Pythia at Hadron Colliders

Torbjörn Sjöstrand

Theory Division, CERN
CH–1211 Geneva 23
Switzerland

Basic Facts:

Program names: Pythia and Jetset [1, 2, 3]
Versions: Pythia 5.6, from April 1993, and
Jetset 7.3, from November 1992
Author: Torbjörn Sjöstrand
Theory Division, CERN
CH–1211 Geneva 23, Switzerland
Phone + 41 - 22 - 767 28 20
E-mail TORSJO @ CERNVM.CERN.CH
Program size: 17173 + 10519 lines
Manual: a complete physics description and manual is available [3]

---

1 Introduction

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 particles produced in the interactions between two incoming particles. This in particular means multiparticle production at e^+e^-, pp and ep colliders, although also other applications are envisaged. Ideally the 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. Many of the components of the programs represent original research, in the sense that physics models have been developed specifically for these programs. 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. However, the current report emphasizes pp physics, i.e. e^+e^- and ep applications are usually not described.

Both programs have a long history, and several manuals have been published. 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 detailed in the full manual and physics description [3], to which interested users are asked to turn for details.

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 structure functions, which defines the partonic substructure in terms of flavour composition and energy sharing.
2. One shower initiator parton from each beam starts off a sequence of branchings, such as $q \to qg$, which build up an initial-state shower.
3. One incoming parton from each of the two showers enters the hard process, where then a number of outgoing partons are produced, usually two. It is the nature of this process that determines the main characteristics of the event.
4. Also the outgoing partons may branch, to build up final-state showers.
5. 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.
6. The QCD confinement mechanism ensures that the outgoing quarks and gluons are not observable, but instead fragment to colour neutral hadrons.
7. Many of the produced hadrons are unstable and decay further.

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

In the following subsections, we will survey the above seven 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 Structure Functions

Pythia contains a rich physics 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 '$2 \to 1$' processes, '$2 \to 2$' ones, '$2 \to 3$' ones, etc. This aspect is very relevant from a programming point of view: the more particles in the final state, the more complicated the phase space and therefore the whole generation procedure. In fact, Pythia is optimized for $2 \to 1$ and $2 \to 2$ processes. There is currently no generic treatment of processes with three or more particles in the final state, but rather a few different machineries, each tailored to the pole structure of a specific class of graphs. This may be seen as a major limitation, and indeed is so at times. However, often one can come quite far with only one or two particles in the final state, since showers will add the required extra activity. The classification may also be misleading at times, since an s-channel resonance is considered as a single particle, even if it is assumed always to decay into two final-state particles. Thus the process $e^+e^- \to W^+W^- \to q_1\overline{q}'_1\, q_2\overline{q}'_2$ is classified as $2 \to 2$, although the decay treatment of the W pair includes the full $2 \to 4$ matrix elements.

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
1	$\mathrm{f}_i\overline{\mathrm{f}}_i \to \gamma^*/\mathrm{Z}^0$
2	$\mathrm{f}_i\overline{\mathrm{f}}_j \to \mathrm{W}^+$
3	$\mathrm{f}_i\overline{\mathrm{f}}_i \to \mathrm{H}^0$
5	$\mathrm{Z}^0\mathrm{Z}^0 \to \mathrm{H}^0$
8	$\mathrm{W}^+\mathrm{W}^- \to \mathrm{H}^0$
10	$\mathrm{f}_i\mathrm{f}_j \to \mathrm{f}_i\mathrm{f}_j$ (QFD)
11	$\mathrm{f}_i\mathrm{f}_j \to \mathrm{f}_i\mathrm{f}_j$ (QCD)
12	$\mathrm{f}_i\overline{\mathrm{f}}_i \to \mathrm{f}_k\overline{\mathrm{f}}_k$
13	$\mathrm{f}_i\overline{\mathrm{f}}_i \to \mathrm{gg}$
14	$\mathrm{f}_i\overline{\mathrm{f}}_i \to \mathrm{g}\gamma$
15	$\mathrm{f}_i\overline{\mathrm{f}}_i \to \mathrm{gZ}^0$
16	$\mathrm{f}_i\overline{\mathrm{f}}_j \to \mathrm{gW}^+$
18	$\mathrm{f}_i\overline{\mathrm{f}}_i \to \gamma\gamma$
19	$\mathrm{f}_i\overline{\mathrm{f}}_i \to \gamma\mathrm{Z}^0$
20	$\mathrm{f}_i\overline{\mathrm{f}}_j \to \gamma\mathrm{W}^+$
22	$\mathrm{f}_i\overline{\mathrm{f}}_i \to \mathrm{Z}^0\mathrm{Z}^0$
23	$\mathrm{f}_i\overline{\mathrm{f}}_j \to \mathrm{Z}^0\mathrm{W}^+$
24	$\mathrm{f}_i\overline{\mathrm{f}}_i \to \mathrm{Z}^0\mathrm{H}^0$
25	$\mathrm{f}_i\overline{\mathrm{f}}_i \to \mathrm{W}^+\mathrm{W}^-$
26	$\mathrm{f}_i\overline{\mathrm{f}}_j \to \mathrm{W}^+\mathrm{H}^0$
28	$\mathrm{f}_i\mathrm{g} \to \mathrm{f}_i\mathrm{g}$
29	$\mathrm{f}_i\mathrm{g} \to \mathrm{f}_i\gamma$
30	$\mathrm{f}_i\mathrm{g} \to \mathrm{f}_i\mathrm{Z}^0$
31	$\mathrm{f}_i\mathrm{g} \to \mathrm{f}_k\mathrm{W}^+$
33	$\mathrm{f}_i\gamma \to \mathrm{f}_i\mathrm{g}$
34	$\mathrm{f}_i\gamma \to \mathrm{f}_i\gamma$
35	$\mathrm{f}_i\gamma \to \mathrm{f}_i\mathrm{Z}^0$
36	$\mathrm{f}_i\gamma \to \mathrm{f}_k\mathrm{W}^+$
53	$\mathrm{gg} \to \mathrm{f}_k\overline{\mathrm{f}}_k$
54	$\mathrm{g}\gamma \to \mathrm{f}_k\overline{\mathrm{f}}_k$
58	$\gamma\gamma \to \mathrm{f}_k\overline{\mathrm{f}}_k$
68	$\mathrm{gg} \to \mathrm{gg}$

no.	subprocess
69	$\gamma\gamma \to \mathrm{W}^+\mathrm{W}^-$
70	$\gamma\mathrm{W}^+ \to \mathrm{Z}^0\mathrm{W}^+$
71	$\mathrm{Z}^0_L\mathrm{Z}^0_L \to \mathrm{Z}^0_L\mathrm{Z}^0_L$
72	$\mathrm{Z}^0_L\mathrm{Z}^0_L \to \mathrm{W}^+_L\mathrm{W}^-_L$
73	$\mathrm{Z}^0_L\mathrm{W}^+_L \to \mathrm{Z}^0_L\mathrm{W}^+_L$
76	$\mathrm{W}^+_L\mathrm{W}^-_L \to \mathrm{Z}^0_L\mathrm{Z}^0_L$
77	$\mathrm{W}^+_L\mathrm{W}^\pm_L \to \mathrm{W}^+_L\mathrm{W}^\pm_L$
81	$\mathrm{f}_i\overline{\mathrm{f}}_i \to \mathrm{Q}_k\overline{\mathrm{Q}}_k$
82	$\mathrm{gg} \to \mathrm{Q}_k\overline{\mathrm{Q}}_k$
83	$\mathrm{q}_i\mathrm{f}_j \to \mathrm{Q}_k\mathrm{f}_l$
84	$\mathrm{g}\gamma \to \mathrm{Q}_k\overline{\mathrm{Q}}_k$
85	$\gamma\gamma \to \mathrm{F}_k\overline{\mathrm{F}}_k$
86	$\mathrm{gg} \to \mathrm{J}/\psi\mathrm{g}$
87	$\mathrm{gg} \to \chi_{0c}\mathrm{g}$
88	$\mathrm{gg} \to \chi_{1c}\mathrm{g}$
89	$\mathrm{gg} \to \chi_{2c}\mathrm{g}$
91	elastic scattering
92	single diffraction
93	double diffraction
95	low-$p_\perp$ production
102	$\mathrm{gg} \to \mathrm{H}^0$
103	$\gamma\gamma \to \mathrm{H}^0$
111	$\mathrm{f}_i\overline{\mathrm{f}}_i \to \mathrm{gH}^0$
112	$\mathrm{f}_i\mathrm{g} \to \mathrm{f}_i\mathrm{H}^0$
113	$\mathrm{gg} \to \mathrm{gH}^0$
114	$\mathrm{gg} \to \gamma\gamma$
115	$\mathrm{gg} \to \mathrm{g}\gamma$
121	$\mathrm{gg} \to \mathrm{Q}_k\overline{\mathrm{Q}}_k\mathrm{H}^0$
122	$\mathrm{q}_i\overline{\mathrm{q}}_i \to \mathrm{Q}_k\overline{\mathrm{Q}}_k\mathrm{H}^0$
123	$\mathrm{f}_i\mathrm{f}_j \to \mathrm{f}_i\mathrm{f}_j\mathrm{H}^0$
124	$\mathrm{f}_i\mathrm{f}_j \to \mathrm{f}_k\mathrm{f}_l\mathrm{H}^0$
131	$\mathrm{gg} \to \mathrm{Z}^0\mathrm{Q}_k\overline{\mathrm{Q}}_k$

no.	subprocess
141	$\mathrm{f}_i\overline{\mathrm{f}}_i \to \gamma/\mathrm{Z}^0/\mathrm{Z}'^0$
142	$\mathrm{f}_i\overline{\mathrm{f}}_j \to \mathrm{W}'^+$
143	$\mathrm{f}_i\overline{\mathrm{f}}_j \to \mathrm{H}^+$
144	$\mathrm{f}_i\overline{\mathrm{f}}_j \to \mathrm{R}$
145	$\mathrm{q}_i\ell_j \to \mathrm{L}_\mathrm{Q}$
147	$\mathrm{dg} \to \mathrm{d}^*$
148	$\mathrm{ug} \to \mathrm{u}^*$
149	$\mathrm{gg} \to \eta_{techni}$
151	$\mathrm{f}_i\overline{\mathrm{f}}_i \to \mathrm{H}'^0$
152	$\mathrm{gg} \to \mathrm{H}'^0$
153	$\gamma\gamma \to \mathrm{H}'^0$
156	$\mathrm{f}_i\overline{\mathrm{f}}_i \to \mathrm{A}^0$
157	$\mathrm{gg} \to \mathrm{A}^0$
158	$\gamma\gamma \to \mathrm{A}^0$
161	$\mathrm{f}_i\mathrm{g} \to \mathrm{f}_k\mathrm{H}^+$
162	$\mathrm{qg} \to \ell\mathrm{L}_\mathrm{Q}$
163	$\mathrm{gg} \to \mathrm{L}_\mathrm{Q}\overline{\mathrm{L}}_\mathrm{Q}$
164	$\mathrm{q}_i\overline{\mathrm{q}}_i \to \mathrm{L}_\mathrm{Q}\overline{\mathrm{L}}_\mathrm{Q}$
165	$\mathrm{f}_i\overline{\mathrm{f}}_i(\to \gamma^*/\mathrm{Z}^0) \to \mathrm{f}_k\overline{\mathrm{f}}_k$
166	$\mathrm{f}_i\overline{\mathrm{f}}_j(\to \mathrm{W}^\pm) \to \mathrm{f}_k\overline{\mathrm{f}}_l$
171	$\mathrm{f}_i\overline{\mathrm{f}}_i \to \mathrm{Z}^0\mathrm{H}'^0$
172	$\mathrm{f}_i\overline{\mathrm{f}}_j \to \mathrm{W}^+\mathrm{H}'^0$
173	$\mathrm{f}_i\mathrm{f}_j \to \mathrm{f}_i\mathrm{f}_j\mathrm{H}'^0$
174	$\mathrm{f}_i\mathrm{f}_j \to \mathrm{f}_k\mathrm{f}_l\mathrm{H}'^0$
176	$\mathrm{f}_i\overline{\mathrm{f}}_i \to \mathrm{Z}^0\mathrm{A}^0$
177	$\mathrm{f}_i\overline{\mathrm{f}}_j \to \mathrm{W}^+\mathrm{A}^0$
178	$\mathrm{f}_i\mathrm{f}_j \to \mathrm{f}_i\mathrm{f}_j\mathrm{A}^0$
179	$\mathrm{f}_i\mathrm{f}_j \to \mathrm{f}_k\mathrm{f}_l\mathrm{A}^0$
181	$\mathrm{gg} \to \mathrm{Q}_k\overline{\mathrm{Q}}_k\mathrm{H}'^0$
182	$\mathrm{q}_i\overline{\mathrm{q}}_i \to \mathrm{Q}_k\overline{\mathrm{Q}}_k\mathrm{H}'^0$
186	$\mathrm{gg} \to \mathrm{Q}_k\overline{\mathrm{Q}}_k\mathrm{A}^0$
187	$\mathrm{q}_i\overline{\mathrm{q}}_i \to \mathrm{Q}_k\overline{\mathrm{Q}}_k\mathrm{A}^0$

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

- Hard QCD processes, e.g. $\mathrm{qg} \to \mathrm{qg}$.
- Soft QCD processes, such as diffractive and elastic scattering, and minimum-bias events.
- Heavy-flavour production, e.g. $\mathrm{gg} \to \mathrm{t}\overline{\mathrm{t}}$.
- Prompt-photon production, e.g. $\mathrm{qg} \to \mathrm{q}\gamma$.
- Photon-induced processes, e.g. $\gamma\mathrm{g} \to \mathrm{q}\overline{\mathrm{q}}$.
- Deep inelastic scattering, e.g. $\mathrm{q}\ell \to \mathrm{q}\ell$.

- W/Z production, such as the $e^+e^- \rightarrow \gamma^*/Z^0$ (also found in JETSET), or $q\overline{q} \rightarrow W^+W^-$.
- 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 $W_LW_L \rightarrow W_LW_L$ (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. $gg \rightarrow \eta_{techni}$.
- 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. These scenarios do not always appear as separate processes, but may just be options to some of the processes above.

This 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.

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

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

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

Structure 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 hadron structure functions 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 structure functions. Some fits are available in PYTHIA, while more can be accessed by an interface to PDFLIB [4].

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

2.2 Initial- and Final-State Radiation

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

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

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 (i.e. the 2/3/4/5-jet composition). 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 α_s determinations.

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

PYTHIA does not contain any full higher-order matrix elements, with loop contributions included (while JETSET does, for $e^+e^- \to \gamma^*/Z^0 \to q\overline{q}$). There are a few cases where higher-order matrix elements are included at the Born level, such as the two first-order processes $qg \to Wq'$ and $q\overline{q}' \to Wg$. The cross-sections for these processes are divergent when the $p_\perp \to 0$. 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, $q\overline{q}' \to W$, 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 $e^+e^- \to Z^0 \to q\overline{q}$ only contains final-state QCD radiation, while $q\overline{q} \to Z^0 \to e^+e^-$ only contains initial-state QCD one. Similarly, the distinction of emission as coming either from the q or from the $\overline{q}$ is arbitrary. In general, the assignment of radiation to a given mother parton is a good approximation for an emission close to the direction of motion of that parton, but not for the wide-angle emission in between two jets, where interference terms are expected to be important.

In both initial- and final-state showers, the structure is given in terms of branchings $a \to bc$, specifically $e \to e\gamma$, $q \to qg$, $q \to q\gamma$, $g \to gg$, and $g \to q\overline{q}$. Each of these processes is characterized by a splitting kernel $P_{a\to bc}(z)$. The branching rate is proportional to the integral $\int P_{a\to bc}(z)\, dz$. The z value picked for a branching describes the energy sharing, with daughter b taking a fraction z and daughter c the remaining $1 - z$ of the 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 spacelike ($m^2 < 0$) and gradually increasing as the hard scattering is approached, while Q^2 is timelike ($m^2 > 0$) 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

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 which takes the rest. For a proton beam, a u quark initiator would leave behind a ud diquark beam remnant, with an antitriplet colour charge. The remnant is therefore colour-connected to the hard interaction, and forms part of the same fragmenting system. It is further customary to assign a primordial transverse momentum to the shower initiator, to take into account the motion of quarks inside the original hadron, 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, which are not understood from first principles.

So far we have assumed that each event only contains one hard interaction, i.e. that each incoming particle has only one parton which takes part in hard processes, and that all other constituents sail through unaffected. This is appropriate in e^+e^- or ep events, but not necessarily so in hadron–hadron collisions. Here each of the beam particles contains a multitude of partons, and so the probability for several interactions in one and the same event need not be negligible. In principle these additional interactions could arise because one single parton from one beam scatters against several different partons from the other beam, or because several partons from each beam take place in separate $2 \to 2$ scatterings. Both are expected, but combinatorics should favour the latter, which is the mechanism considered in PYTHIA [7].

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

In a typical 'minimum-bias' event one therefore expects to find one or a few scatterings at scales around or a bit above $p_{\perp min}$, while a high-$p_\perp$ event also may have additional scatterings at the $p_{\perp min}$ 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 an increased probability or not for additional interactions in an event which is known to contain a hard scattering, compared with one that contains no hard interactions?

2.4 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 regime, 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. Three main schools are usually distinguished, string fragmentation (SF), independent fragmentation (IF) and cluster fragmentation (CF), but many variants and hybrids exist. Being models, none of them can lay claims to being 'correct', although some may be better founded than others. The best that can be aimed for is internal consistency, a good representation of existing data, and a predictive power for properties not yet studied or results at higher energies.

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 $q\overline{q}$ 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 $\overline{q}$ partons move apart from their common production vertex, the physical picture is that of a colour flux tube (or maybe colour vortex line) being stretched between the q and the $\overline{\mathrm{q}}$. The transverse dimensions of the tube are of typical hadronic sizes, roughly 1 fm. If the tube is assumed to be uniform along its length, this automatically leads to a confinement picture with a linearly rising potential. In order to obtain a Lorentz covariant and causal description of the energy flow due to this linear confinement, the most straightforward way is to use the dynamics of the massless relativistic string with no transverse degrees of freedom. The mathematical, one-dimensional string can be thought of as 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 $\kappa \approx 1$ GeV/fm. The expression 'massless' relativistic string is somewhat of a misnomer: κ effectively corresponds to a 'mass density' along the string.

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

In order to generate the quark–antiquark pairs $\mathrm{q}'\overline{\mathrm{q}}'$ which lead to string break-ups, the Lund model invokes the idea of quantum mechanical tunnelling. This leads to a flavour-independent Gaussian spectrum for the $p_\perp$ of $\mathrm{q}'\overline{\mathrm{q}}'$ 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, $\mathrm{u} : \mathrm{d} : \mathrm{s} : \mathrm{c} \approx 1 : 1 : 0.3 : 10^{-11}$. Charm and heavier quarks hence are not expected to be produced in the soft fragmentation, but only in perturbative parton-shower branchings $\mathrm{g} \to \mathrm{q}\overline{\mathrm{q}}$.

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

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

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

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

If several partons are moving apart from a common origin, the details of the string drawing become more complicated. For a $\mathrm{q\overline{q}g}$ event, a string is stretched from the q end via the g to the $\overline{\mathrm{q}}$ end, i.e. the gluon is a kink on the string, carrying energy and momentum. As a consequence, the gluon has two string pieces attached, and the ratio of gluon to quark string force is 2, a number which can be compared with the ratio of colour charge Casimir operators, $N_C/C_F = 2/(1 - 1/N_C^2) = 9/4$. In this, as in other respects, the string model can be viewed as a variant of QCD where the number of colours N_C is not 3 but infinite.

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

For events that involve many partons, there may be several possible topologies for their ordering along the string. An example would be a $\mathrm{q\overline{q}g_1g_2}$ (the gluon indices are here used to label two different gluon-momentum vectors), where the string can connect the partons in either of the sequences $\mathrm{q - g_1 - g_2 - \overline{q}}$ and $\mathrm{q - g_2 - g_1 - \overline{q}}$. The matrix elements that are calculable in perturbation theory contain interference terms between these two possibilities, which means that the colour flow is not always well-defined. Fortunately, the interference terms are down in magnitude by a factor $1/N_C^2$, 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 mesons, it is necessary to put together measured exclusive branching ratios with some inclusive multiplicity distributions to obtain a consistent and reasonably complete set of decay channels, a rather delicate task. For bottom, so far only a rather simple phase-space type of generator has been used for hadronic decays.

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

The normal 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 particles: Z^0, $W^\pm$, H^0, Z'^0, $W'^\pm$, H'^0, A^0, $H^\pm$, R^0, q^*, ℓ^*, and L_Q. Together we call these resonances, and contrast the 'particle decay' treatment of JETSET with the 'resonance decay' one of PYTHIA. 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. $W \to q\overline{q}'$ or $H^0 \to W^+W^-$. 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 is 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 mesons decay weakly. With current 'best bet' mass values, this is not correct, but one should rather consider top decay before fragmentation. An option which does this has therefore been introduced, and will become the default in the future.

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 utitilies. 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 selfcontained.

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 structure 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. An interface to external structure-function libraries is provided, however.

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.

Apart from writing a header, giving a brief initialization information, printing error messages if need be, and responding to explicit requests for listings, all tasks of the programs are performed 'silently'.

Unless explicitly stated (or obvious from the context) all switches and parameters can be changed independently of each other. One should note, however, that if only a few switches/parameters are changed, this may result in an artificially bad agreement with data. Many disagreements can often be cured by a subsequent retuning of some other parameters of the model, in particular those that were once determined by a comparison with data in the context of the default scenario. For example, for e^+e^- annihilation, such a retuning could involve one QCD parameter (α_s or Λ), the longitudinal fragmentation function, and the average transverse 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 γgg 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 decay vertices. The number of lines in current use is given by N, i.e. $1 \leq \mathrm{I} \leq \mathrm{N}$.

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 things like sphericity or 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 the generation before events can be generated, and in that it is not possible to change switches and parameters freely during the course of the run. A fairly precise recipe for how a run should be structured can therefore be given.

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

1. The initialization step. It is here that all the basic characteristics of the coming generation are specified. The material in this section includes the following.
   - 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 GeV $\leq p_{\perp} \leq$ 10 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 structure 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.

3.4 Program Distribution

The Pythia and Jetset programs are distributed as part of the CERN computer software library. Read the CERN Computer Newsletter for announcements of new versions. Copies of the programs are also available via anonymous ftp, e.g. from the asis01 server at CERN. Here also the complete physics description and manual is available as a ready-to-be-printed postscript file. Hardcopies are available on request from the CERN program library office, but can also be picked up at the self-service handout areas of CERN and a number of other major laboratories: DESY, FNAL, SSCL, and so on. For any comments on the official distribution, or for requests for hardcopies of the manual, contact CERNLIB@CERNVM.CERN.CH.

The official distributions usually lag behind the current versions. The latter may be picked up from the 'master copies' found on the `TORSJO 192` disk on `CERNVM`. There you have:

`JETSET73 FORTRAN`	the Jetset code,
`PYTHIA56 FORTRAN`	the Pythia code,
`PYTHIA56 TEX`	the common Pythia/Jetset manual, and
`UPDATE MANUAL`	latest update news.

The versions found here are updated without advance warning, as bugs are corrected or new features added. In addition to the source files, one may also find compiled `TEXT` files, which are ready to be linked if you run jobs on the `CERNVM` machine.

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 separately available [3]. Any serious user should turn to this publication, and to the original physics papers, for further information.

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 that it will never be finished, at least so 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 is 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. 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.6488/92

[4] H. Plothow-Besch, CERN-PPE/92-123 (1992)

[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;
T. Sjöstrand, Nucl. Phys. **B248** (1984) 469