

# High-Energy-Physics Event Generation with PYTHIA 5.7 and JETSET 7.4

Torbjörn Sjöstrand  
Theory Division, CERN  
CH-1211 Geneva 23  
Switzerland

## Abstract

PYTHIA and JETSET are the two main components of the ‘Lund Monte Carlo’ program suite. They can be used to generate high-energy-physics ‘events’, i.e. sets of outgoing particles produced in the interactions between two incoming particles. Ideally these events should have the same average behaviour and the same fluctuations as real data. The underlying physics is not understood well enough to give an exact description; the programs therefore contain a combination of analytical results and various models. Several event classes are considered, within and beyond the standard model. We give a summary of the main physics component of the current versions, PYTHIA 5.7 and JETSET 7.4: hard-scattering matrix elements, parton distributions, initial- and final-state radiation, multiple interactions, beam remnants, fragmentation and decays. A brief outline is also given of some programming aspects. A detailed physics description and manual is available as a separate publication.

*Submitted to Computer Physics Communications*

# New Version Summary

*Title of programs:* PYTHIA 5.7 and JETSET 7.4

*Catalogue number:* —

*Programs obtainable from:* CPC Program Library, Queen's University of Belfast, N. Ireland (see application form in this issue)

*References to most recent previous versions:* Computer Physics Communications **46** (1987) 43 and Computer Physics Communications **43** (1987) 367, respectively

*Authors of most recent previous versions:* H.-U. Bengtsson and T. Sjöstrand, and T. Sjöstrand and M. Bengtsson, respectively

*Do the new versions supersede the previous versions?:* yes

*Licensing provisions:* none

*Computers for which the new versions are designed and others on which they have been tested:* IBM 9000, SUN SPARC ELC, and any other machine with a Fortran 77 compiler

*Installation:* CERN

*Operating systems under which the new versions have been tested:* VM/ESA, Unix

*Programming language used in the new versions:* Fortran 77; is also fully compatible with Fortran 90, i.e. does not make use of any obsolescent features of the Fortran 90 standard

*Memory required to execute with typical data:* 420 kwords

*No. of bits in word:* 32

*No. of processors used:* 1

*Has the code been vectorized?:* no

*No. of lines in distributed programs:* 3 + 18507 + 11261 (test program + PYTHIA + JETSET)

*CPC Program Library subprograms used:* none

*Separate documentation available:* yes; *No. of pages:* about 300

*Keywords:* QCD, standard model, beyond standard model, hard scattering,  $e^+e^-$  annihilation, lepton production, photoproduction, hadronic processes, high- $p_\perp$  scattering, prompt photons, gauge bosons, Higgs physics, parton distribution functions, jet production, parton showers, fragmentation, hadronization, beam remnants, multiple interactions, particle decays, event measures

*Nature of physical problem:* high-energy collisions between elementary particles normally give rise to complex final states, with large multiplicities of hadrons, leptons, neutrinos and photons. The relation between these final states and the underlying physics description is not a simple one, for two main reasons. Firstly, we do not even in principle have a complete understanding of the physics. Secondly, any analytical approach is made untractable by the large multiplicities.

*Method of solution:* complete events are generated by Monte Carlo methods. The complexity is mastered by a subdivision of the full problem into a set of simpler separate tasks. All main aspects of the events are simulated, such as hard-process selection, initial- and final-state radiation, beam remnants, fragmentation, decays, and so on. Therefore events should be directly comparable with experimentally observable ones. The programs can be used to extract physics from comparisons with existing data, or to study physics at future experiments.

*Reasons for the new versions:* the programs have been significantly rewritten and expanded compared to previous versions

*Restrictions on the complexity of the problem:* at very high energies the program may break down for one of two reasons: either the number of particles may exceed the memory space available, or the (mainly) single-precision kinematics may give unacceptable roundoff errors. Many physics restrictions exist, as already noted.

*Typical running time:* 1–100 events per second, depending on process studied

*Unusual features of the program:* none

---

## 1 Introduction

The PYTHIA [1] and JETSET [2] programs are frequently used for event generation in high-energy physics. The emphasis is on multiparticle production in collisions between elementary particles. This in particular means hard interactions in  $e^+e^-$ , pp and ep colliders, although also other applications are envisaged. The programs are intended to generate complete events, in as much detail as experimentally observable ones, within the bounds of our current understanding of the underlying physics. Many of the components of the programs represent original research, in the sense that models have been developed and implemented for a number of aspects not covered by standard theory. Although originally conceived separately, the PYTHIA and JETSET programs today are so often used together that it makes sense to present them here without too much distinction.

Both programs have a long history, and several manuals have come out. The previous round of PYTHIA/JETSET program descriptions is by now over six years old. Meanwhile a large number of additions and changes have been made. Some of the main ones are:

- The generic hard process generation machinery can now be applied equally well to hadron–hadron, hadron–lepton and lepton–lepton interactions, rather than only the first of the three. Also incoming photons are (partly) supported.
- The number of hard scattering subprocesses has almost tripled.
- The treatment of resonances has been improved with  $s$ -dependent widths etc.
- A number of new features have been added to the initial- and final-state radiation machineries, such as photon radiation, various polar angle restrictions, and azimuthal anisotropies.
- Parametrizations of total cross sections and parton distributions have been updated.
- Fragmentation parameters and the particle decay description have been updated to take into account new measurements. In particular, the treatment of top decay is completely changed.
- Particle codes have been changed to the PDG standard.
- Common blocks have been expanded and reorganized to contain much new information, such as the position of secondary vertices.
- A huge number of new options have been added, to allow for a more flexible selection of subprocess properties (flavours, kinematics, etc.) and general physics assumptions.

In the process, the total size of the program code has more than doubled.

This report contains a brief summary of the current status of the two programs, with respect to physics (section 2) and programming (section 3). Both aspects are considered in much more detail in the long writeup [3], to which interested users are asked to turn.

## 2 Physics Overview

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

1. Initially two beam particles are coming in towards each other. Normally each particle is characterized by a set of parton distributions, which defines the partonic substructure in terms of flavour composition and energy sharing.
2. One shower initiator parton from each beam starts off a sequence of branchings, such as  $q \rightarrow qg$ , which build up an initial-state shower.
3. One incoming parton from each of the two showers enters the hard process, where then a number of outgoing partons are produced, usually two. It is the nature of this process that determines the main characteristics of the event.
4. Also the outgoing partons may branch, to build up final-state showers.
5. In addition to the hard process considered above, further semihard interactions may occur between the other partons of two incoming hadrons.
6. When a shower initiator is taken out of a beam particle, a beam remnant is left behind. This remnant may have an internal structure, and a net colour charge that relates it to the rest of the final state.
7. The QCD confinement mechanism ensures that the outgoing quarks and gluons are not observable, but instead fragment to colour neutral hadrons.
8. Many of the produced hadrons are unstable and decay further.

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

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

### 2.1 Hard Processes and Parton Distribution Functions

In JETSET, only two hard processes are available. The first and main one is  $e^+e^- \rightarrow \gamma^*/Z^0 \rightarrow q\bar{q}$ . The quark  $q$  in the final state may be  $d, u, s, c, b$  or  $t$ ; the flavour in each event is picked at random, according to the relative couplings, evaluated at the hadronic c.m. energy. Also the angular distribution of the final  $q\bar{q}$  pair is included.

The other JETSET process is a routine to generate  $ggg$  and  $\gamma gg$  final states, as expected in onium  $1^{--}$  decays such as  $\Upsilon$ . Note that toponium will decay weakly much too fast for these processes to be of any interest.

PYTHIA contains a much richer selection, with close to a hundred different hard processes, see Table 1. These may be classified in many different ways.

One is according to the number of final-state objects: we speak of ‘ $2 \rightarrow 1$ ’ processes, ‘ $2 \rightarrow 2$ ’ ones, ‘ $2 \rightarrow 3$ ’ ones, etc. This aspect is very relevant from a programming point of view: the more particles in the final state, the more complicated the phase space and therefore the whole generation procedure. In fact, PYTHIA is optimized for  $2 \rightarrow 1$  and

Table 1: Subprocesses, according to the subprocess numbering of PYTHIA. ‘f’ denotes a fermion (quark or lepton), ‘Q’ a heavy quark and ‘F’ a heavy fermion. See text for some other notation.

No.	Subprocess	No.	Subprocess	No.	Subprocess
1	$f_i\bar{f}_i \rightarrow \gamma^*/Z^0$	70	$\gamma W^\pm \rightarrow Z^0 W^\pm$	131	$gg \rightarrow Z^0 Q_k \bar{Q}_k$
2	$f_i\bar{f}_j \rightarrow W^\pm$	71	$Z_L^0 Z_L^0 \rightarrow Z_L^0 Z_L^0$	141	$f_i\bar{f}_i \rightarrow \gamma/Z^0/Z'^0$
3	$f_i\bar{f}_i \rightarrow H^0$	72	$Z_L^0 Z_L^0 \rightarrow W_L^+ W_L^-$	142	$f_i\bar{f}_j \rightarrow W'^+$
5	$Z^0 Z^0 \rightarrow H^0$	73	$Z_L^0 W_L^\pm \rightarrow Z_L^0 W_L^\pm$	143	$f_i\bar{f}_j \rightarrow H^+$
8	$W^+ W^- \rightarrow H^0$	76	$W_L^+ W_L^- \rightarrow Z_L^0 Z_L^0$	144	$f_i\bar{f}_j \rightarrow R$
10	$f_i f_j \rightarrow f_i f_j$ (QFD)	77	$W_L^\pm W_L^\pm \rightarrow W_L^\pm W_L^\pm$	145	$q_i \bar{\ell}_j \rightarrow L_Q$
11	$f_i f_j \rightarrow f_i f_j$ (QCD)	80	$q_i \gamma \rightarrow q_k \pi^\pm$	147	$dg \rightarrow d^*$
12	$f_i\bar{f}_i \rightarrow f_k\bar{f}_k$	81	$f_i\bar{f}_i \rightarrow Q_k \bar{Q}_k$	148	$ug \rightarrow u^*$
13	$f_i\bar{f}_i \rightarrow gg$	82	$gg \rightarrow Q_k \bar{Q}_k$	149	$gg \rightarrow \eta_{\text{techni}}$
14	$f_i\bar{f}_i \rightarrow g\gamma$	83	$q_i f_j \rightarrow Q_k f_l$	151	$f_i\bar{f}_i \rightarrow H'^0$
15	$f_i\bar{f}_i \rightarrow gZ^0$	84	$g\gamma \rightarrow Q_k \bar{Q}_k$	152	$gg \rightarrow H^0$
16	$f_i\bar{f}_j \rightarrow gW^\pm$	85	$\gamma\gamma \rightarrow F_k \bar{F}_k$	153	$\gamma\gamma \rightarrow H^0$
18	$f_i\bar{f}_i \rightarrow \gamma\gamma$	86	$gg \rightarrow J/\psi g$	156	$f_i\bar{f}_i \rightarrow A^0$
19	$f_i\bar{f}_i \rightarrow \gamma Z^0$	87	$gg \rightarrow \chi_{0c} g$	157	$gg \rightarrow A^0$
20	$f_i\bar{f}_j \rightarrow \gamma W^\pm$	88	$gg \rightarrow \chi_{1c} g$	158	$\gamma\gamma \rightarrow A^0$
22	$f_i\bar{f}_i \rightarrow Z^0 Z^0$	89	$gg \rightarrow \chi_{2c} g$	161	$f_i g \rightarrow f_k H^+$
23	$f_i\bar{f}_j \rightarrow Z^0 W^\pm$	91	elastic scattering	162	$qg \rightarrow \ell L_Q$
24	$f_i\bar{f}_i \rightarrow Z^0 H^0$	92	single diffraction ( $XB$ )	163	$gg \rightarrow L_Q \bar{L}_Q$
25	$f_i\bar{f}_i \rightarrow W^+ W^-$	93	single diffraction ( $AX$ )	164	$q_i \bar{q}_i \rightarrow L_Q \bar{L}_Q$
26	$f_i\bar{f}_j \rightarrow W^\pm H^0$	94	double diffraction	165	$f_i\bar{f}_i (\rightarrow \gamma^*/Z^0) \rightarrow f_k\bar{f}_k$
28	$f_i g \rightarrow f_i g$	95	low- $p_\perp$ production	166	$f_i\bar{f}_j (\rightarrow W^\pm) \rightarrow f_k\bar{f}_l$
29	$f_i g \rightarrow f_i \gamma$	102	$gg \rightarrow H^0$	171	$f_i\bar{f}_i \rightarrow Z^0 H^0$
30	$f_i g \rightarrow f_i Z^0$	103	$\gamma\gamma \rightarrow H^0$	172	$f_i\bar{f}_j \rightarrow W^\pm H^0$
31	$f_i g \rightarrow f_k W^\pm$	110	$f_i\bar{f}_i \rightarrow \gamma H^0$	173	$f_i f_j \rightarrow f_i f_j H^0$
33	$f_i \gamma \rightarrow f_i g$	111	$f_i\bar{f}_i \rightarrow g H^0$	174	$f_i f_j \rightarrow f_k f_l H^0$
34	$f_i \gamma \rightarrow f_i \gamma$	112	$f_i g \rightarrow f_i H^0$	176	$f_i\bar{f}_i \rightarrow Z^0 A^0$
35	$f_i \gamma \rightarrow f_i Z^0$	113	$gg \rightarrow g H^0$	177	$f_i\bar{f}_j \rightarrow W^\pm A^0$
36	$f_i \gamma \rightarrow f_k W^\pm$	114	$gg \rightarrow \gamma\gamma$	178	$f_i f_j \rightarrow f_i f_j A^0$
53	$gg \rightarrow f_k\bar{f}_k$	115	$gg \rightarrow g\gamma$	179	$f_i f_j \rightarrow f_k f_l A^0$
54	$g\gamma \rightarrow f_k\bar{f}_k$	121	$gg \rightarrow Q_k \bar{Q}_k H^0$	181	$gg \rightarrow Q_k \bar{Q}_k H'^0$
58	$\gamma\gamma \rightarrow f_k\bar{f}_k$	122	$q_i \bar{q}_i \rightarrow Q_k \bar{Q}_k H^0$	182	$q_i \bar{q}_i \rightarrow Q_k \bar{Q}_k H'^0$
68	$gg \rightarrow gg$	123	$f_i f_j \rightarrow f_i f_j H^0$	186	$gg \rightarrow Q_k \bar{Q}_k A^0$
69	$\gamma\gamma \rightarrow W^+ W^-$	124	$f_i f_j \rightarrow f_k f_l H^0$	187	$q_i \bar{q}_i \rightarrow Q_k \bar{Q}_k A^0$

$2 \rightarrow 2$  processes. Often one can come quite far with that, since showers will add the required extra activity. The classification may also be misleading at times, since an  $s$ -channel resonance is considered as a single particle, even if it is assumed always to decay into two final-state particles. Thus the process  $e^+e^- \rightarrow W^+W^- \rightarrow q_1\bar{q}_2 q_3\bar{q}_4$  is classified as  $2 \rightarrow 2$ , although the decay treatment of the  $W$  pair includes the full  $2 \rightarrow 4$  matrix elements.

Another classification is according to the physics scenario. The following major groups may be distinguished:

- Hard QCD processes, e.g.  $qg \rightarrow qg$ .

- Soft QCD processes, such as diffractive and elastic scattering, and minimum-bias events.
- Heavy-flavour production, e.g.  $gg \rightarrow t\bar{t}$ .
- Prompt-photon production, e.g.  $qg \rightarrow q\gamma$ .
- Photon-induced processes, e.g.  $\gamma g \rightarrow q\bar{q}$ .
- Deep inelastic scattering, e.g.  $q\ell \rightarrow q\ell$ .
- W/Z production, such as the  $e^+e^- \rightarrow \gamma^*/Z^0$  already found in JETSET, or  $q\bar{q} \rightarrow W^+W^-$ .
- Standard model Higgs production, where the Higgs is reasonably light and narrow, and can therefore still be considered as a resonance.
- Gauge boson scattering processes, such as  $W_L W_L \rightarrow W_L W_L$  (L = longitudinal), when the standard model Higgs is so heavy and broad that resonant and non-resonant contributions have to be considered together.
- Non-standard Higgs particle production, within the framework of a two-Higgs-doublet scenario with three neutral ( $H^0$ ,  $H'^0$  and  $A^0$ ) and two charged ( $H^\pm$ ) Higgs states.
- Production of new gauge bosons, such as a  $Z'$ ,  $W'$  and R (a horizontal boson, coupling between generations).
- Production of fourth-generation fermions.
- Leptoquark ( $L_Q$ ) production.
- Technicolour, e.g.  $gg \rightarrow \eta_{\text{techni}}$ .
- Compositeness, e.g.  $d^*$  and  $u^*$  production.
- Other deviations from standard model processes, e.g. due to contact interactions or a strongly interacting gauge boson sector.

Scenarios do not always appear as separate processes, but may just be options to some of the other processes above. The list is by no means a survey of all interesting physics. Most notable is the absence of supersymmetric particle production and decay, but many other examples could be found. Also, within the scenarios studied, not all contributing graphs have always been included, but only the more important and/or more interesting ones. In many cases, various approximations are involved in the matrix elements coded.

Note that each process in Table 1 may appear in many disguises. The two processes  $q\bar{q} \rightarrow \gamma^*/Z^0 (\rightarrow e^+e^-)$ , studied in hadron colliders, and  $e^+e^- \rightarrow \gamma^*/Z^0 (\rightarrow q\bar{q})$ , studied in  $e^+e^-$  colliders, are really special cases of a common process,  $f\bar{f} \rightarrow \gamma^*/Z^0 (\rightarrow f'\bar{f}')$ , where  $f$  denotes a fundamental fermion, i.e. a quark, lepton or neutrino. The whole structure is therefore only coded once, and then slightly different couplings and colour prefactors are used, depending on the initial (and final) state considered. Usually the interesting cross-section is a sum over several different initial (and final) states, e.g.  $u\bar{u} \rightarrow \gamma^*/Z^0$  and  $d\bar{d} \rightarrow \gamma^*/Z^0$  in a hadron collider. This kind of summation is always implicitly done.

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

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

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

hard scattering therefore only involves a squared invariant mass  $\hat{s} = x_1 x_2 s = x_1 x_2 E_{\text{cm}}^2$ , where  $E_{\text{cm}}$  is the c.m. energy of the event.

Parton distribution functions are most familiar for hadrons, such as the proton. Hadrons are inherently composite objects, made up of quarks and gluons. Since we do not understand QCD in this region, a derivation from first principles of the parton distribution functions of hadrons does not yet exist. It is therefore necessary to rely on parametrizations, where experimental data are used in conjunction with the evolution equations for the  $Q^2$  dependence, to pin down the parton distributions. Some fits are available in PYTHIA, while more can be accessed by an interface to PDFLIB [4].

Also for fundamental particles, such as the electron, is it convenient to introduce parton distributions. The function  $f_e^e(x)$  thus parametrizes the probability that the electron that takes part in the hard process retains a fraction  $x$  of the original energy, the rest being radiated (into photons) in the initial state. The electron may also contain photons, and the photon may in turn contain quarks and gluons; the distributions of the latter are not fully perturbatively calculable.

## 2.2 Initial- and Final-State Radiation

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

Two traditional approaches exist to the modelling of perturbative corrections. One is the matrix-element method, in which Feynman diagrams are calculated, order by order. In principle, this is the correct approach, which takes into account exact kinematics, and the full interference and helicity structure. The only problem is that calculations become increasingly difficult in higher orders, in particular for the loop graphs. Only in exceptional cases have therefore more than one loop been calculated in full, and we often have no loop corrections at all at our disposal. On the other hand, we have indirect but strong evidence that, in fact, the emission of multiple soft gluons plays a significant rôle in building up the event structure, e.g. at LEP, and this sets a limit to the applicability of matrix elements.

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

Second-order matrix elements are available to give the QCD final-state radiation in the JETSET implementation of the process  $e^+e^- \rightarrow \gamma^*/Z^0 \rightarrow q\bar{q}$ , as an option to the default parton shower picture. In the definition of 3- or 4-jet events, a cut is introduced whereby it is required that any two partons have an invariant mass bigger than some fraction of the c.m. energy.

By contrast, PYTHIA does not contain any full higher-order matrix elements, with

loop contributions included. There are a few cases where higher-order matrix elements are included at the Born level, such as the two first-order processes  $qg \rightarrow Wq'$  and  $q\bar{q}' \rightarrow Wg$ . The cross sections for these processes are divergent when the  $p_{\perp} \rightarrow 0$ . Depending on the physics application, one could then use PYTHIA in one of two ways. In the region of small  $p_{\perp}$ , the preferred option is lowest-order matrix elements,  $q\bar{q}' \rightarrow W$ , combined with parton showers. For the production of a  $W$  at large  $p_{\perp}$ , on the other hand, it is advantageous to generate first-order events, and then add showers only to describe additional softer radiation.

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

In both initial- and final-state showers, the structure is given in terms of branchings  $a \rightarrow bc$ , specifically  $e \rightarrow e\gamma$ ,  $q \rightarrow qg$ ,  $q \rightarrow q\gamma$ ,  $g \rightarrow gg$ , and  $g \rightarrow q\bar{q}$ . Each of these processes is characterized by a splitting kernel  $P_{a \rightarrow bc}(z)$ . The branching rate is proportional to the integral  $\int P_{a \rightarrow bc}(z) dz$ . The  $z$  value picked for a branching describes the energy sharing, with daughter  $b$  taking a fraction  $z$  and daughter  $c$  the remaining  $1 - z$  of the  $a$  energy. Once formed, the daughters  $b$  and  $c$  may in turn branch, and so on.

Each parton is characterized by some virtuality scale  $Q^2$ , which gives an approximate sense of time ordering to the cascade. In the initial-state shower,  $Q^2$  values are space-like ( $m^2 < 0$ ) and gradually increasing as the hard scattering is approached, while  $Q^2$  is time-like ( $m^2 > 0$ ) and decreasing in the final-state showers. Shower evolution is cut off at some lower scale  $Q_0$ , typically around 1 GeV for QCD branchings. The same cut-off scale is also used to regularize the soft-gluon-emission divergences in the splitting kernels. From above, a maximum scale  $Q_{\max}$  is introduced, where the showers are matched to the hard interaction itself. The relation between  $Q_{\max}$  and the kinematics of the hard scattering is uncertain, and the choice made can strongly affect the amount of well-separated jets.

Despite a number of common traits, the initial- and final-state radiation machineries are in fact quite different, which is reflected in the interpretation of  $Q^2$  and  $z$ , in the implementation of coherence effects, and so on. For historical reasons, the final-state shower routine is found in JETSET [5] and the initial-state one in PYTHIA [6].

## 2.3 Beam Remnants and Multiple Interactions

In a hadron-hadron collision, the initial-state radiation algorithm reconstructs one shower initiator in each beam, by backwards evolution from the hard scattering. This initiator only takes some fraction of the total beam energy, leaving behind a beam remnant that takes the rest. For a proton beam, a  $u$  quark initiator would leave behind a  $ud$  diquark beam remnant, with an antitriplet colour charge. The remnant is therefore colour-connected to the hard interaction, and forms part of the same fragmenting system. It is further customary to assign a primordial transverse momentum to the shower initiator, to take into account the motion of quarks inside the original hadron, basically as required by the uncertainty principle. This primordial  $k_{\perp}$  is selected according to some suitable distribution, and the recoil is assumed to be taken up by the beam remnant.

Often the remnant is more complicated, e.g. a  $g$  initiator would leave behind a  $uud$

proton-remnant system in a colour octet state, which can conveniently be subdivided into a colour triplet quark and a colour antitriplet diquark, each of which are colour-connected to the hard interaction. The energy sharing between these two remnant objects, and their relative transverse momentum, introduces additional degrees of freedom.

One would expect an ep event to have only one beam remnant, and an  $e^+e^-$  event none. This is not always correct, e.g. a  $\gamma\gamma \rightarrow q\bar{q}$  interaction in an  $e^+e^-$  event would leave behind the  $e^+$  and  $e^-$  as beam remnants.

So far we have assumed that each event only contains one hard interaction, i.e. that each incoming particle has only one parton that takes part in hard processes, and that all other constituents sail through unaffected. This is appropriate in  $e^+e^-$  or ep events, but not necessarily so in hadron-hadron collisions. Here each of the beam particles contains a multitude of partons, and so the probability for several interactions in one and the same event need not be negligible. The dominant mechanism is expected to be that disjoint pairs of partons, with one parton from each beam, undergo  $2 \rightarrow 2$  scatterings; this is what is considered in PYTHIA [7].

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

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

The understanding of multiple interaction is still very primitive, and even the experimental evidence that it exists at all is rather weak. PYTHIA therefore contains several different options, with a fairly simple one as default. The options differ in particular on the issue of the ‘pedestal’ effect: is there or not an increased probability for additional interactions in an event that is known to contain a hard scattering, compared with one that contains no hard interactions?

## 2.4 Fragmentation

QCD perturbation theory, formulated in terms of quarks and gluons, is valid at short distances. At long distances, QCD becomes strongly interacting and perturbation theory breaks down. In this confinement régime, the coloured partons are transformed into colourless hadrons, a process called either hadronization or fragmentation.

The fragmentation process has yet to be understood from first principles, starting from the QCD Lagrangian. This has left the way clear for the development of a number of different phenomenological models. JETSET is intimately connected with string fragmentation, in the form of the time-honoured ‘Lund model’ [8]. This is the default for all JETSET/PYTHIA applications, but independent fragmentation options also exist, for applications where one wishes to study the importance of string effects.

To understand fragmentation models, it is useful to start with the simplest possible system, a colour-singlet  $q\bar{q}$  2-jet event, as produced in  $e^+e^-$  annihilation. The assumption

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

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

In order to produce the quark–antiquark pairs  $q'\bar{q}'$  leading to string break-ups, the Lund model invokes the idea of quantum mechanical tunnelling. This leads to a flavour-independent Gaussian spectrum for the  $p_{\perp}$  of  $q'\bar{q}'$  pairs, with local compensation between the quark and the antiquark of the pair. The total  $p_{\perp}$  of a hadron is made up out of the  $p_{\perp}$  contributions from the quark and antiquark that together form the hadron.

The tunnelling picture also implies a suppression of heavy-quark production,  $u : d : s : c \approx 1 : 1 : 0.3 : 10^{-11}$ . When the quark and antiquark from two adjacent string breakings are combined to form a meson, it is necessary to choose between the different allowed possibilities, notably between pseudoscalar and vector mesons. This is not well understood in string language, but can be parametrized.

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

In general, the different string breaks are causally disconnected. This means that it is possible to describe the breaks in any convenient order. One therefore is led to write down an iterative scheme for the fragmentation, starting from the endpoints of the string and working inwards. Symmetry requirements constrain the allowed shape of the fragmentation function  $f(z)$ , where  $z$  is the fraction of the remaining light-cone momentum taken by each new particle. The resulting ‘Lund symmetric fragmentation function’ has two free parameters, which are determined from data. Modifications are necessary for heavy flavours at the endpoints of the string.

If several partons are moving apart from a common origin, the details of the string fragmentation become more complicated [9]. For a  $q\bar{q}g$  event, a string is stretched from the  $q$  end via the  $g$  to the  $\bar{q}$  end, i.e. the gluon is a kink on the string, carrying energy and momentum. As a consequence, the gluon has two string pieces attached, and the ratio of gluon to quark string force is 2, a number that can be compared with the ratio of colour charge Casimir operators,  $N_C/C_F = 2/(1 - 1/N_C^2) = 9/4$ . In this, as in other respects,

the string model can be viewed as a variant of QCD where the number of colours  $N_C$  is not 3 but infinite.

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

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

## 2.5 Decays

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

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

The ordinary decay treatment is handled by JETSET, making use of a set of tables where branching ratios and decay modes are stored. In PYTHIA a separate decay treatment exists, used exclusively for a specific list of resonances:  $Z^0$ ,  $W^\pm$ ,  $H^0$ ,  $Z'^0$ ,  $W'^\pm$ ,  $H'^0$ ,  $A^0$ ,  $H^\pm$ ,  $R^0$ ,  $q^*$ ,  $\ell^*$ ,  $L_Q$  and  $\eta_{\text{techni}}$ . What characterizes a (PYTHIA) resonance is that partial widths and branching ratios are calculated dynamically, as a function of the actual mass of a particle.

The decay products of PYTHIA resonances are typically quarks, leptons, or other resonances, e.g.  $W \rightarrow q\bar{q}'$  or  $H^0 \rightarrow W^+W^-$ . In decays to quarks, parton showers are automatically added to give a more realistic multijet structure, and one may also allow photon emission off leptons. If the decay products in turn are resonances, further decays are necessary. Often spin information is available in resonance decay matrix elements,

contrary to the normal state of affairs in ordinary particle decays. This means that the angular orientations in the two decays of a  $W^+W^-$  pair are properly correlated. Occasionally, the information is not available, and then resonances decay isotropically.

The top quark is a special problem. The original machinery was based on the assumption that the  $t$  is long-lived, so that top hadrons have time to form in the fragmentation process, and afterwards these hadrons decay weakly. With current ‘best bet’ mass values, this is not correct, but one should rather consider top decay before fragmentation. This is therefore now the default mode, which means that the top (and also any fourth generation fermions) are treated like the resonances listed above.

### 3 Program Overview

Since the JETSET and PYTHIA programs today are so closely connected, and are gradually coalescing, they are presented together in this report. However, they still appear as separate entities, with slightly different style and emphasis.

JETSET is the older of the two, and is at the origin of the whole ‘Lund’ family of event generators. It can be subdivided in two parts. The larger is a generic package for jet fragmentation, particle decays, final-state parton showers, event-analysis routines, and other utilities. This package can be used in the context of any hard process, provided one is willing to buy the underlying assumption of jet universality, i.e. that the fragmentation process is fundamentally the same whether one is considering an  $e^+e^-$  or a  $pp$  event, and that the only differences are to be found in the parton-level processes involved. This package is not only used by the other ‘Lund’ programs, but also by numerous other programs written to study specific processes. The smaller part of JETSET is a generator for  $e^+e^-$  annihilation events, according to either a parton-shower or a matrix-element approach. The JETSET program is completely self-contained.

PYTHIA is a program made to generate hard or soft processes in collisions between leptons, hadrons and photons, especially at  $e^+e^-$ ,  $ep$  and  $pp$  colliders. Where JETSET is a loose collection of routines that you can combine as desired, PYTHIA is a more structured program, where you initially set up what processes you want to study, and thereafter all events will be generated according to this specification. Included is an extensive library of hard subprocess differential cross-sections, a library of parton distribution functions, a process generation machinery, treatment of initial-state showers and beam remnants, and a few odds and ends. JETSET is used for final-state showers, fragmentation and decay, but no other external libraries are needed.

#### 3.1 Program Philosophy

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

Basically, this means that there are three ways by which you communicate with the programs. First, by setting common block variables, you specify the details of how the programs should perform specific tasks, i.e. which subprocesses should be generated (for PYTHIA), which particle masses should be assumed, which coupling constants used,

which fragmentation scenarios, and so on with hundreds of options and parameters. All of these variables have been assigned sensible default values, so you would only touch those of particular interest to you. Second, by calling subroutines you tell the programs to generate events according to the rules established above. Normally there are few subroutine arguments, and those are usually related to details of the physical situation, such as what c.m. energy to assume for events. Third, you can either look at the common block `LUJETS` to extract information on the generated event, or you can call on various functions and subroutines to analyse the event further for you.

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

On the issue of initialization, `JETSET` and `PYTHIA` behave quite differently. Most `JETSET` routines work without any initialization (except for the one implied by the presence of `BLOCK DATA LUDATA`), i.e. each event and each task stand on their own. Current common block values are used to perform the tasks in specific ways, and those rules can be changed from one event to the next (or even within the generation of one and the same event) without any penalty. In `PYTHIA`, on the other hand, a sizeable amount of initialization is performed in the `PYINIT` call, and thereafter the events generated by `PYEVNT` all obey the rules established at that point. Therefore common block variables that specify methods to be used have to be set before the `PYINIT` call and then not be changed afterwards, with few exceptions.

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

The programs contain a number of checks that requested processes have been implemented, that flavours specified for jet systems make sense, that the energy is sufficient to allow hadronization, that the memory space in `LUJETS` is large enough, etc. If anything goes wrong that the program can catch, an error message will be printed and the treatment of the corresponding event will be cut short. In serious cases, the program will abort. It must be emphasized that not all errors will be caught. In particular, program behaviour is unpredictable if a switch (an integer) is set to an undefined value or a parameter (a real number) is set to a value outside the physically sensible range. Users, beware!

## 3.2 Getting Started with `JETSET`

The most frequently used `JETSET` routine for event generation is `LUEEVT`. A

```
CALL LUEEVT(IFL,ECM)
```

is enough to generate one single event of flavour type `IFL` (with 0 for a mixture of d, u, s, c and b quarks) at a c.m. energy of `ECM`. A corresponding routine `LUONIA` exists for

onium production and decay to ggg or  $\gamma$ gg final states. There are also generic routines to set up user-defined parton configurations and fragment them.

The complete event record, including partons and particles created at various stages of the event evolution, is stored in the common block LUJETS

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

For particle I, K(I,1) gives information on whether or not a jet or particle has fragmented or decayed, K(I,2) gives the particle species code, K(I,3) its origin, K(I,4) and K(I,5) the position of fragmentation/decay products, and P(I,1)–P(I,5) three-momentum, energy and mass. The V matrix contains production/decay vertices. The number of lines in current use is given by N, i.e.  $1 \leq I \leq N$ .

A number of routines are available for manipulating the event record after the event has been generated. Thus CALL LULIST(1) will give a listing of the main information stored in the event record, while CALL LUEDIT(3) will remove everything except stable charged particles. More advanced possibilities include, among others, sphericity and clustering routines.

Apart from the input arguments of subroutine calls, control on the doings of JETSET may be imposed via the LUDAT1, LUDAT2, LUDAT3 and LUDAT4 common blocks. Here sensible default values are always provided. A user might want to switch off all particle decays by putting MSTJ(21)=0 or increase the s/u ratio in fragmentation by putting PARJ(2)=0.40, to give but two examples. It is by exploring the possibilities offered here that JETSET can be turned into an extremely versatile tool, even if all the nice physics is already present in the default values.

### 3.3 Getting Started with PYTHIA

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

The usage of PYTHIA can thus be subdivided into three steps.

1. The initialization step. It is here that all the basic characteristics of the coming generation are specified. The material in this section includes the following.

- Common blocks, at least the following, and maybe some more:

```
COMMON/LUJETS/N,K(4000,5),P(4000,5),V(4000,5)
COMMON/LUDAT1/MSTU(200),PARU(200),MSTJ(200),PARJ(200)
COMMON/PYSUBS/MSEL,MSUB(200),KFIN(2,-40:40),CKIN(200)
COMMON/PYPARS/MSTP(200),PARP(200),MSTI(200),PARI(200)
```

- Selection of required processes. Some fixed ‘menus’ of subprocesses can be selected with different MSEL values, but with MSEL=0 it is possible to compose ‘à la carte’, using the subprocess numbers. To generate processes 14, 18 and 29, for instance, one needs

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

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

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

Unfortunately, initial- and final-state radiation will shift around the kinematics of the hard scattering, making the effects of cuts less predictable. One therefore always has to be very careful that no desired event configurations are cut out.

- Definition of underlying physics scenario, e.g. top mass.
- Selection of parton distribution function sets,  $Q^2$  definitions, and all other details of the generation.
- Switching off of generator parts not needed for toy simulations, e.g. fragmentation for parton level studies.
- Initialization of the event generation procedure. Here kinematics is set up, maxima of differential cross-sections are found for future Monte Carlo generation, and a number of other preparatory tasks carried out. Initialization is performed by PYINIT, which should be called only after the switches and parameters above have been set to their desired values. The frame, the beam particles and the energy have to be specified:

```
CALL PYINIT('CMS', 'p', 'pbar', 1800.)
```

- Any other initial material required by the user, e.g. histogram booking.
2. The generation loop. It is here that events are generated and studied. It includes the following tasks:
    - Generation of the next event, with

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

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

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

## 4 Summary and Outlook

We have here given a very brief introduction to the PYTHIA 5.7 and JETSET 7.4 programs. A more detailed description of physics and programs is available separately [3]. Any serious user should turn to this publication, and to the original physics papers, for further information. Here follows a list of some of the main areas where the PYTHIA/JETSET programs contain original research:

- The string fragmentation model [8].
- The string effect [10].
- Baryon production (diquark/popcorn) [11].
- Fragmentation of multiparton systems [9].
- Fragmentation effects on  $\alpha_s$  determinations [12].

- Initial-state parton showers [6].
- Final-state parton showers [5].
- Photon radiation from quarks [13].
- Deep inelastic scattering [14].
- Photoproduction [15].
- Colour flow in hard scatterings [1].
- Elastic and diffractive cross sections [16].
- Minijets (multiple parton–parton interactions) [7].
- Rapidity gaps [17].
- Jet clustering in  $k_{\perp}$  [2].

The current versions represent the latest in a long row, which are already widely used by the high-energy physics experimental community at (past, present and future) machines such as PETRA, PEP, TRISTAN, DORIS, CESR, LEP, SLC, SPS, Tevatron, HERA, LHC (SSC) and NLC.

The PYTHIA/JETSET programs are continuously being developed. We are aware of many physics shortcomings, which hopefully will be addressed in the future. It is in the nature of a program of this kind never to be finished, at least as long as it is of importance for the high-energy physics experimental community.

Apart from the physics aspects, one may also worry about the programming ones. For instance, for historical reasons, single-precision real numbers are used almost everywhere. With the push to higher energies, this is becoming more and more of a problem, so it would be logical to move to double precision throughout. Another possible change would be an introduction of Fortran 90 programming elements. The JETSET and PYTHIA programs are becoming so intertwined, these days, that it would make sense to join them into one single program. In the process, one would probably also remove a number of options that are no longer used.

No timetable is set up for future changes. After all, this is not a professionally maintained software product, but part of a one-man physics research project. In particular, since no single unifying model exists which ‘does it all’, improved physics understanding and modelling goes hand in hand with program development.

## Acknowledgements

A large number of persons should be thanked for their contributions. Hans-Uno Bengtsson is the originator of the PYTHIA program, and for many years we worked in parallel on its further development. Mats Bengtsson is the main author of the final-state parton-shower algorithm. A few pieces of code have been donated by other persons. Bo Andersson and Gösta Gustafson are the originators of the Lund model, and strongly influenced the early development of the programs. Further comments on the programs have been obtained from users too numerous to be mentioned here, but who are all gratefully acknowledged. To write programs of this size and complexity would be impossible without a strong user feedback. However, the moral responsibility for any remaining errors clearly rests with me.

## References

- [1] H.-U. Bengtsson, Computer Physics Commun. **31** (1984) 323;  
H.-U. Bengtsson and G. Ingelman, Computer Physics Commun. **34** (1985) 251;  
H.-U. Bengtsson and T. Sjöstrand, Computer Physics Commun. **46** (1987) 43
- [2] T. Sjöstrand, Computer Physics Commun. **27** (1982) 243;  
T. Sjöstrand, Computer Physics Commun. **28** (1983) 227;  
T. Sjöstrand, Computer Physics Commun. **39** (1986) 347;  
T. Sjöstrand and M. Bengtsson, Computer Physics Commun. **43** (1987) 367
- [3] T. Sjöstrand, CERN-TH.7112/93
- [4] H. Plochow-Besch, Computer Physics Commun. **75** (1993) 396
- [5] M. Bengtsson and T. Sjöstrand, Nucl. Phys. **B289** (1987) 810
- [6] T. Sjöstrand, Phys. Lett. **157B** (1985) 321;  
M. Bengtsson, T. Sjöstrand and M. van Zijl, Z. Phys. **C32** (1986) 67
- [7] T. Sjöstrand and M. van Zijl, Phys. Rev. **D36** (1987) 2019
- [8] B. Andersson, G. Gustafson, G. Ingelman and T. Sjöstrand, Phys. Rep. **97** (1983) 31
- [9] T. Sjöstrand, Nucl. Phys. **B248** (1984) 469
- [10] B. Andersson, G. Gustafson and T. Sjöstrand, Phys. Lett. **B94** (1980) 211
- [11] B. Andersson, G. Gustafson and T. Sjöstrand, Nucl. Phys. **B197** (1982) 45; Physica Scripta **32** (1985) 574
- [12] T. Sjöstrand, Z. Phys. **C26** (1984) 93;  
M. Bengtsson, T. Sjöstrand and M. van Zijl, Phys. Lett. **B179** (1986) 164
- [13] T. Sjöstrand, in ‘Workshop on Photon Radiation from Quarks’, ed. S. Cartwright, CERN 92-04 (Geneva, 1992), p. 89 and p. 103
- [14] B. Andersson, G. Gustafson, G. Ingelman and T. Sjöstrand, Z. Phys. **C9** (1981) 233;  
M. Bengtsson and T. Sjöstrand, Z. Phys. **C37** (1988) 465
- [15] G.A. Schuler and T. Sjöstrand, Nucl. Phys. **B407** (1993) 539
- [16] G.A. Schuler and T. Sjöstrand, CERN-TH.6837/93, to appear in Phys. Rev. **D**
- [17] Yu.L. Dokshitzer, V.A. Khoze and T. Sjöstrand, Phys. Lett. **B274** (1992) 116