^* - y^*_{\min}$.

J = 8 : y^* selected according to $y^*_{\max} - y^*$.

J = 9 : y^* selected according to $1/\cosh(y^*)$.

J = 10 : $z = \cos(\theta\text{-hat})$ selected evenly between limits.

J = 11 : $z = \cos(\theta\text{-hat})$ selected according to $1/(a-z)$,
 where $a = 1 + 2*m_3^2*m_4^2/\hat{s}$, m_3 and m_4 being the
 masses of the two final state particles.

J = 12 : $z = \cos(\theta\text{-hat})$ selected according to $1/(a+z)$,
 with a as above.

J = 13 : $z = \cos(\theta\text{-hat})$ selected according to $1/(a-z)^2$,
 with a as above.

J = 14 : $z = \cos(\theta\text{-hat})$ selected according to $1/(a+z)^2$,
 with a as above.

J = 15 : τ' selected according to $1/\tau'$.

J = 16 : τ' selected according to $(1 - \tau'/\tau')^3/\tau'^2$.

J = 17 - 20 : reserved for future expansion.

ICOL : contains information on different colour flow topologies in
 hard 2 \rightarrow 2 processes.

COMMON/PYINT3/XSFX(2,-40:40),ISIG(1000,3),SIGH(1000)

Purpose: to store information on structure functions, subprocess cross-
 sections and different final state relative weights. These variables
 must not be changed by the user.

XSFX : current values of structure functions (multiplied by x) on
 beam and target side.

ISIG(ICHN,1) : incoming parton/particle on the beam side to the hard
 interaction for allowed channel no. 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 no. ICHN. See also comment above.

ISIG(ICHN,3) : colour flow type for allowed channel no. ICHN; see
 MSTI(2) list. See also comment above. 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
 no. ICHN, i.e. hard matrix element value times structure
 functions, for current kinematical setup (in addition, Jacobian
 factors are included in the figures, as used to speed up
 generation). See also comment for ISIG(ICHN,1).

COMMON/PYINT4/WIDP(21:40,0:40),WIDE(21:40,0:40),WIDS(21:40,3)

Purpose: to store partial and effective decay widths for the different
 resonances. These variables must not be changed by the user.

WIDP(KF,J) : gives partial decay widths of resonances into different
 channels (in GeV), given that all physically allowed final
 states are included.

KF : standard KF code for resonance considered.
J : enumerates the different decay channels possible for resonance KF, as stored in the JETSET LUDAT3 commonblock, with the first channel in J = 1, etc.

WIDE(KF,J) : gives effective decay widths of resonances into different channels (in GeV), given the decay modes actually left open in the current run. The on/off status of decay modes is set by the MDME switches in JETSET; see section 2.9.

KF : standard KF code for resonance considered.
J : enumerates the different decay channels possible for resonance KF, as stored in the JETSET LUDAT3 commonblock, with the first channel in J = 1, etc.

WIDS(KF,J) : gives a dimensionless suppression factor, which is defined as the ratio of the total width of channels switched on to the total width of all possible channels (replace width by width-squared for a pair of resonances). The on/off status of channels is set by the MDME switches in JETSET; see section 2.9. The information in WIDS is used e.g. in cross-section calculations.

KF : standard KF code for resonance considered.
J = 1 : suppression when a pair of resonances of type KF are produced together. When an antiparticle exists, the particle-antiparticle pair (like W+ W-) is the relevant combination, else the particle-particle one (like Z0 Z0).
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.

COMMON/PYINT5/NGEN(0:200,3),XSEC(0:200,3)

Purpose: to store information necessary for cross-section calculation and differential cross-section maximum violation. These variables must not be changed by the user.

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 normally can 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_T, i.e. ISUB = 95).

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.

COMMON/PYINT6/PROC(0:200)

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.

2.9. The JETSET routines

Since the fragmentation and decay of an original parton configuration is handled by the JETSET routines, using the LUJETS common block described above, all of the non-e+e- routines and parameters are at the disposal of the PYTHIA user. This section is intended as a brief reminder of some of the most useful ones, in particular for hadron physics applications. Further details may be obtained by consulting the JETSET manual.

The subroutine LULIST can be used to list an event; for output that fits onto a normal terminal screen CALL LULIST(1) is recommended, whereas CALL LULIST(2) gives a slightly more comprehensive listing. The event record nominally contains not only the stable final particles, but also the original partons and intermediate resonances. With CALL LUEDIT(1) all partons/particles that have fragmented/decayed are removed from the event record, and N is updated accordingly. A CALL LUEDIT(2) will remove neutrinos as well, and CALL LUEDIT(3) will only leave charged, stable particles. Some of the information indirectly stored in LUJETS can be accessed more easily by using the PLU and KLU functions, thus e.g. $y = \text{PLU}(I,17)$ gives the true rapidity and $\eta = \text{PLU}(I,19)$ the pseudorapidity of particle I.

With MSTP(111) = 0, fragmentation and decays are switched off in the PYEVNT call. If one desires to fragment the event at some later opportunity, this can be achieved by a CALL LUEXEC. The action of the program during this call, which is done automatically from PYEVNT for MSTP(111) = 1, can be controlled by the switches and parameters in the LUDAT1 common block:

COMMON/LUDAT1/MSTU(200),PARU(200),MSTJ(200),PARJ(200)

Thus, MSTJ(21) = 0 will switch off particle decays but keep jet fragmentation on, MSTJ(1) can be set to give independent fragmentation rather than the standard string fragmentation, etc. Contrary to most PYTHIA variables, the user can freely change the values during the course of the run, since no initialization is involved.

The same commonblock also contains a number of conversion factors and coupling constants. Thus MSTU(101) can be set to used fix or running α_{em} , PARU(101) to set the α_{em} value, PARU(102) to set $\sin^2(\theta_W)$, PARU(121) - PARU(129) to modify the couplings of a Z', PARU(131) - PARU(135) those of a W', and PARU(141) - PARU(143) those of a H⁺⁻. These couplings can not be changed in midrun, since they are used in the initialization to estimate cross-section maxima.

The common block LUDAT2 contains several pieces of information on particle and flavour properties:

```
COMMON/LUDAT2/KCHG(500,3),PMAS(500,4),PARF(2000),VCKM(4,4)
```

The particle masses may be the most interesting variables here. Since these masses are directly taken over by the PYTHIA routines, any mass changes should be made before the PYINIT call. Of particular interest are the following:

PMAS(6,1) : (D=120. GeV/c²) mass of top quark.

PMAS(7,1) : (D=120. GeV/c²) mass of l quark.

PMAS(8,1) : (D=200. GeV/c²) mass of h quark.

PMAS(17,1) : (D=100. GeV/c²) mass of chi lepton.

PMAS(23,1) : (D=91.2 GeV/c²) Z⁰ nominal mass.

PMAS(24,1) : (D=80.0 GeV/c²) W^{+/-} nominal mass.

PMAS(25,1) : (D=50. GeV/c²) H⁰ nominal mass.

PMAS(32,1) : (D=500. GeV/c²) Z'⁰ nominal mass.

PMAS(34,1) : (D=500. GeV/c²) W'^{+/-} nominal mass.

PMAS(37,1) : (D=300. GeV/c²) H^{+/-} nominal mass.

PMAS(40,1) : (D=5000. GeV/c²) R nominal mass.

For Z⁰, W^{+/-}, H⁰, Z'⁰, W'^{+/-}, H^{+/-}, and R, the resonance width is calculated at PYINIT initialization, taking into account the channels assumed possible for resonance decays (for H⁰ this will depend very much on the mass assumed), so the nominal widths stored in PMAS(KF,2) are actually overwritten at this point.

Note that the first index in the KCHG and PMAS arrays is not the standard KF code described in section 2.7, which may be both positive and negative, and extends to very large values. Instead a compressed code KC is used, which only runs between 1 and 500, i.e. which contains no distinction between particles and antiparticles. The KF and KC codes agree between 1 and 40, i.e. for quarks, leptons and gauge bosons, while meson and baryon codes have been compressed. Thus, several hadrons containing heavy quarks (some c and b baryons, most t, l and h hadrons) have been lumped together, with masses and charges instead derived from the quark content (and, for the mass, the spin). The user is not supposed to learn the KC code, but instead use the LUCOMP function to translate from KF to KC code, i.e. KC = LUCOMP(KF). PMAS(LUCOMP(663),1) = 110. changes the toponium mass to 110 GeV, to give one example. LUCOMP will return KC = 0 for any particle code unknown in the program.

The common block LUDAT3 contains all particle decay data:

```
COMMON/LUDAT3/MDCY(500,3),MDME(2000,2),BRAT(2000),KFDP(2000,5)
```

It can be used to switch off the decay of some particles selectively, and even to switch on and off separate decay modes for a given particle. These switches are used extensively within PYTHIA, both to select allowed final state flavours and to calculate the reduction in cross-section due to these restrictions. MDCY(KC,1) is an on/off switch (= 0 off, = 1 on) for the decay of compressed particle code KC. The decay of Z⁰, both in PYTHIA and in JETSET, is thus switched off by putting MDCY(23,1) = 0. In order to keep the event record from becoming uncomfortably large, switching off pi⁰ decays by putting MDCY(LUCOMP(111),1) = 0 is probably the single most effective action.

All the decay channels defined in the program are stored sequentially

in MDME, BRAT and KFDP. The entry point into this table and the number of channels stored for a given particle is given by the last two entries of the MDCY array. In order to obtain a list of particle data, including all decay channels, do CALL LULIST(12) (and take out a hardcopy for future reference). For resonances only, the same information is available with CALL PYSTAT(2). For a given decay channel IDC, the switch in MDME(IDC,1) is of special interest. It can be used to switch on or off that particular channel, or leave it selectively open. Effective branching ratios are automatically recalculated among the decay channels left open. If a particle is allowed to decay by the MDCY(KC,1) value, at least one channel must be left open by the user. Since the MDME(IDC,1) code is of special interest in PYTHIA for selecting flavours in single gauge boson or gauge boson pair decays, the JETSET manual description on this point is reproduced in extenso:

- = -1 : this is a non-standard model decay mode, which by default is assumed not to exist. It is therefore not simulated; neither does it contribute to the total width of the particle. Normally, this option is used for decays involving fourth generation or H[±] particles.
- = 0 : channel is switched off. It is not simulated, but still contributes to the total width of the particle.
- = 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 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 its own antiparticle.
- = 4 : in the production of a pair of equal or charge conjugate resonances in PYTHIA, say H⁰ → 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.
Within JETSET, 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.
Within JETSET, 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. Thus 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, users are warned unwittingly to set fourth generation channels 0 (rather than -1), since by now the support for a fourth generation is small.

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. H⁰ → W⁺ W⁻ → e⁺ ν_e d⁺ u⁺ or d⁺ u⁺ e⁻ ν_ebar, but the former only one

of them. In calculations of the joint branching ratio, this makes a factor 2 difference.

One should note that, whereas the MDME code can be used to set allowed decay channels both for resonances (Z^0 , $W^{+/-}$, H^0 , Z'^0 , $W'^{+/-}$, $H'^{+/-}$, and R) and for ordinary particles (π^0 , K , D , B , top mesons, etc.), it is only for the resonances that cross-sections are modified to take into account the allowed channels. The reason is obvious: resonances are only produced as part of the hard interaction, or as decay products of other resonances, according to fairly strict rules, whereas ordinary particles can be produced anywhere in the event (charm and bottom quark pairs may appear inside initial and final state showers, e.g.) in a rather haphazard fashion. It is therefore not possible to generate unbiased events in a consistent manner for ordinary particles. (One might imagine doing things properly for the t quark, since it is not likely to be produced in showers, but this has not been done so far.)

For QCD processes, like the pair production of a new heavy quark (ISUB = 81 and 82), imagined 'decay modes' are stored for the gluon among the ordinary ones. Branching ratios here have no meaning, but the MDME(IDC,1) switch can be used to enable or disable the production of different new quarks. In the current version of the massive matrix element treatment, only one heavy flavour can be generated in a given one; if the user has left several flavours switched on, only the heaviest is actually used. Similarly, the 'decay modes' stored for quarks indicate which branchings of the type $q \rightarrow q' W$ are allowed, as used e.g. in $W^+ W^- \rightarrow H^0$ production.

It is also possible to include completely new decay modes, or even a completely new particle. This is described in the JETSET manual.

The event analysis routines can be called on to evaluate the event stored in the LUJETS common block. Particularly useful is the cluster finding routine LUCCELL, which assumes a (modifiable) grid of cells in pseudorapidity and azimuthal angle and carries out a cluster finding algorithm a la the UA1 one. With a CALL LUCCELL(NJET) the number of jets found is given by NJET and the position and ET of those jets is stored after the event proper, in lines N+1 through N+NJET.

The routine LUGIVE can be used to feed in commonblock values both to JETSET and to PYTHIA. Since LUGIVE checks on array boundaries and also writes on output the variables which have been changes, this may at times be a better way to set up the values to be used by the program. A call might look like

```
CALL LUGIVE('MSEL=0;MSUB(142)=1;PMAS(34,1)=300.;CKIN(1)=200.')
```

i.e. several commands can be put in one if separated by semicolon (but it is of course also possible to do one command at a time and call LUGIVE as often as desired). Note that for people who want data card driven programs, this facilitates the construction of a driver routine for PYTHIA.

2.10. On cross-sections

This section includes two related topics. The first is how events are generated according to the proper matrix elements, the second how the final cross-sections (available e.g. with PYSTAT) are calculated.

In PYRAND, the variables used in the generation of hard scattering $2 \rightarrow n$ processes are τ (= \hat{s}/s , i.e. mass-squared of scattering subsystem scaled by total CM energy-squared), y^* (rapidity of scattering subsystem in the CM frame of the event), z (= $\cos(\hat{\theta})$, i.e. cosine of scattering angle in CM frame of the hard interaction; this applies to $2 \rightarrow 2$ and $2 \rightarrow 4$ processes only), and τ' (= \hat{s}'/s , i.e. mass-squared of complete three- or four-body final state scaled by total CM energy-squared; this applies to $2 \rightarrow 3$ and $2 \rightarrow 4$ processes only). These variables are generated according to the distributions $h_1(\tau)/\tau$, $h_2(y^*)$, $h_3(z)$, and $h_4(\tau')/\tau'$, where

$$h_1(\tau) = c_0 + I_0/I_1 * c_1 * 1/\tau + I_0/I_2 * c_2 * 1/(\tau + \tau_R) + \\ + I_0/I_3 * c_3 * \tau / ((s * \tau - m_R^2)^2 + (m_R * \Gamma_R)^2) + \\ + I_0/I_4 * c_4 * 1/(\tau + \tau_{R'}) + I_0/I_5 * c_5 * \\ * \tau / ((s * \tau - m_{R'}^2)^2 + (m_{R'} * \Gamma_{R'})^2)$$

$$h_2(y^*) = I_0/I_1 * c_1 * (y^* - y^*_{\min}) + I_0/I_1 * c_2 * (y^*_{\max} - y^*) + \\ + I_0/I_3 * c_3 * 1/\cosh(y^*)$$

$$h_3(z) = c_0 + I_0/I_1 * c_1 * 1/(a - z) + I_0/I_2 * c_1 * 1/(a + z) + \\ + I_0/I_3 * c_3 * 1/(a - z)^2 + I_0/I_4 * c_4 * 1/(a + z)^2$$

$$h_4(\tau') = c_0 + I_0/I_1 * c_1 * (1 - \tau/\tau')^3 / \tau',$$

where subscripts R and R' in h_1 denote values (of τ , mass and width) pertaining to a resonance R or R', and a in h_3 denotes $1 + 2 * (m_3 * m_4 / \hat{s})^2$ (= 1 for massless products).

In each of the above cases, I_0 denotes the integral over the quantity multiplying coefficient c_n ; thus, e.g.,

$$h_1: I_0 = \int d\tau / \tau = \ln(\tau_{\max} / \tau_{\min})$$

$$h_3: I_0 = \int dz = z_{-\max} - z_{-\min} + z_{+\max} - z_{+\min}$$

$$h_4: I_0 = \int d\tau' / \tau' = \ln(\tau'_{\max} / \tau'_{\min})$$

(for h_2 , where no c_0 occurs, I_0 is defined to be $\int dy^* = y^*_{\max} - y^*_{\min}$), and the coefficients c_n are normalized, such that $\sum(n) c_n = 1$. (The symbols \int and \sum are used to denote integral and sum over, respectively.)

For other types of processes, e.g. elastic and diffractive scattering, other variables are used in the generation, but the structure of the generation procedure is similar.

After this primary generation, the variables are then weighted with the relevant matrix elements, etc., according to standard Monte Carlo techniques.

The process-dependent weights are coded in the routine PYSIGH, as follows. The cross-sections for 2 → 1 and 2 → 2 processes can be written, respectively,

$$\sigma(2 \rightarrow 1) = \text{Int } dx_1 \text{ Int } dx_2 \text{ Sum}(ijk) f_i f_j * \\ * d\sigma\text{-hat}(ijk)$$

and

$$\sigma(2 \rightarrow 2) = \text{Int } dx_1 \text{ Int } dx_2 \text{ Int } dt\text{-hat} \text{ Sum}(ijk) f_i f_j * \\ * d\sigma\text{-hat}(ijk)/dt\text{-hat},$$

where Int and Sum denote respectively integral and sum over, f_i the structure function for parton i , and $d\sigma\text{-hat}$ the differential cross-section for the hard scattering. Converting to the variables τ , y^* , and $z = \cos(\theta\text{-hat})$, the cross-sections can then be rewritten as

$$\sigma(2 \rightarrow 1) = \pi/s * \text{Int } d\tau h_1(\tau)/\tau \text{ Int } dy^* h_2(y^*) * \\ * 1/(\tau*h_1(\tau)) * 1/h_2(y^*) * \text{Sum}(ijk) F_i F_j * \\ * s\text{-hat}/\pi*d\sigma\text{-hat}(ijk) = \\ = \pi/s * I(\tau) * I(y^*) * 1/(\tau*h_1) * 1/h_2 * \\ * \text{Sum}(ijk) F_i F_j * s\text{-hat}/\pi*d\sigma\text{-hat}(ijk)$$

and

$$\sigma(2 \rightarrow 2) = \pi/s * \text{Int } d\tau h_1(\tau)/\tau \text{ Int } dy^* h_2(y^*) * \\ * \text{Int } dz h_3(z) * 1/(\tau*h_1(\tau)) * 1/h_2(y^*) * \\ * 1/2*\beta_{34}/h_3(z) * \text{Sum}(ijk) F_i F_j * \\ * s\text{-hat}^2/\pi*d\sigma\text{-hat}(ijk)/dt\text{-hat} = \\ = \pi/s * I(\tau) * I(y^*) * I(z) * 1/(\tau*h_1) * \\ * 1/h_2 * 1/2*\beta_{34}/h_3 * \text{Sum}(ijk) F_i F_j * \\ * s\text{-hat}^2/\pi*d\sigma\text{-hat}(ijk)/dt\text{-hat},$$

where F_i denotes $x*f_i$, $\beta_{34} = \lambda(1, r_3, r_4)$ with λ the usual λ -function and $r_i = m_i^2/s\text{-hat}$, and

$$I(\tau) = \text{Int } d\tau h_1(\tau)/\tau$$

$$I(y^*) = \text{Int } dy^* h_2(y^*)$$

$$I(z) = \text{Int } dz h_3(z),$$

and the h 's are distributions used in the actual generation of the kinematical variables τ , y^* and z (see above), such that

$$\text{Int } d\tau h_1(\tau)/\tau = \ln(\tau_{\text{max}}/\tau_{\text{min}})$$

$$\text{Int } dy^* h_2(y^*) = y^*_{\text{max}} - y^*_{\text{min}}$$

$$\text{Int } dz h_3(z) = z_{\text{-max}} - z_{\text{-min}} + z_{\text{+max}} - z_{\text{+min}}.$$

For each subprocess in PYSIGH, what is coded under the relevant

ISUB is thus the dimensionless quantity

$$s\text{-hat}/\pi * d\text{sigma-hat}(ijk) \quad (2 \rightarrow 1 \text{ processes})$$

$$s\text{-hat}^2/\pi * d\text{sigma-hat}(ijk)/dt\text{-hat} \quad (2 \rightarrow 2 \text{ processes})$$

which is then multiplied by the structure functions (or, rather, $x*f$) and a common factor, containing π/s , the conversion factor from GeV^{-2} to mb, $I(\tau)$, $I(y^*)$ (and, for $2 \rightarrow 2$ processes, $I(z)$) and $1/(\tau*h1)$, and $1/h2$ (and, for $2 \rightarrow 2$ processes, also $1/2*\beta_{34}/h3$). It is the latter product which is returned in a PYSIGH call; we will denote this $\text{SIGH}(\tau, y^*, z)$ in the following.

For $2 \rightarrow 1$ processes with $d\text{sigma-hat}$ given in the zero-width approximation, the delta function in τ has been replaced by the (modified) Breit-Wigner

$$1/\pi * s * H_R / ((s * \tau - m_R^2)^2 + H_R^2),$$

where

$$H_R = s * \tau / m_R * \Gamma_R,$$

with m_R and Γ_R the mass and the width of the resonance R , respectively. The Breit-Wigner thus contains an s -dependent width which improves the description of the line shape as compared to the case with widths calculated at nominal resonance mass. In the $2 \rightarrow 2$ processes 71 - 77, where the Higgs occurs in s -, t - and u -channel exchanges, the width is calculated at nominal Higgs mass, however.

For effective $2 \rightarrow 3, 4$ processes (e.g. $Z0 Z0 \rightarrow H0$), the common factor further includes the integration over the extra variable τ' , i.e. a factor $I(\tau') * f(\tau')/h4(\tau')$, where

$$f(\tau') = (1 + \tau/\tau') \ln(\tau'/\tau) - 2(1 - \tau/\tau')$$

and

$$I(\tau') = \int dt\tau' h4(\tau')/\tau' = \ln(\tau'_{\text{max}}/\tau'_{\text{min}}).$$

A user wanting to include new processes (experts only!), will have to insert then the function $s\text{-hat}/\pi * d\text{sigma-hat}(ijk)$ or $s\text{-hat}^2/\pi * d\text{sigma-hat}(ijk)/dt\text{-hat}$ under the appropriate ISUB heading (in addition to changes in e.g. SUBROUTINE PYSCAT).

Let us now describe how these differential cross-sections are used to derive the total cross-sections (within the user-defined cuts) for the allowed processes. We will first consider high- p_T event generation only, and later cover low- p_T physics.

At initialization of a $2 \rightarrow 2$ process, the coefficients c_n appearing in the expressions $h1(\tau)$, $h2(y^*)$ and $h3(z)$ are first optimized so as to fit the actual variation of the cross-section as well as can be achieved, i.e. to see to it that $\text{SIGH}(\tau, y^*, z)$ is a slowly varying

function of its arguments. Thereafter the maximum of $SIGH(\tau, y^*, z)$ is searched for. This gives a preliminary cross-section estimate, which is printed among the initialization information, and which represents an overestimate of the actual cross-section. It is this maximum which is used below to find the relative weight for selecting a processes. For $2 \rightarrow 1$, $2 \rightarrow 3$ and $2 \rightarrow 4$ processes, the same philosophy is used, with trivial changes in the number of variables that have to be considered.

In particular, note that SIGH is selected so as to allow a direct comparison between the different classes of events. More precisely, the SIGH value returned by PYSIGH is the total integrated cross-section a process would have inside the simulated phase space volume, if the actual variation of the cross-section were perfectly described by $h_1(\tau)/\tau * h_2(y^*) * h_3(z)$ (or equivalent), with overall normalization given by the differential cross-section value in the point selected.

In order to generate an event, first a process is chosen, thereafter a set of kinematical variables for this process is selected, as described above. The probability for accepting this set is given by the actual quantity $SIGH(\tau, y^*, z)$, divided by the maximum of this quantity, as found earlier. In case of rejection, both process and kinematical variables are to be chosen anew.

For each individual subprocess, there are counters giving the number of points tried and the sum of the SIGH values in these points. At the end of each PYEVT call, the ratio of these two numbers is calculated, thus providing a Monte Carlo estimate of the actual cross-section for each process. If only very few events have been generated, the statistical fluctuations are large. Therefore it is recommended to call PYSTAT only after all events have been generated, although a call in principle could be made at any time. In PYSTAT the cross-sections will be listed together with the number of phase space points tried and the number of events actually generated.

The cross-sections for each subprocess are calculated taking into account which initial and final states are allowed, by the KFIN array in the common block PYSUBS, and the MDME(IDC,1) values in the common block LUDAT3, respectively. By default, everything up to and including top is allowed, while it is assumed that no fourth generation exists. If some initial states are switched off, the cross-sections will be reduced accordingly. For several processes, like $q + g \rightarrow q + g$, the flavours in the final state are the ones of the initial state, and for these MDME has no effect. In processes like $g + g \rightarrow q + qb$ or $q + qb \rightarrow Z^0/\gamma^* \rightarrow \mu^+ + \mu^-$, on the other hand, the quark and lepton flavours allowed in the final state can be selected using MDME, and cross-sections are only based on those channels that are actually allowed. We emphasize that, for resonances like Z^0 or $W^{+/-}$, the total widths are assumed given as the sum over three (or, optionally, four) full generations of quarks and leptons. By only allowing a few channels, we do thus in no way alter the basic properties of the resonances, but only restrict the generation of events to those particular channels we happen to be interested in, with the cross-sections reduced accordingly. Particles like H^0 , Z' and W' can either

decay directly into a $q\bar{q}$ or $l\bar{l}$ pair, or into a (gauge) boson pair, which then can decay further. Therefore the production cross-section given by the program is reduced, not only by switching off some decay channels of the primary resonance, but also by switching off subsequent Z, W or H^+ decay modes (and the branching ratios of the primary resonance into the boson pair is reduced accordingly). Even if the decay of a resonance is switched off completely, by using the MDCY array in the LUDAT3 commonblock, cross-sections are still based on the decay channels allowed by the MDME values in the same commonblock. Note that what is said here about resonances does not hold for ordinary particles or even the top quark, where a suppression of some decay channels are never taken into account in PYTHIA cross-sections, but have to be compensated for by the user himself.

The cross-section values printed by PYSTAT are stored in XSEC(ISUB,3) in common block PYINT5. In particular, XSEC(0,3) = PARI(1) gives the total cross-section for all subprocesses studied in current run. Events are generated with unit weight. At the end of a run, histogram contents etc. can be converted from number of events to differential cross-sections by multiplying by XSEC(0,3) and dividing by the total number of events, the latter stored in MSTI(5). This ratio is given in PARI(2). For histograms one should also divide by the bin width.

What has been said so far applies when only hard interactions are studied. The total hadronic cross-section, as well as the elastic, single diffractive and double diffractive cross-sections are taken from parametrizations available in the literature. For the total and elastic cross-sections this means the fits of Block-Cahn [Blo85], for diffractive events the ansatz of Goulianos [Gou83] is used. The cross-section for non-diffractive low- p_T events is then obtained by subtraction. When the multiple interaction formalism is used to generate these latter events, some of them will actually contain a semihard interaction, and be listed in PYSTAT as such, with only the events without any semihard interactions being classified as low- p_T (ISUB = 95). This matter of classification does not remove the constraint that the total cross-section is fixed, however. Neither can the KFIN switches be set to suppress some of the flavours of the initial hadrons (the inclusion of heavy flavours or not can still be set using MSTP(54), however).

2.11. Examples

The program is built as a slave system, i.e. the user supplies the main program, which calls on the PYTHIA and JETSET routines to perform specific tasks and then resumes control.

A typical program for analysis of collider events at 540 GeV CM energy with a minimum p_T of 10 GeV/c at the hard scattering (because of initial state radiation, fragmentation effects, etc., the actual p_T -cutoff will be smeared around this value) might look like

```
COMMON/LUJETS/N,K(4000,5),P(4000,5),V(4000,5)
COMMON/PYSUBS/MSEL,MSUB(200),KFIN(2,-40:40),CKIN(200)
COMMON/PYPARS/MSTP(200),PARP(200),MSTI(200),PARI(200)
...           ! set all common block variables that
```

```

...                ! did not have desired default values
CKIN(3)=10.        ! lower p_T cutoff
CALL PYINIT('CMS','p','pbar',540.)      ! initialize
...                ! initialize analysis statistics
DO 100 IEVENT=1,1000          ! loop over events
CALL PYEVNT          ! generate event
IF(IEVENT.EQ.1) CALL LULIST(1)      ! list first event
...                ! insert desired analysis chain for
...                ! each event
100 CONTINUE
CALL PYSTAT(1)        ! print cross-sections
...                ! user output
END

```

References

- Alt89 G. Altarelli, B. Mele, M. Ruiz-Altaba, Z. Physik C45 (1989) 109
- Ben84 H.-U. Bengtsson, Computer Phys. Comm. 31 (1984) 323
- Ben84a H.-U. Bengtsson, G. Ingelman, LU TP 84-3, Ref.TH.3820-CERN
- Ben85 H.-U. Bengtsson, G. Ingelman, Computer Phys. Comm. 34 (1985) 251
- Ben85a H.-U. Bengtsson, W.-S. Hou, A. Soni, D.H. Stork,
Phys. Rev. Lett. 55 (1985) 2762
- Ben87 H.-U. Bengtsson, T. Sjostrand, Computer Phys. Comm. 46 (1987) 43
- Blo85 M. M. Block, R. N. Cahn, Rev. Mod. Phys. 57 (1985) 563
M. M. Block, R. N. Cahn, in Physics Simulations at High Energy,
eds. V. Barger, T. Gottschalk, F. Halzen (World Scientific,
Singapore, 1987), p. 89
- BLS87 M. C. Bento, C. H. Llewellyn Smith, Nucl. Phys. B289 (1987) 36
- Com77 B. L. Combridge, J. Kripfganz, J. Ranft, Phys. Lett. 70B (1977)
234
- Con71 V. Constantini, B. de Tollis, G. Pistoni, Nuovo Cim. 2A (1971)
733
- Cut78 R. Cutler, D. Sivers, Phys. Rev. D17 (1978) 196
- DFL88 M. Diemoz, F. Ferroni, E. Longo, G. Martinelli, Z. Physik C39
(1988) 21
- Dic88 D. A. Dicus, S. S. D. Willenbrock, Phys. Rev. D37 (1988) 1801
- Dun86 M. J. Duncan, G. L. Kane, W. W. Repko, Nucl. Phys. B272 (1986)
517
- EHL84 E. Eichten, I. Hinchliffe, K. Lane, C. Quigg, Rev. Mod. Phys.

56 (1984) 579; 58 (1985) 1065

- Ell86 R. K. Ellis, J. C. Sexton, Nucl. Phys. B269 (1986) 445
- Ell88 R. K. Ellis, I. Hinchliffe, M. Soldate, J. J. van der Bij,
Nucl. Phys. B297 (1988) 221
- Fad89 V. Fadin, V. Khoze, JETP Lett. 46 (1987) 417, Yad. Fiz 48
(1988) 487, in Proceedings of the 24th Winter School of the
LNPI (Leningrad, 1989), vol. I, p. 3
V. Fadin, V. Khoze, T. Sjostrand, CERN-TH.5394/89 (1989), to
appear in the proceedings of the 24th Rencontres de Moriond
- Gou83 K. Goulianos, Phys. Rep. 101 (1983) 169
- Gun86 J.F. Gunion, Z. Kunszt, Phys. Rev. D33 (1986) 665,
errata as private communication from the authors
- Hal78 F. Halzen, D. M. Scott, Phys. Rev. D18 (1978) 3378
- PDG88 Particle Data Group, G. P. Yost et al.,
Phys. Lett. 204B (1988) 1
- Sjo87 T. Sjostrand, M. van Zijl, Phys. Rev. D36 (1987) 2019
- Sjo90 JETSET 7.3 program and manual (unpublished, obtainable on
request)
- Tun87 W.-K. Tung, in Physics Simulations at High Energy,
eds. V. Barger, T. Gottschalk, F. Halzen (World Scientific,
Singapore, 1987), p. 601
W.-K. Tung, private communication
- Zer90 P. Zerwas, private communication
