


```

*****
* 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 7.2, 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	[this]	new particle and subprocess codes, new commonblock structure, new kinematics selection, some lepton-lepton and lepton-hadron interactions, new subprocesses

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. The current version is therefore the first major revision since 4.8.

1.3. Major Changes from PYTHIA 4.8

The present version, PYTHIA 5.3, of the Lund Monte Carlo for Hadronic Processes represents 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, 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 overlaid events, where one single bunch crossing gives rise to several hadron-hadron interactions.
-

1.4. 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 7.2. Comments on compiler optimization level, random number generators and machine precision problems are the same as given in the JETSET 7.2 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.

1.5. 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 PYTHIA 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, PYTHIA and PYSTAT.
 - The ISUB classification code for subprocesses included in PYTHIA is tabulated in 2.2.
 - The following two subsections, 2.3 and 2.4, 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 2.3 is collected the switches for the type of process to generate and kinematical constraints. Subsection 2.4 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 2.5 and 2.6, in the former the general type of process and some kinematical variables, in the latter the detailed list of all particles generated.
 - In 2.7 the further subroutines, functions and commonblocks in PYTHIA are listed, with brief information about their purpose.
 - The fragmentation routines of JETSET 7.2 are amply documented elsewhere [Sjo89], but in 2.8 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 2.9.
 - Finally, subsection 2.10 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 PYTHIA 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.

- = 'FIXT' : fixed target experiment, with beam particle momentum pointing in +z direction.
- = 'CMS' : colliding beam experiment in CM frame, with beam momentum in +z direction and target momentum 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 uncharged antiparticle may be denoted either by 'bar' or '~' at the end of

the name. It is also possible to insert an underscore ('_') directly after 'nu' in neutrino names.

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

WIN : related to energy of system, exact meaning depends on FRAME.
FRAME='FIXT' : momentum of beam particle (in GeV/c).
FRAME='CMS' : total energy of system (in GeV).
FRAME='USER' : dummy.

SUBROUTINE PYTHIA

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.)

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 decay channels for the resonances, with partial decay widths, branching ratios and effective branching ratios (in the event some channels have been excluded by the user) for each.
- = 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, 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 2.7) 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`, the two incoming flavours at the hard interaction, `E_CM`, `m-hat`, `s-hat`, `t-hat`, `u-hat`, `p_T-hat`, `m'-hat`, `Q^2`, `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, $2 \rightarrow 3$ 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 $2 \rightarrow 1$ process (e.g. $q \bar{q} \rightarrow \gamma^* \rightarrow e^+ e^-$), but a $2 \rightarrow 2$ 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, $2 \rightarrow 1$ and $2 \rightarrow 2$ graphs may sometimes be folded with two $1 \rightarrow 2$ splittings to form effective $2 \rightarrow 3$ or $2 \rightarrow 4$ processes ($W^+ W^- \rightarrow H^0$ is folded with $q \rightarrow q' W$ to give $q q' \rightarrow q'' q''' H^0$).

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 2.3 and 2.5. 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 is included as well (with possibilities to switch off either, if so desired). 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.

a) $2 \rightarrow 1$, tree

```

+   1   f_i f_i~ -> Z^0
+   2   f_i f_j~ -> W^+
+   3   f_i f_i~ -> H^0
+   4   gamma W^+ -> W^+
+   5   Z^0 Z^0 -> H^0
+   6   Z^0 W^+ -> W^+
+   7   W^+ W^- -> Z^0
+   8   W^+ W^- -> H^0

```

b) $2 \rightarrow 2$, tree

```

+  11   f_i f_j(~) -> f_i f_j(~)
+  12   f_i f_i~ -> f_k f_k~
+  13   f_i f_i~ -> g g
+  14   f_i f_i~ -> g gamma
+  15   f_i f_i~ -> g Z^0
+  16   f_i f_j~ -> g W^+
+  17   f_i f_i~ -> g H^0
+  18   f_i f_i~ -> gamma gamma
+  19   f_i f_i~ -> gamma Z^0
+  20   f_i f_j~ -> gamma W^+
+  21   f_i f_i~ -> gamma H^0
+  22   f_i f_i~ -> Z^0 Z^0
+  23   f_i f_j~ -> Z^0 W^+
+  24   f_i f_i~ -> Z^0 H^0
+  25   f_i f_i~ -> W^+ W^-
+  26   f_i f_j~ -> W^+ H^0
+  27   f_i f_i~ -> H^0 H^0
+  28   f_i g -> f_i g
+  29   f_i g -> f_i gamma
+  30   f_i g -> f_i Z^0
+  31   f_i g -> f_k W^+
+  32   f_i g -> f_i H^0
+  33   f_i gamma -> f_i g
+  34   f_i gamma -> f_i gamma
+  35   f_i gamma -> f_i Z^0
+  36   f_i gamma -> f_k W^+
+  37   f_i gamma -> f_i H^0
+  38   f_i Z^0 -> f_i g
+  39   f_i Z^0 -> f_i gamma
+  40   f_i Z^0 -> f_i Z^0
+  41   f_i Z^0 -> f_k W^+
+  42   f_i Z^0 -> f_i H^0
+  43   f_i W^+ -> f_k g
+  44   f_i W^+ -> f_k gamma
+  45   f_i W^+ -> 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 $g g \rightarrow f_k f_{k\sim}$
 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 $g g \rightarrow g g$
 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 $Z^0 W^+ \rightarrow Z^0 W^+$
 74 $Z^0 H^0 \rightarrow Z^0 H^0$
 75 $W^+ W^- \rightarrow \gamma \gamma$
 + 76 $W^+ W^- \rightarrow Z^0 Z^0$
 + 77 $W^+ W^{+(-)} \rightarrow W^+ W^{+(-)}$
 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 $f_i f_{i\sim} \rightarrow Q_i Q_{i\sim}$
 + 82 $g g \rightarrow Q_i Q_{i\sim}$

d) "minimum bias"

+ 91 elastic scattering
 + 92 single diffraction
 + 93 double diffraction
 94 central diffraction
 + 95 low- p_T production

e) $2 \rightarrow 1$, loop

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

f) $2 \rightarrow 2$, box

+ 111 $f_i f_{i\sim} \rightarrow g H^0$
 + 112 $f_i g \rightarrow f_i H^0$
 + 113 $g g \rightarrow g H^0$
 + 114 $g g \rightarrow \gamma \gamma$

```

115    g g -> gamma Z0
116    g g -> Z0 Z0
117    g g -> W+ W-

```

g) 2 -> 3, tree

```

121    g g -> f_k f_k~ H0

```

h) non-standard model, 2 -> 1

```

+ 141    f_i f_i~ -> Z'0
+ 142    f_i f_j~ -> H+
+ 143    f_i f_j~ -> R

```

i) non-standard model, 2 -> 2

```

+ 161    f_i g -> f_k H+

```

For many of the subprocesses, further notes and qualifications may be of interest. These are given below.

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.

The subprocesses 15, 16, 30 and 31, with a $Z0/W+$ recoiling against a q/g jet, are also effectively generated by initial state radiation corrections to the subprocesses 1 and 2, which cover the production of a single $Z0/W+$. In order to avoid double-counting, these processes should therefore not be switched on at the same time. The basic rule is to use 1 and 2 for the inclusive generation of a $Z0/W+$ and 15, 16, 30 and 31 for the study of the $Z0/W+$ subsample with high transverse momentum.

The introduction of the $Z'0$ with full interference structure has caused some redundancy, in the sense that e.g. the process $q + q \rightarrow Z0/\gamma^*$ can be run either as subprocess 1 with full interference ($MSTP(43) = 3$) or as subprocess 141 including only the $Z0/\gamma^*$ part of the matrix element ($MSTP(44) = 4$). The only difference between the two options is that when run through subprocess 1, $Z0$ can be set to decay also to the charged Higgs, $Z0 \rightarrow H+ H-$.

Finally, a few remarks about Higgs production seem in order. The subprocess $t t \rightarrow H0$ (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 $g g \rightarrow t t \rightarrow H0$ (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, $V V \rightarrow H0$, 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. Finally, 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.

2.3. 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 ($Z0$, $W+$, $H0$, $Z'0$, $H+$ or $R0$) or QCD production of new flavours (ISUB = 12, 53, 81, 82), switches in the JETSET program may be used to this end; see section 2.8.

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^0 production (ISUB = 1).
- = 12 : $W^{+/-}$ production (ISUB = 2).
- = 13 : Z^0 + jet production (ISUB = 15, 30).
- = 14 : $W^{+/-}$ + jet production (ISUB = 16, 31).
- = 15 : pair production of different combinations of gamma, Z^0 and $W^{+/-}$ (except gamma + gamma; see MSEL = 10) (ISUB = 19, 20, 22, 23, 25).
- = 16 : H^0 production (ISUB = 3, 5, 8, 102).
- = 17 : H^0 + Z^0 or H^0 + $W^{+/-}$ (ISUB = 24, 26).
- = 21 : Z'^0 production (ISUB = 141).
- = 22 : $H^{+/-}$ production (ISUB = 142).
- = 23 : R^0 production (ISUB = 143).

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.

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 2.6, 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.

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 $\text{CKIN}(11) \leq \text{CKIN}(9)$ and $\text{CKIN}(12) \leq \text{CKIN}(10)$.

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-hat) values in a hard 2 -> 2 scattering, where theta-hat is the scattering angle in the rest frame of the hard interaction.
 CKIN(31), CKIN(32) : (D=2.,-1.) range of allowed m'-hat values, where m'-hat is the mass of the complete three- or four-body final state in 2 -> 3 or 2 -> 4 processes (while m-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(42) : (D+2.,-1.) range of allowed mass value for Z* or W* in processes H -> Z + Z* or H -> W + W*, * denoting a gauge boson below its nominal mass. This option only works if m_H is so small that not both gauge bosons can be produced on mass-shell (moduli the ordinary Breit-Wigner). IF CKIN(42) < 0., the upper limit is inactive, i.e. becomes m_H - m_Z (or m_H - m_W).

2.4. The General Switches and Parameters

In addition to the event information described in section 2.5, 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 alpha_strong at hard interaction, in the routine ULALPS.

= 0 : alpha_strong is fixed at value PARU(111).

= 1 : first order running alpha_strong.

= 2 : second order running alpha_strong.

MSTP(3) : (D=2) selection of Lambda value in alpha_strong for

MSTP(2) >= 1.

= 1 : Lambda is given by PARP(1) for hard interactions, by PARP(61)

- for spacelike showers and by PARJ(81) for timelike ones.
- = 2 : Lambda 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, = 0.16 GeV for DFLM1, = 0.26 GeV for DFLM2 and = 0.36 GeV for DFLM3, respectively (cf. MSTP(51)). This Lambda 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 Lambda 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 -> 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.
- = 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 $\alpha_{\text{strong}} Q^2$ argument, $\alpha_{\text{strong}} = \alpha_{\text{strong}}(\text{PARP}(33) \cdot Q^2)$.
- 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 `alpha_strong` value appearing in the threshold factor (which is not the same as the `alpha_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 `alpha_strong` value appearing in the threshold factor is taken to be running, with argument $Q^2 = m_Q \sqrt{(\sqrt{\hat{s}} - 2m_Q)^2 + \Gamma_Q^2}$. Here m_Q is the nominal heavy quark mass, Γ_Q is the width of the heavy quark mass distribution, and \hat{s} is the invariant mass-squared of the heavy quark pair. The Γ_Q value has to be stored by the user in `PARP(35)`. The regularization of `alpha_strong` at low Q^2 is given by `MSTP(36)`.
- `MSTP(36)` : (D=2) regularization of `alpha_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(41)` : (D=1) (C) master switch for all resonance decays (Z^0 , W^{+-} , H^0 , Z'^0 , H^{+-} , R).
 - = 0 : off.
 - = 1 : on.
- `MSTP(42)` : (D=0) (C) mass treatment of final state particles with finite width.
 - = 0 : particles put on mass-shell.
 - = 1 : mass generated according to Breit-Wigner.
- `MSTP(43)` : (D=3) treatment of Z^0/γ^* interference in matrix elements.
 - = 1 : only γ^* included.
 - = 2 : only Z^0 included.
 - = 3 : complete Z^0/γ^* structure (with interference) included.
- `MSTP(44)` : (D=7) treatment of $Z'^0/Z^0/\gamma^*$ interference in matrix elements.
 - = 1 : only γ^* included.
 - = 2 : only Z^0 included.
 - = 3 : only Z'^0 included.
 - = 4 : only Z^0/γ^* (with interference) included.
 - = 5 : only Z'^0/γ^* (with interference) included.
 - = 6 : only Z'^0/Z^0 (with interference) included.
 - = 7 : complete $Z'^0/Z^0/\gamma^*$ structure (with interference) included.
- `MSTP(45)` : (D=1) (C) angular orientation of W and Z decay products for $W W$, $Z Z$, $W Z$, $W \gamma$ and $Z \gamma$ pair processes, including pairs from H^0 decays.
 - = 0 : independent decay of two resonances, isotropic in CM frame of each resonance.
 - = 1 : correlated decay angular distributions according to proper matrix elements.
- `MSTP(51)` : (D=1) choice of structure functions; see also `MSTP(52)`. Note that, since all parametrizations have some region of applicability, the structure functions are assumed frozen 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.

= 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 1 ($\Lambda = 0.26$ GeV) for p, Owens set 1 for pi. Comment as for =11.

= 13 : Diemoz-Ferroni-Longo-Martinelli set 1 ($\Lambda = 0.36$ 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 in structure functions, and thus also for initial state spacelike showers.

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=0) 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_{Tmin} , 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) ≥ 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, Lambda_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) ≥ 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 $c_{\min} = 2 \cdot m_{\text{quark}} / \sqrt{s} = 0.6 \text{ GeV} / \sqrt{s}$.
 = 1 : 1 for meson, $2 \cdot (1 - \chi)$ for baryon, i.e. simple counting rules.
 = 2 : $(k+1) \cdot (1 - \chi)^k$, 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 PARP(94) or 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 PYZSTAT(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 : known; kinematics selection coefficients stored by user in COEF(ISUB,J) (J = 1 - 20) in /PYINT2/ and maximum of the corresponding differential cross-section times Jacobians in XSEC(ISUB,1) in /PYINT5/. This is to be done for each included subprocess ISUB before initialization, with the sum of all XSEC(ISUB,1) values, except for ISUB = 95, stored in XSEC(0,1).
 MSTP(122) : (D=1) initialization and differential cross-section maximization printout.

- = 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 2.6. This value should never be reduced, but may be increased at a later date if more complicated processes are included.

MSTP(131) : (D=0) master switch for overlaid 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 overlaid 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. It is thus not possible to have two rare events overlaid.

- = 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 overlaid events.

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

= 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 $n_{\text{bar}} = \sigma_{\text{soft}} * \text{PARP}(131)$ and $f = \sigma_{\text{hard}}/\sigma_{\text{soft}}$. The probability that a bunch crossing will give i events is then

$$\text{Prob}(i) = \exp(-n_{\text{bar}}) * n_{\text{bar}}^i / i! * f * i$$
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 $n_{\text{bar}} < 40$, i.e. that an average beam crossing does not contain more than 40 overlaid events. The multiplicity distribution is truncated above 100, or when the probability for a multiplicity has fallen below 10^{-6} , whichever occurs sooner. See also PARI(91) - PARI(93).

MSTP(134) : (D=1) a user selected multiplicity, i.e. total number of overlaid events, to be generated in the next PYTHIA call. May be reset for each new event, but must be in the range $1 \leq \text{MSTP}(134) \leq 100$.

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 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 [E1186].

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 Γ_Q for the heavy flavour studied in processes ISUB = 81 or 82; to be used for the threshold factor when MSTP(35) = 2.

PARP(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) Lambda value used in spacelike parton shower (see MSTP(64)).

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 α_{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 α_{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 $p_{T_{\text{min}}}$ 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 α_{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

$$\text{prob} = \text{PARP}(87) * (1 - (p_T^2 / (p_T^2 + \text{PARP}(88) * (\text{PARP}(82))^2))^2).$$

PARP(89) : (D=0.) minimum value of p_T allowed at initialization of multiple interactions. (Related to numerical precision problems.)

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 chi 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 chi 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 chi 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^PARP(98) behaviour in the splitting distribution.

PARP(101) : (D=-0.02 GeV^2) 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=0.4) fraction of total probability that is shared democratically between the COEF coefficients open for the given variable, with remainin fraction distributed according to the optimization results of PYMAXI.

PARP(131) : (D=0.01 mb^(-1)) in the overlaid 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 100 overlaid events, the initialization procedure will crash if nbar is above 40.

2.5. General Event Information

When an event is generated with PYTHIA, 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 2.6.

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 2.3, 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 $2 \rightarrow 2$ processes. With $i = \text{MSTI}(15)$, $j = \text{MSTI}(16)$ and $k = \text{MSTI}(2)$, the QCD possibilities are, in the classification scheme of [Ben84]:

ISUB = 11, $i = j$, $q_i q_i \rightarrow q_i q_i$;

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, $q_i q_{-i} \rightarrow q_l q_{-l}$;

k = 1 : only possibility.

ISUB = 13, $q_i q_{-i} \rightarrow g g$;

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, $g g \rightarrow q_l q_l$;

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.

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 $\text{MSTP}(125) = 0$.

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

MSTI(6) : current frame of event, cf. $\text{MSTP}(124)$.

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 overlaid events generated in latest PYTHIA call.
MSTI(42) - MSTI(50) : ISUB codes for the events 2 - 10 generated in the overlaid 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 overlaid 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-hat² 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-hat <= m'-hat <= E_{CM}.
PARI(20) : s'-hat = m'-hat²; 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) : Q² 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) : $\tau = \hat{s}/s = x_1 x_2 = \text{PARI}(33) \text{PARI}(34)$.

PARI(37) : $y^* = (1/2) * \text{alog}(x_1/x_2)$, i.e. rapidity of the hard interaction subsystem in the CM frame of the event-as-a-whole.

PARI(38) : $\tau' = \hat{s}'/s = \text{PARI}(20)/\text{PARI}(12)$.

PARI(41) : $\cos(\theta\text{-hat})$, where $\theta\text{-hat}$ is the scattering angle of a $2 \rightarrow 2$ (or $2 \rightarrow 1 \rightarrow 2$) 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.

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

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

PARI(93) : the probability that an event of the allowed type will be produced at a beam crossing. Only relevant for MSTP(133) = 1 or 2.

2.6. The Event Record

When an event is generated by a PYTHIA 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 7.2 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	
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/c^2).

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 $2 \rightarrow 2$ hard scattering, lines 7 and 8 give the two outgoing partons/particles from the hard interaction, before any final state radiation. For $2 \rightarrow 2$ processes proceeding via an intermediate resonance such as Z^0/γ^* , $W^{+/-}$, H^0 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 Z^0 coming from decay of H^0
14	21	12	12	electron neutrino from first Z^0 decay
15	21	-12	12	electron antineutrino from first Z^0 decay
16	21	5	13	b quark from second Z^0 decay
17	21	-5	13	b^- quark from second Z^0 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 7.2 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.

2.7. 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 PYOVLY

Purpose: to determine the number of overlaid 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 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 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 PYDIFF

Purpose: to handle diffractive and elastic scattering events.

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.

SUBROUTINE PYSIGH

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^- .

FUNCTION PYGAMM(X)

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

SUBROUTINE PYSPLI

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

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, Q²). 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' -> q" q'" H0).

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 and 82.

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 primary documentation lines before resonance decays 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(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 overlaid events selected.

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

MINT(83) : number of lines in the event record already filled by previously considered overlaid 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(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) : sum of final state weights for the different pieces contributing to γ^*/Z^0 or $\gamma^*/Z^0/Z'^0$ production, 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(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 overlaid 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(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) : $1 - (\text{sum of x-values})$ 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 turnoff 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 \rightarrow 1 processes (irrespective of subsequent decays).
- = 2 : 2 \rightarrow 2 processes (i.e. the bulk of processes).
- = 3 : 2 \rightarrow 3 processes (like $q q' \rightarrow q'' q''' H^0$).
- = 4 : 2 \rightarrow 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.

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

$J = 2$: τ selected according to $1/\tau^2$.

$J = 3$: τ selected according to $1/(\tau(\tau+\tau_r))$, where $\tau_r = m^2/s$ is τ value of resonance; only used for resonance production.

$J = 4$: τ selected according to Breit-Wigner; only used for resonance production.

$J = 5$: τ selected according to $1/(\tau(\tau+\tau'_r))$, where $\tau'_r = m'^2/s$ is τ value of second resonance; only used for simultaneous production of two resonances.

$J = 6$: τ selected according to second Breit-Wigner; only used 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 $1/(1-z)$.

$J = 12$: $z = \cos(\theta_{\text{hat}})$ selected according to $1/(1+z)$.

$J = 13$: $z = \cos(\theta_{\text{hat}})$ selected according to $1/(1-z)^2$.

$J = 14$: $z = \cos(\theta_{\text{hat}})$ selected according to $1/(1+z)^2$.

$J = 15$: 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 -> 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.

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 2.8.

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 2.8. 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 $Z^0 Z^0$).

J = 2 : suppression for a particle of type KF when produced on its own, or together with a particle of another type.

J = 3 : suppression for an antiparticle of type KF when produced on its own, or together with a particle of another type.

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.

2.8. 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 $\text{MSTP}(111) = 0$, fragmentation and decays are switched off in the PYTHIA 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 PYTHIA for $\text{MSTP}(111) = 1$, can be controlled by the switches and parameters in the LUDAT1 common block:

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

Thus, $\text{MSTJ}(21) = 0$ will switch off particle decays but keep jet fragmentation on, $\text{MSTJ}(1)$ 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 ever involved.

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=60. GeV/c^2) mass of top quark.
PMAS(7,1) : (D=120. GeV/c^2) mass of l quark.
PMAS(8,1) : (D=200. GeV/c^2) mass of h quark.
PMAS(17,1) : (D=60. GeV/c^2) mass of chi lepton.
PMAS(23,1) : (D=92.4 GeV/c^2) Z0 nominal mass.
PMAS(24,1) : (D=81. GeV/c^2) W+/- nominal mass.
PMAS(25,1) : (D=15. GeV/c^2) H0 nominal mass.
PMAS(32,1) : (D=900. GeV/c^2) Z'0 nominal mass.
PMAS(37,1) : (D=300. GeV/c^2) H+/- nominal mass.
PMAS(40,1) : (D=5000. GeV/c^2) R nominal mass.
```

For Z_0 , $W_{+/-}$, H_0 , R , $H_{+/-}$, and Z'_0 , the resonance width is calculated at PYINIT initialization, taking into account the channels assumed possible for resonance decays (for H_0 this will depend very much on the mass assumed), so the nominal widths stored in $\text{PMAS}(KF,2)$ are actually not used for this purpose.

Note that the first index in the KCHG and PMAS arrays is not the standard KF code described in section 2.6, 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. Normally, this option is used for decays involving fourth generation or H^\pm particles.
- = 0 : channel is switched off.
- = 1 : channel is switched on.
- = 2 : channel is switched on for a particle but off for an antiparticle. It is also on for a particle 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 $H0 \rightarrow W^+ W^-$, 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.

All the options above may be freely mixed. The difference, for those cases where both make sense, between using values 2 and 3 and using 4 and 5 is that the latter automatically include charge conjugate states, e.g. $H^0 \rightarrow W^+ W^- \rightarrow e^+ \nu_e \bar{d} \text{ or } \bar{d} u e^- \nu_e \text{bar}$, but the former only one of them. In calculations of the joint branching ratio, this makes a factor 2 difference.

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.

2.9. On cross-sections

As already mentioned, the routine PYSTAT can be used to print out a table of cross-sections in the current run. A few comments about these cross-sections may be in order. We will first consider high- p_T event generation only, and later cover low- p_T physics.

At initialization, maxima for the various differential cross-sections are searched for (for efficiency reasons, the set of independent kinematical variables is not τ , y^* and t -hat, but rather a suitably transformed set of these, for which the differential cross-section is less rapidly varying). In case several different high- p_T processes are allowed, these maxima are used to find the initial relative probability for each process. In order to generate an event, first a process is chosen, thereafter a set of kinematical variables for this process is selected. The probability for accepting this set is given by the differential cross-section in this point divided by the maximum 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 differential cross-section in these points (multiplied by the integral of the allowed phase space, which is just a simple multiplicative factor). At the end of each PYTHIA call, the ratio of these two numbers is calculated, 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 $q + qb \rightarrow Z0/\gamma^* \rightarrow \mu^+ + \mu^-$, 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 $Z0$ 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. The $H0$ is a special case; it can either decay directly into a $q\text{-}q\bar{}$ or $l\text{-}l\bar{}$ pair, or into a $Z0Z0$ or W^+W^- pair, which then can decay further. Therefore the cross-section for Higgs production given by the program is reduced, not only by switching off some decay channels of the $H0$ itself, but also by switching off subsequent Z or W decay modes (and the branching ratios for $H0$ decays into $Z0Z0$ or W^+W^- are reduced accordingly). This is also true for $Z0$ when allowed to decay into a H^+H^- pair. 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 IPROD values.

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).

2.10. 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 PYTHIA                                     ! generate event
IF(IEVENT.EQ.1) CALL LULIST(2)                ! list first event
...           ! insert desired analysis chain for
...           ! each event
100 CONTINUE
CALL PYSTAT(1)                                ! print cross-sections
...           ! user output
END
```

References

- 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
- 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
- DFL88 M. Diemoz, F. Ferroni, E. Longo, G. Martinelli, Z. Physik C39
(1988) 21
- Ell86 R. K. Ellis, J. C. Sexton, Nucl. Phys. B269 (1986) 445
- 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
- PDG88 Particle Data Group, G. P. Yost et al.,
Phys. Lett. 204B (1988) 1
- Sjo89 JETSET 7.2 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
