LU TP 84-3

Ref.TH.3820-CERN

February 1984

The Lund Monte Carlo for High- p_T Physics

H.-U.Bengtsson

Department of Theoretical Physics

University of Lund

Sölvegatan 14 A

S-223 62 Lund, Sweden*

G.Ingelman

Theory Division, CERN

CH-1211 Geneva 23, Switzerland

ABSTRACT: We present a Monte Carlo program for high- p_T jet physics and prompt photon production in hard hadron-hadron collisions, based on the Lund string model for jet fragmentation. The theoretical background is treated in detail; in particular the colour field topology and the resulting string structure are discussed. A prominent part of the paper is dedicated to a manual on how to use the FORTRAN 77 program, PYTHIA version 3.3, successor to and synthesis of earlier high- p_T programs.

* Present address: UCLA, Los Angeles, USA

1. Introduction

For several years there has been a great interest in hard hadronic collisions, despite their complicated structure. Experiments made at accelerators working at ever increasing energies (SPS, FNAL, ISR) have provided us with a continuous outpour of data demanding theoretical explanation. With the CERN $\bar{p}p$ -collider a new, exciting energy domain has recently been opened.

On the theoretical side there is a general consensus that reactions with a large momentum transfer, Q^2 , can be understood as hard scattering between essentially free partons within the colliding hadrons. This partonic interaction, which can be calculated using perturbative QCD, occurs on a very small space-time scale of less than one fermi. In the following, much larger space-time interval, the outgoing coloured partons begin to feel the non-perturbative confining force, connecting them via colour force fields which then fragment into the observable particles. In order to compare the predictions from perturbative QCD with experimental data it is necessary to take this soft hadronization into account in a consistent way. In fact, it has turned out that some of the characteristics of the reaction on the parton level are distorted (or, in extreme cases, completely washed out) by the hadronization process.

Until a reliable hadronization theory is found one has to use more or less elaborate models, and we will in the following use the Lund model [1]. The general assumption here is that when the partons move apart they stretch the connecting force fields, and it is these fields that eventually break up to produce the final state hadrons. The confining force limits the transverse dimensions of the colour field, compressing it to an essentially one-dimensional structure: a colour flux tube or a vortex line. The dynamics of this system are then modelled by those of the massless relativistic string. The model thus provides a relativistically covariant and causal way of partitioning the energy and momenta of a colour force field

among the final state hadrons.

Obviously, the particle distributions will depend on the exact topology of the force fields, different field configurations leading to different distributions. As simple an example as that of fig.1, with two colour triplets and two colour antitriplets, already admits of two distinct possibilities for the colour field; introducing also colour octets (gluons or hadrons having emitted a gluon) quickly multiplies the ambiguities. A knowledge of the cross-section for a particular Feynman diagram will then not suffice to specify unequivocally the final state. Rather, one has to start from the field topology and calculate the cross-section belonging to a definite formation. It has been shown in [2] how a given string configuration can be associated with a specific colour amplitude in a gauge invariant way. Simply stated, each 'switch', at which a colour α arrives and is 'shunted' on as β , corresponds to a factor $T_{\alpha\beta}^a$ (cf. fig.3), where $T^a \equiv \lambda^a/2$, with λ^a the ordinary SU(3) matrices. This connection then suggests the following scheme for calculating the desired cross-sections:

A.1. Specify a subprocess and draw the Feynman diagrams in a given order of α_s .

A.2. Write down the total amplitudes according to the Feynman rules.

B.1. Find the different final state string configurations for the subprocess.

B.2. Calculate the colour parts of the amplitudes for the final states according to [2].

C.1. Compare with the amplitudes from step A.2 to find the total amplitude for the different final state configurations.

C.2. Use these amplitudes to compute the corresponding cross-sections.

Up to this point our discussion has been perfectly general, the scheme being applicable to any model for colour fields; we now specialize to the case of the Lund model. We are not arguing that this is the way of treating the colour force field, but adapt the model rather for the sake of simplicity. More complex models can be conceived of, but using this extreme (in the sense of the next paragraph) model to calculate measurable quantities, we can then compare with experiments to look for discrepancies, which will tell us something about the nature of the confining force field. In order to do this, it is essential that the Monte Carlo program is able to simulate the complete final state of particles, e.g. also those originating from the hadron spectator remnants. In fact, since in the string model only colour singlet systems are recognized, this is automatically included.

The main feature of the Lund model [1], with respect to the colour field structure, is that it recognizes only triplet strings. Colour octets (i.e. gluons and hadrons having emitted a gluon) are treated as kinklike excitations of the string: the force field is always stretched from colour triplet (quark or antiquark) to colour antitriplet (antiquark or diquark) via the colour octets. This type of string model is supported by experiments: as pointed out in [3], it implies correlations between multiplicity and angular distributions for hadrons produced in $q\bar{q}g$ -events - correlations that have actually been observed by the JADE collaboration at PETRA [4]. The Monte Carlo implementation of the Lund model for soft hadronization is expounded in [5].

The purpose of this paper, which is a further development of an earlier work [6], is to present the Lund Monte Carlo program for high- p_T physics, PYTHIA version 3.3, and elaborate on its physical motivation, giving a complete description of all $\alpha_s \cdot \alpha_{em}$ - and α_s^2 -processes in terms of the string model. In Section 2 we show an example of how the given scheme works for α_s^2 -processes (outlining the calculations for $gg \rightarrow gg$ in Appendix 2). In Section 3 we deal somewhat with the remaining problems for a Monte Carlo implementation of high- p_T physics and in Section 4 touch briefly on the issues of a continuous transition between high- p_T and low- p_T and the question of K-factors in high- p_T . Section 5 contains an exhaustive description of the program and Section 6 some examples on how to use it. The cross-sections for the different final state string configurations are presented in Table 1 at the end.

2. Colour Flows, String Configurations and Cross-sections

When we turn our attention to the allowed reactions of order α_s^2 , and $\alpha_s \cdot \alpha_{em}$ we are faced with a somewhat complex situation. There are in all ten different possible subprocesses, and each of these will in general give rise to several distinct final state string configurations; our task is to enumerate these and calculate the corresponding cross-sections. We set about doing this by following the recipe given in the Introduction. We take a specific reaction, draw the Feynman diagrams contributing to this process and write down the amplitudes for each diagram according to the Feynman rules. Rewriting the colour part of the amplitudes to tally with the form of the colour factors given in [2], we then collect all non-colour factors belonging to a particular colour amplitude: this will yield the spin and kinematics part of the amplitude belonging to that colour amplitude, i.e. that string configuration. Summing and averaging over spins for this part, and separately over colours in the colour part of the amplitude, then gives the desired cross-section.

The easiest way to acquaint oneself with the method employed is to study a specific example, and to this end we choose here to treat the case of $q_i g \rightarrow q_i g$. There are three Feynman diagrams contributing to this process (figs.2a-c, corresponding to the s-, t- and u-channel, respectively). In order conveniently to discuss the situation, we number the partons 1 through 4 and distribute colours as in fig.2. The amplitudes associated with each diagram can now easily be obtained from the Feynman rules, viz.:

$$s: -ig^2 T_{\alpha\gamma}^a T_{\gamma\beta}^b \bar{u}_i^\beta(q_4) \not{\epsilon}_3 \frac{\not{\epsilon}_1 + \not{\epsilon}_2}{\hat{s}} \not{\epsilon}_2 u_i^\alpha(q_1)$$

$$t: g^2 f^{abc} T_{\alpha\beta}^c \bar{u}_i^\beta(q_4) \frac{\epsilon_2^\kappa \epsilon_3^\lambda \gamma^\mu}{\hat{t}} C_{\kappa\lambda\mu}(-q_2, q_3, q_2 - q_3) u_i^\alpha(q_1)$$

$$u: -ig^2 T_{\alpha\gamma}^b T_{\gamma\beta}^a \bar{u}_i^\beta(q_4) \not{\epsilon}_2 \frac{\not{\epsilon}_1 - \not{\epsilon}_3}{\hat{u}} \not{\epsilon}_3 u_i^\alpha(q_1)$$

(See Appendix 2 for a definition of $C_{\kappa\lambda\mu}$.)

The possible final string configurations are just as simply found. The gluon, 2, was originally contained in the hadron incident from the right, the remnant of which, $\bar{2}$, now continues as a colour octet with the anticolours of 2. Likewise, the quark, 1, stems from the hadron incident from the left, and the remnant of this, called $\bar{1}$, continues as a colour antitriplet, with the anticolour of quark 1. There now exist two possible colour flows, A and B: either colour flows from 1 to 2 to 3 to 4 (A), or from 1 to 3 to 2 to 4 (B) (see figs.3a-b). The two possibilities correspond to the two final string configurations in figs.4a-b, where the colour force field is stretched either from the resulting quark, 4, to gluon 3 to hadron remnant $\bar{2}$ to hadron remnant $\bar{1}$ (A), or from 4 to $\bar{2}$ to 3 to $\bar{1}$ (B). The colour amplitudes A and B are easily read off from [2]; A represents the colour factor $T_{\alpha\gamma}^a T_{\gamma\beta}^b$ and B the colour factor $T_{\alpha\gamma}^b T_{\gamma\beta}^a$. By inspection, the s-channel amplitude is immediately seen to contain colour factor A, while the u-channel amplitude contains factor B; using the identity $f^{abc} T_{\alpha\beta}^c$

$= -i[T^a, T^b]_{\alpha\beta}$, the t-channel amplitude is found to contain both of them. Introducing the symbols M_s , M_t and M_u to denote those parts of the s-, t- and u-channel amplitudes that are not related to colour, we find for colour flow A the amplitude $M_s + M_t$ and for colour flow B the amplitude $M_u - M_t$. The cross-sections will then be given by

$$\sum_{\text{spins}} [M_s + M_t]^2 \sum_{\text{colours}} [A]^2 \quad \text{and} \quad \sum_{\text{spins}} [M_u - M_t]^2 \sum_{\text{colours}} [B]^2$$

(In this case they can also be read off directly from published calculations, e.g. [7,8].)

Of course, since we want to do a Monte Carlo simulation of these types of events, we shall have to deal specifically with the flavours of the hadron remnant $\bar{2}$ - a problem to which we will come back later (Section 3)-; it should be emphasized, however, that our arguments for the string configurations possible, in no way depends on our chosen prescription for dealing with the remnant flavours: the simple treatment of the remnant as effectively a gluon yields the correct answer to that part of the problem.

All other processes can now be treated in much the same way, the only case demanding special attention being $gg \rightarrow gg$; this reaction is dealt with in Appendix 2. The results are presented in Table 1, which gives, for each process, contributing diagrams, possible colour flows, string configurations, colour factors, amplitudes and cross-sections.

3. Hadron Remnants

There remains to deal with hadron remnants that are not simple quarks or diquarks.

For the case where the hadron has emitted a gluon, the remnant is a colour octet consisting of a colour triplet (a quark) and a colour antitriplet (for a meson: an antiquark; for a baryon: a diquark), loosely bound together. In the Lund model

it is then obvious how this octet should be connected to the rest of the system: the triplet (antitriplet) is simply hooked to the matching antitriplet (triplet) with the usual triplet string; this prescription is easily seen to be void of ambiguities. The schematic final state of e.g. fig.4b will thus really look like fig.5.

When a sea quark (antiquark) has been removed from the hadron, the remnant will be a colour antitriplet (triplet) consisting in the meson case of a quark and two antiquarks (two antiquarks and a quark); in the baryon case of three quarks and an antiquark (four quarks); and in the antibaryon case of four antiquarks (a quark and three antiquarks). For this case, the remnant is split into a colour neutral hadron and a colour antitriplet (triplet), i.e. in the case of a meson: a meson and an antiquark (a meson and a quark); in the case of a baryon: a meson and a diquark (a baryon and an quark); and in the case of an antibaryon: an antibaryon and an antiquark (a meson and an antidiquark). The particle formed is chosen from the set of possibilities allowed by flavour conservation according to the proper SU(6) weights, once more resulting in a prescription void of ambiguities.

The momenta of the hadron remnants we proceed to divide between the fragments in the following way. The gluon is considered as having been radiated off by a particular quark (antiquark) inside the hadron, so that the quark (antiquark) has to 'pay' for this by compensating for all of the primordial p_T of the gluon. To clarify: letting $\bar{k}_{Tq}$ denote the 'pre-primordial' p_T of the quark (i.e. the primordial p_T the quark would have had, had no gluon been emitted) and $\bar{p}_{Tg}$ the primordial p_T of the gluon, if a baryon emitted a gluon we will have $\bar{p}_T(\text{quark}) = \bar{k}_{Tq} - \bar{p}_{Tg}$; $\bar{p}_T(\text{diquark}) = -\bar{k}_{Tq}$. When a seaquark is pulled out of the hadron, its primordial p_T is compensated for by the colour neutral hadron formed; thus, for a baryon we will (in obvious notation) have $\bar{p}_T(\text{meson formed}) = -\bar{p}_{T \text{ sea}} + \bar{p}_{T \text{ val}}$; $\bar{p}_T(\text{diquark}) = -\bar{p}_{T \text{ val}}$.

The energies of the hadron fragments are given by simple constituent counting. Sea quarks are considered to contribute no dynamics (apart from their primordial k_T ; see above), just their flavour content; thus, if a baryon has emitted a gluon or a sea quark has been pulled out of it, there will in both cases be 3 quarks among which the remaining energy is to be distributed (remember that we choose to ignore the left-over sea antiquark in this connection!). For the case of a meson, this number will be 2. Consequently, a quark ought to take on the average $1/3$ of the remaining energy in the baryon, and $1/2$ in the meson case. This can be realized most simply by giving an energy fraction χ to the quark, with χ governed by the distribution $(1 - \chi)^{n-2} d\chi$, where n is the number of constituents to share the energy. A diquark in a baryon will then take its fraction χ according to the distribution $[1 - (1 - \chi)]d\chi \equiv \chi d\chi$. This approach has been used in order not to introduce further free parameters into the model.

4. Low- p_T , K-factors and All That

One could still speculate somewhat upon the transition between high- p_T and low- p_T physics. For low values of p_T in high- p_T reactions, the process $gg \rightarrow gg$ will be the dominating one, and a natural way to view low- p_T physics would be as the collapse of the string system in this process as the gluons become increasingly softer. The final result in the Lund model will then be two overlapping triplet strings. However, we know that in the fragmentation region low- p_T events look more as though the particles were produced from a single triplet string. A way out of this dilemma would be to say that maybe strings are not stretched all the way out to the endpoint partons. Perhaps the colour octets going out transversely only stretch the strings a corresponding distance in the incident parton direction, i.e., instead of the hitherto considered situation, as shown in fig.6a, the string configuration will be like fig.6b, where we have some kind of 'junction' (cf. [9]). In this case, as the gluons grow softer, they will stretch 'their' string pieces a correspondingly shorter distance, and

the collapse will result in just one string. A good testing-ground for discriminating between these different string concepts would be the QCD Compton reaction $q_i g \rightarrow q_i \gamma$; this kind of test is discussed in more detail in [10].

Finally, the question of K-factors in high- p_T will be briefly touched upon. One could argue, that the K-factor in Drell-Yan [11] arises from a propensity of the reacting quark and antiquark to match their colours. If no such matching occurred, the colours would admit of annihilation only in one case out of three. However, if the quarks and antiquarks in some way 'felt' each other's colours and tried to adjust them by radiating off very soft gluons, annihilation could ideally, if colour were thrown off with impunity, take place every time. The observed K-factor value of around 2 would then correspond to some sort of interpolation between these two extremes: the quarks and antiquarks can 'sense' each other's colours and adjust them, albeit not infinitely easy. But if this is how the K-factor arises in Drell-Yan reactions, there is nothing preventing the same mechanism from occurring in all reactions where a colour is annihilated, and this would effect also some of the subprocesses studied here. The consequences would be a raised cross-section for these processes (e.g. $gg \rightarrow q_i \bar{q}_i$, or state A in $q_i g \rightarrow q_i g$; see Table 1). This mechanism gains some support from the observation [12] of a K-factor around 2 in ψ -production in pp-collisions, which is thought to be dominated by gluon-gluon fusion.

5. The Program Components

PYTHIA version 3.3 is a successor to the earlier Lund high- p_T program PYTHIA [6]. From the physics point of view the main difference is the inclusion of scale-breaking structure functions. The code has to a large extent been rewritten also for efficiency reasons, the new version being significantly faster. The overall structure is, however, similar to the earlier version(s), although no explicit backwards compatibility exists. By changing the status code IPY(12) (see common PYPARA) to 0 (which gives simple scaling structure

functions), the user will be able to generate exactly the same type of events as with earlier versions of PYTHIA, for purposes of comparison. The program is built as a number of subroutines, two of which (PYINIT and PYTHIA) are called by the user, while the rest are mainly for internal use (but cf. PYKCUT). Information about the generated event is stored in a set of common blocks. All names of subroutines, functions and common blocks (apart from common LUJETS, see below) have been standardized to six letters, beginning with the identifier PY. The user supplies his own main program; avoiding names beginning with PY, he may then rest assured that no accidental ambiguities in the naming of routines will occur.

The jet fragmentation and particle decays are taken care of by the Lund Monte Carlo for jet fragmentation, JETSET [5], the point of contact between the two programs being the common block LUJETS, where the event record is stored. The user will thus need at least a working knowledge of Section 6.1 in [5], which deals with the organisation of the event record; a familiarity with the functions KLU and PLU, and the subroutines LULIST and LUEDIT would also be useful. PYTHIA version 3.3 by default assumes that the version of JETSET used is 5.2 (new standard); however, by changing the status code IPY(6) (see common PYPARA) to 2 it will be compatible also with JETSET version 4.3 (old standard).

The program is written in FORTRAN 77, its only non-standard feature being the requirements of a random number generator RANF and the ordinary (real) gamma function GAMMA. For the output, a standard output file with logical file number 6 is assumed.

5.1. Overall Flow of Program

The user initializes the generation with a call to PYINIT, specifying what kind of reaction to be studied (e.g. $\bar{p}p$ at $\sqrt{s} = 540$ GeV). In this subroutine all parameters that are constant in a given reaction are calculated and stored in common blocks; furthermore, the cross-sections for the different subprocesses

are set equal to zero, to be updated during the actual generation. Finally, via calls to the MINUIT-routines [13] included in the package, PYINIT computes the maximum of the distribution

$$\xi(x_1, x_2, \hat{s}, Q^2) = \sum_{i,j,k} \xi_{ij}^k(x_1, x_2, \hat{s}, Q^2)$$

$$\xi_{ij}^k(x_1, x_2, \hat{s}, Q^2) \equiv x_1 f_i^{\text{beam}}(x_1, Q^2) x_2 f_j^{\text{target}}(x_2, Q^2) \frac{1}{\pi} \times \\ \times \frac{d\sigma_{ij}^k(\hat{s}, \hat{t})}{d\hat{t}}$$

where the xf 's are the structure functions and

$$\frac{d\sigma_{ij}^k}{d\hat{t}}$$

are the parton level cross sections given in table 1. The indices i and j ($-4, \dots, 4$) represent the reacting partons and k ($1, \dots, 4$) the different possibilities for the final state; the summation goes over all subprocesses included. Each subsequent call to PYTHIA, which administers the generation chain, will then produce one event of the type specified in PYINIT.

The generation process takes its beginning in subroutine PYRAND. The variables x_1 and x_2 (the energy fractions of the reacting partons) and $\hat{t}$ (the Mandelstam momentum transfer variable) are chosen according to the wellknown cross section

$$d\sigma = \sum_{ijk} x_1 f_i^{\text{beam}}(x_1, Q^2) x_2 f_j^{\text{target}}(x_2, Q^2) \frac{1}{x_1 x_2} \frac{d\sigma_{ij}^k(\hat{s}, \hat{t})}{d\hat{t}} dx_1 dx_2 d\hat{t} \\ = \frac{\pi}{x_1 x_2} \xi(x_1, x_2, \hat{s}, Q^2) dx_1 dx_2 d\hat{t}$$

For efficiency reasons it is, however, advantageous to make a change of variables to $y_1 = x_1 x_2$ and $y_2 = x_1 - x_2$, rewriting the cross section as

$$d\sigma = \pi \frac{dy_1 dy_2 d\hat{t}}{y_1 \sqrt{y_2^2 + 4y_1}} \xi(y_1, y_2, \hat{t}, Q^2)$$

and choose the variables y_1 , y_2 and $\hat{t}$ exactly according to the first factor and thereafter use the maximum of ξ (as calculated in PYINIT) for a weighting procedure. The set $[i, j, k]$ is then generated from the individual weights ξ_{ij}^k defined above. To speed up the generation, it may in some cases (depending on the

$$\frac{d\sigma_{ij}^k}{d\hat{t}}$$

of the subprocesses included) be advantageous to take out factors of $1/\hat{s}$ or $1/\hat{t}$ from the ξ_{ij}^k 's; this can be regulated by changing the status codes IPY(14) and IPY(15) (see common PYPARA). A call to PYCONF takes the set $[i, j, k]$ and connects the strings according to Table 1. In subroutine PYJETS, energy and momenta are given to all partons; the string systems are then checked against energy cuts in subroutine PYSCUT. The cut employed for use with JETSET 5.2 is:

$$M_{\min}^2 = (1 \text{ GeV} + M_1^{\text{end}} + M_2^{\text{end}})^2$$

where M_i^{end} represents the masses of the partons at the ends of the string. For use with JETSET 4.3, somewhat harder cuts are employed; here, each leg of a string (where a leg is defined as the piece of string between any two adjacent partons) has to fulfill the above energy requirement (a gluon, 'split in halves', is assigned the value $M = 0.5 \text{ GeV}$ to account for the extra particle formed at the gluon corner [5]). Furthermore, for strings with one or more gluons, a cut to measure the degree of 'transversality' is also introduced when using JETSET 4.3:

$$\frac{(p_1^{\text{end}} + p_g)^2 (p_g + p_2^{\text{end}})^2}{(p_1^{\text{end}} + p_2^{\text{end}})^2} > 3 (\text{GeV})^2$$

These cuts will influence the calculation of cross-sections, since different subprocesses have different probabilities for surviving the cuts.

Finally, a call to subroutine LUEXEC in JETSET generates the hadrons in the high- p_T event (i.e. administers the fragmentation and particle decay).

5.2. Description of Subroutines and Functions

SUBROUTINE PYINIT(FRAME,BEAM,TARGET,WIN,QTMIN)

Purpose: to initialize the generation procedure.

Arguments:

FRAME: a character variable to specify the coordinate frame of the experiment. The following possibilities are included: 'FIXT' for fixed target experiments, 'CMS' for colliding beams in the center of momentum frame and 'USER', which gives the user full freedom to specify his frame by storing the momenta of the incoming particles in the P-array in common block LUJETS (p_x , p_y and p_z for beam particle should be stored as $P(1,1)$, $P(1,2)$ and $P(1,3)$ respectively; similarly for target particle as $P(2,1)$, $P(2,2)$ and $P(2,3)$).

BEAM, TARGET: character variables to specify beam and target particles, respectively (beam is defined as the particle moving in the positive z-direction; target as the particle at rest for FRAME='FIXT' or moving in the negative z-direction for FRAME='CMS'). Possible arguments are 'PI+' (π^+), 'PI-' (π^-), 'P' (p), 'PBAR' ($\bar{p}$), 'N' (n) and 'NBAR' ($\bar{n}$).

WIN: momentum (in GeV/c) for beam particle when FRAME = 'FIXT'; $\sqrt{s}$ (in GeV) when FRAME = 'CMS'; not used when FRAME = 'USER'.

QTMIN: minimum transverse momentum (in GeV/c) on the parton level (i.e. the minimum p_T required for a parton scattered in the hard process, in addition to other cuts; see above). Since the program is designed for high- p_T physics, it is recommended that the value of QTMIN never be set lower than 2.

Procedure: The initial state particles are transformed to their cms and the subprocesses to be included are set up in the INCLU-array.

For the Monte Carlo generation procedure an estimate of the

maximum value of the differential cross-section is needed. This is here calculated by minimizing the negative of the differential cross-section, using an adaption of the MINUIT routines. If this maximum is already known (from an earlier run having identical specifications as regards initial hadrons; their energies; subprocesses included; QTMIN; IPY(I), where I = 2, 8-15, and PYPAR(I), where I = 1-4, 11; see common PYPARA), the user can provide this maximum value in PYVAR(1), in which case the program skips this calculation.

SUBROUTINE PYTHIA

Purpose: to administer the generation of one high- p_T event using values set up in the latest PYINIT call. Also calculates all cross-sections in mbarn.

SUBROUTINE PYRAND

Purpose: to generate all quantities needed to specify the hard scattering on the parton level.

Procedure: The two parton momentum fractions x_1 and x_2 (X(1) and X(2)) for the partons in the beam and target particles, respectively, and the Mandelstam momentum transfer variable $\hat{s}$ (THAT) are chosen according to the differential cross-section formula. (More specifically: $Y1 = X(1)*X(2)$ and $Y2 = X(1) - X(2)$ are chosen instead for reasons of efficiency.) A call to PYKCUT is made, whereby the user can apply cuts on kinematical variables already on the parton level, before the event is actually generated. Also, the flavours of the reacting partons as well as the specific subprocess are chosen.

SUBROUTINE PYCONF

Purpose: to determine the colour field configuration, i.e. stretch the strings between the colour charged partons.

SUBROUTINE PYSPLI(KPART,KFLIN,KFLCH,KFLSP)

Purpose: to analyse hadron remnants, and if necessary split them into two fragments (for the case of a colour octet remnant: a colour triplet and an antitriplet; for the case of a colour triplet remnant - applicable when a sea quark has been removed -: a colour triplet and a colour neutral hadron).

Arguments:

KPART: flavour code for original hadron.

KFLIN: flavour code for parton that has been removed from hadron.

KFLCH: flavour code for one of the two hadron remnant fragments. For the case of a gluon emitted: flavour code for the quark (diquark) considered to have emitted the gluon; for the case of a sea quark (antiquark) removed: flavour code for hadron formed; when neither case applies, KFLCH = 0.

KFLSP: flavour code for the other of the two hadron remnant fragments.

Procedure: the different possibilities are chosen according to the proper SU(6) weights.

SUBROUTINE PYJETS

Purpose: to give energy and momenta to the partons in the reaction, and to check whether the strings of the final state contains enough energy to allow fragmentation.

SUBROUTINE PYSCUT

Purpose: to check that each of the strings in the string system contains enough energy to allow fragmentation.

Procedure: when used with JETSET 5.2 mild cuts are applied, amounting to a check that the invariant mass of each string is greater than $1 \text{ GeV} + M_1^{\text{end}} + M_2^{\text{end}}$, where M_i^{end} represents the mass of the partons at the ends of the string. When used with

JETSET 4.3, the same check is made for each leg of the strings (a leg being defined as the piece of string between any two adjacent partons in the string, and a gluon 'split in halves' being assigned a mass of 0.5 GeV to account for the extra particle formed at the gluon corner). In this case, the strings are also checked against a 'transversality cut'; see Section 5.1.

FUNCTION PYDSIG(I,J,KTYPE,SHAT,THAT,UHAT)

Purpose: gives differential cross-sections for all different subprocesses and colour flows of order α_s^2 and $\alpha_s \cdot \alpha_{em}$, with and without interference terms.

Arguments:

I,J: specifies the types of the reacting partons (-4,...,4; 4 = charm, 3 = strange, 2 = down, 1 = up, 0 = gluon, -1 = antiup, etc.).

KTYPE: specifies the different final states (1,...,4) for a given initial set of reacting partons; see Tables 2 & 3.

SHAT,THAT,UHAT: the Mandelstam variables $\hat{s}$, $\hat{t}$, $\hat{u}$.

SUBROUTINE PYSTRC(X,Q2)

Purpose: to turn the structure functions for proton/ π^+ given by PYSTFU into the required nucleon/antinucleon/pion structure functions using isospin and/or charge reflection.

Arguments:

X: array (X(2)) containing the momentum fractions of the reacting partons from beam and target particles respectively.

Q2: momentum transfer Q^2 .

SUBROUTINE PYSTFU(KFA,X,Q2,XPQ)

Purpose: to give parton structure functions.

Arguments:

KFA: absolute value of particle flavour code; 17 = π^+ , 41 = p.

X: momentum fraction carried by parton.

Q2: momentum transfer Q^2 .

XPQ: array (XPQ(-4:4)) that on return contains the values of the parton structure functions. Index: 4 = charm, 3 = strange, 2 = down, 1 = up, 0 = gluon, -1 = antiup, etc.

Procedure: different parametrizations of proton and π^+ structure functions are included. The choice is regulated by IPY(12). Note that what is coded is $x \cdot q(x, Q^2)$, etc.

SUBROUTINE PYROBO(PINX,PINY,PINZ,EIN,POUTX,POUTY,POUTZ,EOUT)

Purpose: to transform momenta from the cms of incoming partons to the corresponding momenta in the cms of the incoming particles, and record the rotations and boosts used.

Arguments:

PINX,PINY,PINZ,EIN: momenta and energy in cms of incoming partons.

POUTX,POUTY,POUTZ,EOUT: momenta and energy in cms of incoming particles.

Procedure: the rotations and boosts which take momenta from the cms of incoming partons to the cms of the incoming particles are stored as THE(2), PHI(2), BETA(2,1), BETA(2,2) and BETA(2,3) in common PYLORE.

SUBROUTINE PYFRAM(IFRAME)

Purpose: to transform the event into the desired coordinate frame.

Argument:

IFRAME: specification of frame; 1 = frame specified by user in last PYINIT call (i.e. either 'FIXT', 'CMS' or 'USER'); 2 = cms of incoming particles; 3 = cms of incoming reacting partons.

Procedure: starting from the current frame which is known by the program (stored as IPY(30)), the event is transformed to the frame specified by IFRAME.

SUBROUTINE PYKCUT(X,SHAT,THAT,QT,Q2,ICUT)

Purpose: to make kinematical cuts on the parton level before the event is generated.

Arguments:

X: array (X(2)) with the momentum fractions of the two reacting partons (from beam and target, respectively).

SHAT: $\hat{s}$, i.e. invariant mass squared of the reacting partons.

THAT: $\hat{t}$, the Mandelstam momentum transfer variable on the parton level.

QT: transverse momentum of scattered partons in cms of incoming reacting partons.

Q2: Q^2 , i.e. momentum transfer squared, used in structure functions and α_s . Note that different definitions can be chosen via IPY(11).

ICUT: a flag, which should be set equal to 0 if the event passes the cuts, and 1 otherwise.

Remarks: this routine is actually called before structure functions and QCD matrix elements are evaluated, so that cuts already at this level will speed up the program. Note, however, that in its present form this is a dummy routine which makes no cuts and which should be replaced by a user who wants to perform these kinds of cuts.

SUBROUTINE PYPRKT(PTX,PTY)

Purpose: to give primordial p_T to partons.

Arguments:

PTX,PTY: x- and y-components of primordial p_T .

Procedure: if $IPY(17) = 0$, PTX and PTY are set equal to zero (i.e. no primordial p_T is given to the parton); if $IPY(17) = 1$ (default), primordial p_T is chosen according to a Gaussian distribution with width $PYPAR(5)$, cut off at a maximum p_T of $PYPAR(6)$.

SUBROUTINE PYCHID(KPART,KFL,CHI)

Purpose: to give energy fractions to the fragments when a hadron remnant is split.

Arguments:

KPART: flavour code for original hadron.

KFL: flavour code of fragment of hadron remnant.

CHI: energy fraction χ taken by fragment.

Procedure: the energy fraction χ (CHI) of a quark (antiquark) is chosen according to the distribution $(1 - \chi)^k d\chi$, where $k = n - 2$, with n the number of partons among which the energy should be distributed. For a diquark, the distribution used is $[1 - (1 - \chi)^k] d\chi$; see Section 3.

SUBROUTINE PYALPH(Q2)

Purpose: to calculate the running coupling constant in QCD, α_s , as a function of the squared momentum transfer Q^2 .

Argument:

Q2: Q^2 , i.e. momentum transfer squared (definition regulated by $IPY(11)$).

Remarks: the number of active flavours is Q^2 -dependent, see $IPY(10)$. The value of the QCD parameter, Λ_{QCD} , is given by $PYPAR(3)$.

SUBROUTINE PYTHAT(SHAT,SQTMIN,THATL,THATU)

Purpose: to calculate the lower and upper limits of the Mandelstam momentum transfer variable $\hat{t}$ for a given parton-parton invariant mass squared, $\hat{s}$, and squared minimum parton transverse momentum, p_T^2 .

Arguments:

SHAT: $\hat{s}$.

SQTMIN: p_T^2 .

THATL: lower limit for $\hat{t}$.

THATU: upper limit for $\hat{t}$.

FUNCTION PYEULB(X,Y)

Purpose: calculates the Euler beta-function $\Gamma(x) \cdot \Gamma(y) / \Gamma(x+y)$ for use in the structure function parametrizations.

Arguments:

X,Y: rather obvious.

Remark: the (ordinary) gamma-function must be supplied in the program as FUNCTION GAMMA(X) (exists in FORTRAN 77 at the CERN IBM).

SUBROUTINE PYWHAT

Purpose: to fill the identifier of the subprocess, ICONF, in common PYPROC (see this and Table 4).

SUBROUTINE PYSIGM(NPAR,DERIV,DIFSIG,XF,IFLAG)

Purpose: to give the differential cross-section to the routines that find its maximum value.

Procedure: this routine, which gives the negative of the differential cross-section (DIFSIG), is called by the minimization routines to obtain an estimate of the maximum value of the differential cross-section for use in the Monte

Carlo generation procedure. Strictly speaking, only the part of the cross-section which can not be simulated exactly is included (see also IPY(14), IPY(15) and subroutine PYINIT).

SUBROUTINE PYTIME(TIME)

Purpose: to monitor the cpu time called for during the minimization.

Remark: in its present shape, the routine calls the machine dependent routine TIMEX in the IBM system at CERN. For users of other systems, it should be replaced, or the argument (TIME) simply set to zero. The time information is in no way essential.

```

SUBROUTINE PYCMND.....COMAND
SUBROUTINE PYINTO(...)...INTOEX(...)
SUBROUTINE PYMIDA.....MIDATA
.....ENTRY PYMID2.....MIDAT2
SUBROUTINE PYMINN.....MINNEW
SUBROUTINE PYMPRI(...)...MPRINT(...)
..FUNCTION PYPINT(...)...PINTF(...)
SUBROUTINE PYRAZZ(...)...RAZZIA(...)
SUBROUTINE PYSIMP.....SIMPLX

```

These routines, which are taken from the MINUIT program package [13] (the original names are given to the right), are used for the minimization procedure described above. They are slightly modified in order to be conveniently integrated into the program, and normally the user need not worry about them. There are, however, possibilities for the user to change the requirements of the numerical accuracy of the minimum found, via the common block PYMINU. As an extra safeguard against terminating at a local minimum, a new call to PYSIMP, with new error definitions and step sizes, is made after the first minimum has been found.

5.3. Description of Common Blocks

As stated before, the complete event record is stored in the common block LUJETS, described in [5]. Additional information about the event is stored in the common blocks PYCROS and PYPROC; the performance of the program is regulated via the status codes and parameters in common blocks PYPARA and PYSUBS. All parameters are given sensible default values in block data PYDATA; this is indicated below by (D=...). These values may be changed by the user in order to modify the behaviour of the program. Note, however, that some of the parameters are interrelated.

COMMON/PYPARA/IPY(30),PYPAR(20),PYVAR(10)

Purpose: to give access to status codes and parameters which regulate the performance of the program.

IPY(1): (D=2) determines the extent to which the program will react if the maximum calculated in PYINIT is violated during the simulation.

=0: the generation will stop and a message be printed.

=1: the generation will continue but a warning be printed. Each time the violation reaches a subsequently higher value, a new warning will be printed.

=2: the generation will continue but a warning be printed and the maximum increased to the higher value.

Note: in case the maximum is violated and execution continues, the simulated distributions will not be quite correct.

IPY(2): (D=4) number of flavours that can be produced in the reactions $q_i \bar{q}_i \rightarrow q_k \bar{q}_k$ or $gg \rightarrow q_k \bar{q}_k$.

IPY(3): (D=2) regulates whether the incoming particles and/or incoming reacting partons are to be included in the event record (i.e. stored in common LUJETS).

=0: incoming particles and partons excluded.

=1: incoming particles stored in lines 1-2 in common LUJETS.

=2: incoming particles stored in lines 1-2 and incoming reacting partons in lines 3-4 in common LUJETS.

IPY(4): (D=1) determines in what coordinate frame the event is to be presented.

=1: frame specified by user in latest call to PYINIT.

=2: cms of incoming particles; beam particle along positive z-axis.

=3: cms of reacting partons; parton from beam along positive z-axis.

IPY(5): presently unused.

IPY(6): (D=1) determines the criteria to be imposed on the final partonic system in order that it will be accepted by the fragmentation routines.

=0: no cuts; can be used together with independent fragmentation.

=1: minimum cuts for string fragmentation as in JETSET 5.2, i.e. a minimum invariant mass for each colour singlet system is required.

=2: harder cuts for use with JETSET 4.3, requiring a minimum invariant mass for each 'leg' (i.e. piece of string between any two adjacent partons) and a minimum 'transversality' of the string (see Section 5.1).

IPY(7): (D=1) regulates whether fragmentation is to be performed or not. (Fragmentation can e.g. be skipped to gain computer time, if only parton level is to be studied.)

=0: skip fragmentation. Note that by a call to LUEXEC, the user can then let the system fragment afterwards.

=1: perform fragmentation.

IPY(8): (D=1) regulates inclusion of interference terms in the QCD matrix elements and choice of fragmentation scheme.

=1: exclude interference terms; string fragmentation.

=2: exclude interference terms; independent jet fragmentation.

=3: include interference terms; string fragmentation.

=4: include interference terms; independent jet fragmentation.

IPY(9): (D=1) regulates whether α_s is to be Q^2 -dependent or not.

=0: fixed α_s , value given by PYPAR(2).

=1: running α_s .

IPY(10): (D=4) maximum number of active flavours in expression for running α_s the actual number included in a given event depends on Q^2 .

IPY(11): (D=1) regulates the choice of definition of Q^2 .

=1: $Q^2 = 2 \cdot \hat{s} \cdot \hat{t} \cdot \hat{u} / (\hat{s}^2 + \hat{t}^2 + \hat{u}^2)$.

=2: $Q^2 = p_T^2$, i.e. parton transverse momentum squared.

=3: $Q^2 = -\hat{t}$.

IPY(12): (D=1) regulates the choice of structure function parametrizations.

=0: simple scaling structure functions from [14,15], including only valence quarks and gluons.

=1: Q^2 -dependent structure functions from [16] (proton) and [15] (pion).

=2: Q^2 -dependent structure functions from [15] (proton and pion).

IPY(13): (D=3) maximum number of quark flavours in structure function parametrizations; 2 in case of IPY(12)=0, otherwise 3.

IPY(14): (D=0) regulates the simulation of the variable Y_1 defined above (see routine PYRAND), which is chosen exactly according to the distribution $\ln(Y_1)/Y_1^{**}(\text{IPY}(14)+1)$. (Of course, the rest of the differential cross-section is used for weighting later.) Depending on the energy scale and subprocesses included, the generation of events might be faster if this parameter is set instead equal to 1 or 2. See also IPY(15).

IPY(15): (D=0) regulates the simulation of the variable $\hat{t}$, which is chosen exactly according to the distribution $1/\hat{t}^{**}\text{IPY}(15)$. (Of course, the rest of the differential cross-section is used for weighting later.) Depending on the energy scale and the subprocesses included, the generation of events might be faster if this parameter is instead set equal to 1 or 2. See also IPY(14).

IPY(16): (D=1) regulates the distribution used in PYCHID for partitioning the energy among the fragments of a complicated hadron remnant as described in Section 3. The energy fraction, χ , is simulated according to a distribution $(k+1)(1-\chi)^k$ with mean value $1/(k+2)$. See Section 3; $k \equiv n - 2$.

=1: $k=0$ for meson remnants and 1 for baryon remnants. The numbers are chosen according to simple counting rules, i.e. $k+2$ is the number of partons to share the energy in the hadron remnant.

=2: k is a real number given by $PYPAR(7) - PYPAR(10)$.

IPY(17): (D=1) regulates the choice of primordial p_T distribution used in PYPRKT.

=0: no primordial p_T .

=1: primordial p_T simulated according to Gaussian distribution with width given by $PYPAR(5)$, cut off at a maximum $p_T = PYPAR(6)$.

IPY(18)-IPY(28): presently unused.

IPY(29): internal flag to signal failure to pass cuts. (This leads to the event being simulated anew.)

IPY(30): gives the current coordinate frame of the event being simulated. The code is the same as for IPY(4).

PYPAR(1): (D=0.0073) fine structure constant, i.e. α_{em} .

PYPAR(2): (D=0.2) α_s when constant.

PYPAR(3): (D=0.3) Λ_{QCD} used in running α_s .

PYPAR(4): (D=0.4) Λ_{QCD} used in parametrizations of structure functions.

PYPAR(5): (D=0.44) width in GeV/c of Gaussian distribution for primordial p_T .

PYPAR(6): (D=2.) maximum value in GeV/c of primordial p_T , i.e.

the value at which the Gaussian is cut off.

PYPAR(7): (D=0.) when IPY(16)=2 (see this) gives the value of the parameter k for the case where a pion remnant is split into two fragments.

PYPAR(8): (D=0.) ditto for the case where a pion remnant is split into a particle and a spectator fragment.

PYPAR(9): (D=1.) ditto for the case where a nucleon remnant is split into two fragments.

PYPAR(10): (D=1.) ditto for the case where a nucleon remnant is split into a particle and a spectator fragment.

PYPAR(11): (D=1.) K-factor used in subprocesses where colour annihilates. Default corresponds to the case without a K-factor. See Section 4.

PYPAR(12)-PYPAR(20): presently unused.

PYVAR(1): (D=0.) maximum of differential cross-section used for weighting. If this value is known from an earlier run it can be supplied by the user. In this case the program skips the (superfluous) calculation.

PYVAR(2): value of α_s in current event.

PYVAR(3)-PYVAR(10): presently unused.

COMMON/PYSUBS/ISELEC,ISUBPR(9),INCLU(-4:4,-4:4,4)

Purpose: to allow the user to run the program with any desired subset of high- p_T processes.

ISELEC: (D=0) a switch to select between preprogramming and full user control.

=0: all high- p_T subprocesses included.

=1: allows the user to choose between different preprogrammed

subsets of high- p_T subprocesses by setting the corresponding entries of ISUBPR = 1.

=2: gives the user full freedom to include subprocesses after his own choice by setting the corresponding entries of INCLU = 1.

ISUBPR: (D=1 for ISELEC=0, otherwise 0) an array (1:9) to be set only when ISELEC = 1. By setting ISUBPR(ICONF) = 1, the corresponding set of subprocesses will be included in the generation. (See Table 4 for correspondence between value of ICONF and subprocess.) Example: if ISUBPR(8) and ISUBPR(9) are set equal to 1, prompt photon production from the Compton and annihilation channels will be included.

INCLU: (D=1 for ISELEC=0, 1 or 0 as governed by ISUBPR for ISELEC=1, 0 for ISELEC=2) an array (-4:4,-4:4,4) to be set only when ISELEC = 2. By setting INCLU(I,J,K) = 1, the corresponding subprocess will be included in the generation (see Tables 2 & 3 for correspondence between elements of INCLU and actual process). Example: if INCLU(0,2,2) is set equal to 1, the reaction $g d \rightarrow g d$ (configuration B) will be included.

COMMON/PYCROS/CROSS(0:9),NPASS(0:9),NGEN(0:9),NREP,PRECRS(0:9)

Purpose: to give the cross-sections for different final states, and the total cross-section for a reaction.

CROSS: an array (0:9) which on return gives the cross-sections in mbarn for different final states; the correspondence between index and final state is given in Table 3 (cf. ICONF in common block PYPROC).

Note: all other arguments of this common block are used internally and should not be touched. The entries of CROSS are set equal to zero after a call to PYINIT and then updated with each call to PYTHIA (i.e. each event generated); no summation and averaging has to be made after the required number of events has been generated. The entries of PRECRS (corresponding to different sums of ξ_{ij}^k) are calculated for all subprocesses with each call to PYTHIA. NREP measures how many times a set of random variables has been generated; PRECRS(ICONF)/NREP thus

gives a first estimate of the cross-section for the subprocess defined by ICONF (see Table 4). NGEN(ICONF) measures how many times the subprocess defined by ICONF has been submitted to a test for fragmentation cuts and NPASS(ICONF) how many times it has passed that test. Hence, the cross-section, CROSS(ICONF), for that particular subprocess will be given by PRECRS(ICONF)/NREP*NPASS(ICONF)/NGEN(ICONF) times a constant factor converting this pure number into mbarn. For subprocesses that never passed the cuts, the cross-section is approximated instead with PRECRS(ICONF)/NREP*NPASS(0)/NGEN(0); 0 referring to the totality of subprocesses. Obviously, the cross-sections thus calculated are subject to some amount of uncertainty; for the rarer subprocesses especially, the fluctuations may become important, which should be borne in mind when quoting actual numbers.

COMMON/PYPROC/IN(2),KTYPE,ICONF,X(2),SHAT,THAT,Q2

Purpose: to specify the event on the parton level.

IN: flavour number of reacting partons (1 refers to beam, 2 to target); the notation used is 4 = charm, 3 = strange, 2 = down, 1 = up, 0 = gluon, -1 = antiup, etc.

KTYPE: number (1,...,4) to specify the result of a reaction on the parton level. The set [IN(1),IN(2),KTYPE] thus corresponds to the arguments of the array INCLU (cf. also subroutine PYDSIG).

ICONF: an identifier to specify the general type of subprocess that has occurred, cf. Table 3.

X: the energy fraction taken by the reacting parton (1 refers to beam, 2 to target).

SHAT: $\hat{s}$, i.e. invariant mass of the reacting partons.

THAT: $\hat{t}$, i.e. the Mandelstam momentum transfer variable for the subprocess.

Q2: Q^2 , i.e. the momentum transfer squared of the subprocess.

COMMON/PYINT1/S,YMIN,SQTMIN,XQ(2,-4:4),QTX,QTY,DIR,FLAFAC

Purpose: for internal use.

S: s , i.e. invariant mass squared of the reaction.

YMIN: minimum allowed value of the variable $Y_1 = X(1)*X(2)$
(from kinematics this is seen to be $4.*Q_{TMIN}^2/S$; cf. PYINIT).

SQTMIN: minimum allowed transverse momentum squared (i.e. Q_{TMIN}^2).

XQ: an array containing the value of the partonic structure functions. The first index refers to origin (1 = beam; 2 = target), while the second refers to the parton flavour number.

QTX, QTY: p_{Tx} and p_{Ty} , respectively, of the scattered partons.

DIR: 1. or -1., depending on whether the scattered parton is moving with positive or negative p_z .

FLAFAC: suppression factor for heavy flavour production (for quark-antiquark annihilation and gluon fusion processes). This factor is included since the QCD matrix elements used are for massless quarks.

COMMON/PYINT2/ISP(2), ICH(2), IRE(2), KFLIN(2), KFLCH(2), NPART,
NPRIM, CHI(2), PTIN(2,2), PTSP(2,2)

Purpose: for internal use.

ISP: line number (in common LUJETS) for spectator from beam (1) and target (2).

ICH: line number (in common LUJETS) for possible other hadron fragment (or directly produced hadron) from beam (1) and target (2).

IRE: line number (in common LUJETS) of scattered partons from beam (1) and target (2). (In some cases this numbering is obviously arbitrary.)

KFLIN: flavour code for incoming partons from beam (1) and target (2).

KFLCH: flavour code for possible extra hadron fragments (or directly produced hadrons) from beam (1) and target (2). (If no such fragment or hadron, KFLCH = 0.)

NPART: number of partons in the string system.

NPRIM: number of hadrons produced directly from beam and target remnants.

CHI: energy fraction taken by fragment of hadron remnant from beam (1) and target (2). Cf. Section 3 and subroutine PYCHID.

PTIN: primordial p_T of incoming partons. First index corresponds to origin (1 = beam; 2 = target); second index corresponds to x- and y-components.

PTSP: ditto for spectator partons.

COMMON/PYINT3/VIOL,VIOLMX,CONV,KI(2,2),PI(2,5),NMAX(-4:4,-4:4)

Purpose: for internal use.

VIOL: the value of $\xi(x_1, x_2, \xi, Q^2)/\xi_{\max}$ for each event, where $\xi_{\max}$ is the maximum estimated by PYINIT (see Section 5.1).

VIOLMX: current maximum value of VIOL (should be ≤ 1), see also IPY(1).

CONV: conversion factor to turn the entries of PRECRS into cross-sections expressed in mbarn.

KI,PI: arrays to store incoming particles temporarily.

NMAX: array containing maximum value of possible subprocesses for a given set of initial partons.

COMMON/PYLORE/THE(2),PHI(2),BETA(2,3)

Purpose: for internal use. Contains rotations and boosts needed to make transformations between different coordinate frames.

THE: polar angle.

PHI: azimuthal angle.

BETA: boost vector (second index corresponds to the x-, y- and z-components, respectively).

Note: the index (for BETA: the first index) corresponds to the following cases. 1: transformation from cms of incoming particles to the user specified frame. (This can of course be equal to the cms of the incoming particles, in which case THE(1), PHI(1), BETA(1,1), BETA(1,2) and BETA(1,3) are all equal to 0.) 2: transformation from cms of the reacting partons to the cms of the incoming particles. (THE, PHI, BETA to be applied in this order for these transformations.)

```
COMMON/PYMINU/XKIN(4),UKIN(4),WKIN(4),AIN(4),BIN(4),MAXFIN,  
RELUP,RELERR,RELER2,FCNMAX
```

Purpose: to supply the minimization routines with starting values, boundary values and error sizes.

XKIN: parameter number

UKIN: starting value of parameter.

WKIN: approximate step size to be used for parameter.

AIN: lower bound on parameter.

BIN: upper bound on parameter.

MAXFIN: (D=1000) maximum number of function evaluations allowed; the program will terminate if this is exceeded.

RELUP: (D=0.01) parameter UP in MINUIT routines.

RELERR: (D=0.05) tolerance on the minimum function value (EPSI in MINUIT).

RELER2: (D=0.05) multiplies RELERR to get a new tolerance when the second call to PYSIMP is made.

FCNMAX: absolute value of minimum found, returned as estimated maximum.

Note: for the arrays the index refers to Y1, Y2 and THAT, respectively, i.e. the minimization is performed with respect to these variables.

```
COMMON/PYMINC/NAMKIN(4),NAM(30)
```

Purpose: to give character names to the parameters in the minimization.

NAMKIN,NAM: character variables containing names of parameters.

6. Examples on How To Use the Program

The Monte Carlo is built as a slave system, i.e. the user supplies his own main program, which calls on the PYTHIA subroutines to perform different tasks and then resumes control. Below are some examples of statements that could appear in the main program. We recall that the program should be loaded together with the fragmentation routines JETSET version 5.2 or 4.3 [5]. The user has to supply a random number generator and the gamma function, since the program makes the references RANF(0) and GAMMA(X) to external real functions. The time needed to generate one event is around 0.05 s to 0.08 s, on the IBM 3081 at CERN, depending on the collision energy and the minimum transverse momentum (Q_{MIN}) required for the high p_T jets.

An ordinary pion-proton collision at SPS energies and with a required minimum p_T on the parton level of 2 GeV/c may be generated with

```
CALL PYINIT('FIXT','PI+','P',350.,2.)
CALL PYTHIA
```

The event can then be listed with

```
CALL LULIST(1)
```

which will give also a check on charge, energy and momentum conservation.

An actual analysis of, say, 10000 $\bar{p}p$ -collisions at collider energies and with a required minimum p_T on the parton level of 10 GeV/c, including a read-out of the cross-section, could be structured in the following way:

```
COMMON/LUJETS/N,K(250,2),P(250,5)
COMMON/PYCROS/CROSS(0:9),NPASS(0:9),NGEN(0:9),NREP,
&PRECRS(0:9)
CALL PYINIT('CMS','PBAR','P',540.,10.)
DO 100 I=1,10000
CALL PYTHIA
```

.	analysis
.	of the
.	generated
.	event

```
100 CONTINUE
```

```

      .
      .
      .
      WRITE(6,*)' Total cross-section is',      read-out
      &CROSS(0),'mbarn'                          of CROSS
      .
      .
      .
      END

```

A study of prompt photon production at ISR energies can be realized most simply like this:

```

COMMON/LUJETS/N,K(250,2),P(250,5)
COMMON/PYSUBS/ISELEC,ISUBPR(9),INCLU(-4:4,-4:4,4)
COMMON/PYCROSS/CROSS(0:9),NPASS(0:9),NGEN(0:9),NREP,
&PRECRS(0:9)
ISELEC=1
ISUBPR(8)=1
ISUBPR(9)=1
CALL PYINIT('CMS','P','P',63.,4.)
DO 100 I=1,10000
CALL PYTHIA
      .
      .
      .
      .
      analysis
      of the
      generated
      event
100 CONTINUE

```

```

      .
      WRITE(6,*)' Cross sections:',CROSS(8),
      &CROSS(9), 'mbarn.'
      .

```

```
END
```

Acknowledgements

It is pleasure to thank Torbjörn Sjöstrand for valuable suggestions and helpful comments on the program. We are also grateful to F. James for assistance concerning the MINUIT-routines. Needless to say, however, all responsibility for programming errors rests solely with the authors.

Appendix 1: Test Run Output

Example of an event generated with CALL PYINIT('CMS','P','PBAR',63.,6.) and CALL PYTHIA, listed with CALL LULIST(1). As can be seen, the process that occurred was $gg \rightarrow gg$, conf. A1+A2+B1+B2.

EVENT LISTING

I	ORI	PART/JET	PX	PY	PZ	E	M
1	0	P + B	0.000	0.000	31.486	31.500	0.938
2	0	P B- B	0.000	0.000	-31.486	31.500	0.938
3	0	G JETB	0.174	0.013	10.598	10.599	0.000
4	0	G JETB	-0.355	-0.674	-4.565	4.628	0.000
5	0	DU 0JETF	0.268	-0.193	8.109	8.142	0.650
6	0	DUA0JETF	-0.087	-0.247	-20.875	20.887	0.650
7	0	U JETF	-0.442	0.180	12.684	12.697	0.325
8	0	G JETF	5.889	-3.892	3.674	7.958	0.000
9	0	G JETF	-6.071	3.230	2.359	7.270	0.000
10	0	UA JETF	0.442	0.921	-5.951	6.046	0.325
11	5	N 0	0.051	0.099	3.904	4.017	0.940
12	6	P B-	-0.008	-0.098	-12.902	12.936	0.938
13	6	P +	-0.617	-0.081	0.227	1.148	0.938
14	5	N B	-0.349	-0.148	0.014	1.014	0.940
15	6	P B-	0.743	0.194	-3.889	4.074	0.938
16	6	PI0 D	0.274	-0.326	-1.483	1.549	0.135
17	6	RHO0 D	-0.574	0.487	0.152	1.087	0.770
18	6	PI +	0.614	0.038	0.087	0.637	0.140
19	6	PI B-	-0.548	-0.571	-0.209	0.830	0.140
20	5	P +	0.594	-0.034	1.333	1.735	0.938
21	8	DELT- D	2.056	-0.606	3.249	4.083	1.233
22	9	K* 0 D	-3.126	1.904	1.207	3.958	0.898
23	7	ETAP D	-0.512	0.643	7.880	7.981	0.958
24	8	PI0 D	0.071	-1.011	0.488	1.133	0.135
25	7	PI +	0.992	-0.628	3.817	3.996	0.140
26	8	DELTB+ D	1.388	-0.682	0.091	1.980	1.233
27	8	ETA D	-0.130	-0.509	-0.071	0.763	0.549
28	9	PI0 D	0.916	-0.475	1.129	1.535	0.135
29	10	K B-	-0.338	0.227	-0.270	0.695	0.494

Event listing, cont.

30	10	K*	0	D	-0.329	0.602	-2.197	2.471	0.898
31	10	PI	+		0.687	-0.069	-0.608	0.930	0.140
32	10	PI0		D	-0.491	0.400	-1.205	1.368	0.135
33	9	K*	B-	D	-1.024	-0.310	0.300	1.425	0.892
34	10	OMEG		D	-0.340	0.956	-1.045	1.653	0.782
35	16	GAMM			0.018	-0.006	-0.224	0.224	0.000
36	16	GAMM			0.256	-0.321	-1.260	1.325	0.000
37	17	PI	+		0.062	-0.009	-0.150	0.215	0.140
38	17	PI	B-		-0.636	0.496	0.303	0.873	0.140
39	21	N	0		1.914	-0.582	2.648	3.449	0.940
40	21	PI	B-		0.142	-0.024	0.601	0.634	0.140
41	22	K	+		-1.014	0.750	0.354	1.400	0.494
42	22	PI	B-		-2.113	1.154	0.853	2.558	0.140
43	23	PI	+		-0.032	0.103	1.459	1.470	0.140
44	23	PI	B-		0.066	0.060	1.068	1.081	0.140
45	23	ETA		D	-0.546	0.480	5.353	5.430	0.549
46	24	GAMM			0.060	-0.597	0.359	0.699	0.000
47	24	GAMM			0.011	-0.414	0.129	0.434	0.000
48	26	N	B		1.301	-0.579	0.256	1.725	0.940
49	26	PI	+		0.087	-0.104	-0.165	0.255	0.140
50	27	GAMM			0.208	-0.286	-0.046	0.357	0.000
51	27	GAMM			-0.339	-0.223	-0.025	0.406	0.000
52	28	GAMM			0.711	-0.370	0.789	1.124	0.000
53	28	GAMM			0.205	-0.105	0.340	0.411	0.000
54	30	K	0	D	-0.410	0.593	-1.353	1.612	0.498
55	30	PI0		D	0.080	0.008	-0.845	0.859	0.135
56	32	GAMM			-0.372	0.353	-1.034	1.155	0.000
57	32	GAMM			-0.120	0.047	-0.170	0.213	0.000
58	33	K	B	D	-0.604	-0.480	0.176	0.935	0.498
59	33	PI	B-		-0.420	0.170	0.124	0.490	0.140
60	34	PI	+		0.091	0.060	-0.102	0.204	0.140
61	34	PI	B-		-0.391	0.750	-0.588	1.040	0.140
62	34	PI0		D	-0.039	0.145	-0.355	0.409	0.135
63	45	GAMM			-0.418	0.366	5.062	5.093	0.000
64	45	GAMM			-0.128	0.114	0.291	0.337	0.000
65	54	K0L			-0.410	0.593	-1.353	1.612	0.498
66	55	GAMM			-0.026	-0.028	-0.126	0.132	0.000
67	55	GAMM			0.107	0.036	-0.719	0.727	0.000

Event listing, cont.

68	58	K0S	D	-0.604	-0.480	0.176	0.935	0.498	
69	62	GAMM		-0.045	0.150	-0.202	0.255	0.000	
70	62	GAMM		0.005	-0.004	-0.153	0.153	0.000	
71	68	PI	+	-0.141	-0.073	-0.113	0.240	0.140	
72	68	PI	B-	-0.462	-0.407	0.289	0.694	0.140	
SUM:				0.000	0.000	-0.000	0.000	63.000	11.238

Appendix 2: $gg \rightarrow gg$

(Note: this appendix is taken verbatim from [6].)

The amplitude describing the process $gg \rightarrow gg$ has contributions from four different diagrams, representing s-, t- and u-channel exchanges and the 4-gluon interaction. According to Table 1, these are given by:

$$s: -ig^2 \frac{1}{\hat{s}} f^{abe} f^{cde} \epsilon_1^\kappa \epsilon_2^\lambda \epsilon_3^\mu \epsilon_4^\nu C_{\kappa\lambda\tau}(-q_1, -q_2, q_1+q_2) \times \\ \times C_{\mu\nu\tau}(q_3, q_4, -q_3-q_4) \equiv f^{abe} f^{cde} M_s \quad (1)$$

$$t: -ig^2 f^{dae} f^{bce} \frac{1}{\hat{t}} \epsilon_1^\kappa \epsilon_2^\lambda \epsilon_3^\mu \epsilon_4^\nu C_{\nu\kappa\tau}(q_4, -q_1, q_1-q_4) \times \\ \times C_{\lambda\mu\tau}(-q_2, q_3, q_2-q_3) \equiv f^{dae} f^{bce} M_t \quad (2)$$

$$u: -ig^2 f^{cae} f^{bde} \epsilon_1^\kappa \epsilon_2^\lambda \epsilon_3^\mu \epsilon_4^\nu C_{\mu\kappa\tau}(q_3, -q_1, q_1-q_3) \times \\ \times C_{\lambda\nu\tau}(-q_2, q_4, q_2-q_4) \equiv f^{cae} f^{bde} M_u \quad (3)$$

$$4: -ig^2 f^{abe} f^{cde} (g_{\kappa\mu} g_{\lambda\nu} - g_{\kappa\nu} g_{\lambda\mu}) - ig^2 f^{dae} f^{bce} \times \\ \times (g_{\kappa\mu} g_{\lambda\nu} - g_{\kappa\lambda} g_{\nu\mu}) + ig^2 f^{cae} f^{bde} (g_{\kappa\lambda} g_{\mu\nu} - g_{\kappa\nu} g_{\lambda\mu}) \equiv \\ \equiv f^{abe} f^{cde} M_4^s + f^{dae} f^{bce} M_4^t + f^{cae} f^{bde} M_4^u \quad (4)$$

Here, $C^{\alpha\beta\gamma}$ is defined as

$$C^{\alpha\beta\gamma}(q_1, q_2, q_3) = (q_1 - q_2)^\gamma g^{\alpha\beta} + (q_2 - q_3)^\alpha g^{\beta\gamma} + \\ + (q_3 - q_1)^\beta g^{\alpha\gamma} \quad (5)$$

From Table 1 we can also read off the colour factors corresponding to the different possible colour flows:

$$A_1: T_{\delta\alpha}^a T_{\alpha\beta}^b T_{\beta\gamma}^c T_{\gamma\delta}^d \equiv \text{Tr}(T^a T^b T^c T^d) \quad (6)$$

$$A_2: T_{\delta\gamma}^d T_{\gamma\beta}^c T_{\beta\alpha}^b T_{\alpha\delta}^a \equiv \text{Tr}(T^d T^c T^b T^a) \quad (7)$$

$$B_1: T_{\delta\alpha}^a T_{\alpha\beta}^b T_{\beta\gamma}^d T_{\gamma\delta}^c \equiv \text{Tr}(T^a T^b T^d T^c) \quad (8)$$

$$B_2: T_{\delta\gamma}^c T_{\gamma\beta}^d T_{\beta\alpha}^b T_{\alpha\delta}^a \equiv \text{Tr}(T^c T^d T^b T^a) \quad (9)$$

$$C_1: T_{\delta\alpha}^a T_{\alpha\beta}^c T_{\beta\gamma}^b T_{\gamma\delta}^d \equiv \text{Tr}(T^a T^c T^b T^d) \quad (10)$$

$$C_2: T_{\delta\gamma}^d T_{\gamma\beta}^b T_{\beta\alpha}^c T_{\alpha\delta}^a \equiv \text{Tr}(T^d T^b T^c T^a) \quad (11)$$

Using the identity

$$[T^a, T^b]_{\alpha\beta} = if^{abc} T_{\alpha\beta}^c \quad (12)$$

we now prove the following:

$$A_1 - B_1 = \text{Tr}(T^a T^b [T^c, T^d]) = if^{cde} \text{Tr}(T^a T^b T^e) \quad (13)$$

$$\begin{aligned} A_2 - B_2 &= \text{Tr}([T^d, T^c] T^b T^a) = if^{dce} \text{Tr}(T^e T^b T^a) = \\ &= -if^{cde} \text{Tr}(T^b T^a T^e) \end{aligned} \quad (14)$$

Thus:

$$\begin{aligned} (A_1 + A_2) - (B_1 + B_2) &= if^{cde} \text{Tr}([T^a, T^b] T^e) = \\ &= -f^{abf} f^{cde} \text{Tr}(T^f T^e) = -\frac{1}{2} f^{abe} f^{cde} \end{aligned} \quad (15)$$

or

$$f^{abe} f^{cde} = -2(A_1 + A_2) + 2(B_1 + B_2) \quad (16)$$

In the same way it can be shown that

$$f^{cae} f^{bde} = -2(B_1 + B_2) + 2(C_1 + C_2) \quad (17)$$

$$f^{dae} f^{bce} = 2(C_1 + C_2) - 2(A_1 + A_2) \quad (18)$$

The total amplitude for the process $qq \rightarrow gg$ can then be rewritten:

$$\begin{aligned} F_{gg \rightarrow gg} &= f^{abe} f^{cde} M_s + f^{dae} f^{bce} M_t + f^{cae} f^{bde} M_u + \\ &+ f^{abe} f^{cde} M_4^s + f^{dae} f^{bce} M_4^t + f^{cae} f^{bde} M_4^u = \\ &= -(A_1 + A_2) 2(M_s + M_t + M_4^s + M_4^t) + (B_1 + B_2) \times \\ &\times 2(M_s - M_u + M_4^s + M_4^u) + (C_1 + C_2) 2(M_u + M_t - M_4^u + M_4^t) \end{aligned} \quad (19)$$

To find the cross-section for a certain colour flow, say A_1 (or A_2), we thus need to compute

$$\sum_{\text{spins}} [M_s + M_t + M_4^s + M_4^t]^2 \quad (20)$$

Let us first note that the spin-summation will contain factors like

$$\sum_{\text{spin1}} \epsilon_1^\lambda \epsilon_1^\mu$$

Defining

$$\begin{aligned} A^{12\lambda\mu} &= \sum_{\text{spin1}} \epsilon_1^\lambda \epsilon_1^\mu = \sum_{\text{spin2}} \epsilon_2^\lambda \epsilon_2^\mu = -g^{\lambda\mu} + \frac{2}{\hat{s}} (q_1^\lambda q_2^\mu + q_1^\mu q_2^\lambda) \\ A^{34\lambda\mu} &= \sum_{\text{spin3}} \epsilon_3^\lambda \epsilon_3^\mu = \sum_{\text{spin4}} \epsilon_4^\lambda \epsilon_4^\mu = -g^{\lambda\mu} + \frac{2}{\hat{s}} (q_3^\lambda q_4^\mu + q_3^\mu q_4^\lambda) \end{aligned} \quad (21)$$

it is only a matter of some straightforward algebra to prove the following useful identities:

$$A_{\alpha}^{12\alpha} = A_{\alpha}^{34\alpha} = -2 \quad (22)$$

$$q_1^\alpha A_{\alpha\beta}^{12} = q_2^\alpha A_{\alpha\beta}^{12} = q_3^\alpha A_{\alpha\beta}^{34} = q_4^\alpha A_{\alpha\beta}^{34} = 0 \quad (23)$$

$$q_1^\alpha q_1^\beta A_{\alpha\beta}^{34} = q_4^\alpha q_4^\beta A_{\alpha\beta}^{12} = -\hat{t}(1 + \frac{\hat{t}}{\hat{s}}) \quad (24)$$

$$A^{12\alpha\beta} A_{\beta}^{12\gamma} = A^{12\alpha\gamma}; \quad A^{34\alpha\beta} A_{\beta}^{34\gamma} = A^{34\alpha\gamma} \quad (25)$$

$$A^{12\alpha\beta} A_{\alpha\beta}^{12} = A^{34\alpha\beta} A_{\alpha\beta}^{34} = 2 \quad (26)$$

$$A^{12\alpha\beta} A_{\alpha\beta}^{34} = 2(1 + 2\frac{\hat{t}}{\hat{s}} + 2\frac{\hat{t}^2}{\hat{s}^2}) \quad (27)$$

$$q_1^\alpha q_4^\beta A_{\beta\gamma}^{12} A_{\alpha}^{34\gamma} = -\hat{t}(1 + 3\frac{\hat{t}}{\hat{s}} + 2\frac{\hat{t}^2}{\hat{s}^2}) \quad (28)$$

$$\begin{aligned} q_1^\alpha q_1^\beta A_{\beta\gamma}^{34} A_{\delta\alpha}^{12\gamma\delta} A_{\delta\alpha}^{34} &= q_4^\alpha q_4^\beta A_{\beta\gamma}^{12} A_{\delta\alpha}^{34\gamma\delta} A_{\delta\alpha}^{12} = \\ &= -\hat{t}(1 + 5\frac{\hat{t}}{\hat{s}} + 8\frac{\hat{t}^2}{\hat{s}^2} + 4\frac{\hat{t}^3}{\hat{s}^3}) \end{aligned} \quad (29)$$

$$A_{\alpha\beta}^{12} A_{\gamma\delta}^{34\beta\gamma} A_{\gamma\delta}^{12} A_{\delta\alpha}^{34} = 2(1 + 4\frac{\hat{t}}{\hat{s}} + 12\frac{\hat{t}^2}{\hat{s}^2} + 16\frac{\hat{t}^3}{\hat{s}^3} + 8\frac{\hat{t}^4}{\hat{s}^4}) \quad (30)$$

$$\begin{aligned}
q_1^\alpha q_4^\beta A_{\beta\gamma}^{12} A^{34\gamma\delta} A_{\delta\epsilon}^{12} A^{34\epsilon}{}_\alpha = \\
= -\hat{t}(1 + 7\frac{\hat{t}}{\hat{s}} + 18\frac{\hat{t}^2}{\hat{s}^2} + 20\frac{\hat{t}^3}{\hat{s}^3} + 8\frac{\hat{t}^4}{\hat{s}^4})
\end{aligned} \quad (31)$$

$$C^{\alpha\beta\tau}(-q_1, -q_2, q_1+q_2) A_{\alpha\alpha'}^{12} A_{\beta\beta'}^{12} = (q_1 - q_2)^\tau A_{\alpha'\beta'}^{12} \quad (32)$$

$$C^{\alpha\beta\tau}(q_3, q_4, -q_3-q_4) A_{\alpha\alpha'}^{34} A_{\beta\beta'}^{34} = (q_4 - q_3)^\tau A_{\alpha'\beta'}^{34} \quad (33)$$

$$\begin{aligned}
C^{\alpha\beta\tau}(q_4, -q_1, q_1-q_4) A_{\alpha\alpha'}^{34} A_{\beta\beta'}^{12} = & -2q_1^\alpha A_{\alpha\alpha'}^{34} A_{\beta\beta'}^{12\tau} - \\
& -2q_4^\beta A_{\alpha'\beta'}^{34\tau} A_{\beta\beta'}^{12} + (q_1 + q_4)^\tau A_{\alpha'\beta'}^{34\alpha} A_{\beta'\alpha}^{12}
\end{aligned} \quad (34)$$

$$\begin{aligned}
C^{\alpha\beta\tau}(-q_2, q_3, q_2-q_3) A_{\alpha\alpha'}^{12} A_{\beta\beta'}^{34} = & -2q_1^\beta A_{\alpha'\beta'}^{12\tau} A_{\beta\beta'}^{34} - \\
& -2q_4^\alpha A_{\alpha\alpha'}^{12} A_{\beta\beta'}^{34\tau} - (q_2 + q_3)^\tau A_{\alpha'\beta}^{12} A_{\beta'}^{34\beta}
\end{aligned} \quad (35)$$

Armed with this knowledge, we may confidently proceed to compute the cross-section for the colour flows A_1 & A_2 in $gg \rightarrow gg$; a few more pages of algebra then produces the pleasingly symmetric result

$$\frac{\pi \alpha_s^2}{\hat{s}^2} \frac{9}{4} \left(\frac{\hat{s}^2}{\hat{t}^2} + 2\frac{\hat{s}}{\hat{t}} + 3 + 2\frac{\hat{t}}{\hat{s}} + \frac{\hat{t}^2}{\hat{s}^2} \right) \quad (36)$$

The formulae for B_1 , B_2 and C_1 , C_2 are obtained by simply permuting $\hat{s}$, $\hat{t}$ and $\hat{u}$. We observe in particular, that our result for the total cross-section for $gg \rightarrow gg$ (given some trivial manipulation of the Mandelstam variables) agrees with the generally quoted expression:

$$\frac{\pi \alpha_s^2}{\hat{s}^2} \frac{9}{2} \left(3 - \frac{\hat{t}\hat{u}}{\hat{s}^2} - \frac{\hat{u}\hat{s}}{\hat{t}^2} - \frac{\hat{s}\hat{t}}{\hat{u}^2} \right) \quad (37)$$

References

- [1] B.Andersson, G.Gustafson, G.Ingelman, T.Sjöstrand
Phys.Rep. 97 (1983) 31
- [2] G.Gustafson
Z.Phys. C15 (1982) 155
- [3] B.Andersson, G.Gustafson, T.Sjöstrand
Phys.Lett. 94B (1980) 211
- [4] JADE collaboration, W.Bartel et al.
Phys.Lett. 101B (1981) 129
- [5] T.Sjöstrand
Comp.Phys.Comm. 27 (1982) 243
- [6] H.-U.Bengtsson
Lund preprint LU TP 82-15,
to be published in Comp.Phys.Comm.
- [7] F.Halzen, D.M.Scott
Phys.Rev. D18 (1978) 3378
- [8] R.Cutler, D.Sivers
Phys.Rev. D17 (1978) 196
- [9] I.Montvay
Phys.Lett. 84B (1979) 331
- [10] B.Andersson, H.-U.Bengtsson, G.Gustafson
Lund preprint LU TP 82-10
- [11] R.K.Ellis
In ed. J.Tran Thanh Van,
Proc. 1st Moriond Workshop
(1981, Edition Frontières, Dreux) 231

- [12] R.Phillips
In ed. L.Durand and L.G.Pondrom
Proc. XXth International Conference on High Energy Physics
(1981, AIP, New York)

- [13] T.James, M.Roos
Comp.Phys.Comm. 10 (1975) 343

- [14] G.Altarelli, G.Martinelli
Phys.Lett. 76B (1978) 89

- [15] J.F.Owens, E.Reya
Phys.Rev. D17 (1978) 3003

- [16] M.Gluck, E.Hoffman, E.Reya
Z.Phys. C13 (1982) 119

TABLE 1: Subprocesses, String Configurations and Cross-sections
(Note: this table is taken verbatim from [6].)

Process: $q_i \bar{q}_i \rightarrow g\gamma$

Diagrams:

Amplitudes:

$$t: -ie_i g T_{\alpha\beta}^a \bar{v}_i(q_2) \not{\epsilon}_3 \frac{\not{q}_1 - \not{q}_4}{\hat{t}} \not{\epsilon}_4 u_i^\alpha(q_1)$$

$$u: -ie_i g T_{\alpha\beta}^a \bar{v}_i(q_2) \not{\epsilon}_4 \frac{\not{q}_1 - \not{q}_3}{\hat{u}} \not{\epsilon}_3 u_i^\alpha(q_1)$$

Colour flow:

String configuration:

Colour factor: $T_{\alpha\beta}^a$

Amplitude:

$$-ie_i g \bar{v}_i(q_2) (\not{\epsilon}_3 \frac{\not{q}_1 - \not{q}_4}{\hat{t}} \not{\epsilon}_4 + \not{\epsilon}_4 \frac{\not{q}_1 - \not{q}_3}{\hat{u}} \not{\epsilon}_3) u_i^\alpha(q_1)$$

Cross-section:

$$\frac{\pi \alpha_s \alpha_{em}}{\hat{s}^2} \frac{8}{9} q_i^2 \frac{\hat{t}^2 + \hat{u}^2}{\hat{t}\hat{u}} \quad (q_i \equiv e_i/e)$$

TABLE 1, cont.

Process: $q_i g \rightarrow q_i \gamma$ ($\bar{q}_i g \rightarrow \bar{q}_i \gamma$)

Diagrams:

Amplitudes:

$$s: -ie_i g T_{\alpha\beta}^a \bar{u}_i^\beta(q_4) \not{\epsilon}_3 \frac{\not{\epsilon}_1 + \not{\epsilon}_2}{\hat{s}} \not{\epsilon}_2 u_i^\alpha(q_1)$$

$$u: -ie_i g T_{\alpha\beta}^a \bar{u}_i^\beta(q_4) \not{\epsilon}_2 \frac{\not{\epsilon}_1 - \not{\epsilon}_3}{\hat{u}} \not{\epsilon}_3 u_i^\alpha(q_1)$$

$$(T_{\alpha\beta}^a \leftrightarrow T_{\beta\alpha}^a; u \leftrightarrow \bar{v})$$

Colour flow:

String configuration:

Colour factor: $T_{\alpha\beta}^a (T_{\beta\alpha}^a)$

Amplitude:

$$-ie_i g \bar{u}_i^\beta(q_4) (\not{\epsilon}_3 \frac{\not{\epsilon}_1 + \not{\epsilon}_2}{\hat{s}} \not{\epsilon}_2 + \not{\epsilon}_2 \frac{\not{\epsilon}_1 - \not{\epsilon}_3}{\hat{u}} \not{\epsilon}_3) u_i^\alpha(q_1)$$

$$(u \leftrightarrow \bar{v})$$

Cross-section:

$$\frac{\pi \alpha_s \alpha_{em}}{\hat{s}^2} \frac{1}{3} q_i^2 \frac{\hat{s}^2 + \hat{u}^2}{-\hat{s}\hat{u}} \quad (q_i \equiv e_i/e)$$

TABLE 1, cont.

Process: $q_i q_j \rightarrow q_i q_j$ ($\bar{q}_i \bar{q}_j \rightarrow \bar{q}_i \bar{q}_j$)

Diagrams:

Amplitudes:

$$t: ig^2 T_{\alpha\delta}^a T_{\beta\gamma}^a \frac{1}{\hat{t}} \bar{u}_i^\delta(q_4) \gamma_\kappa u_i^\alpha(q_1) \bar{u}_j^\gamma(q_3) \gamma^\kappa u_j^\beta(q_2)$$

$$u: -ig^2 T_{\beta\delta}^a T_{\alpha\gamma}^a \frac{1}{\hat{u}} \bar{u}_i^\delta(q_4) \gamma_\kappa u_i^\beta(q_2) \bar{u}_j^\gamma(q_3) \gamma^\kappa u_j^\alpha(q_1)$$

$$(T_{\alpha\delta}^a T_{\beta\gamma}^a \leftrightarrow T_{\delta\alpha}^a T_{\gamma\beta}^a; T_{\beta\delta}^a T_{\alpha\gamma}^a \leftrightarrow T_{\delta\beta}^a T_{\gamma\alpha}^a; u \leftrightarrow \bar{v})$$

Colour flows:

String configurations:

Colour factors: A: $T_{\alpha\delta}^a T_{\beta\gamma}^a (T_{\delta\alpha}^a T_{\gamma\beta}^a)$ B: $T_{\beta\delta}^a T_{\alpha\gamma}^a (T_{\delta\beta}^a T_{\gamma\alpha}^a)$

Amplitudes:

$$A: ig^2 \frac{1}{\hat{t}} \bar{u}_i^\delta(q_4) \gamma_\kappa u_i^\alpha(q_1) \bar{u}_j^\gamma(q_3) \gamma^\kappa u_j^\beta(q_2)$$

$$B: -ig^2 \frac{1}{\hat{u}} \bar{u}_i^\delta(q_4) \gamma_\kappa u_i^\beta(q_1) \bar{u}_j^\gamma(q_3) \gamma^\kappa u_j^\alpha(q_1)$$

$$(u \leftrightarrow \bar{v})$$

Cross-sections:

$$A: \frac{\pi \alpha_s^2}{\hat{s}^2} \frac{4}{9} \frac{\hat{s}^2 + \hat{u}^2}{\hat{t}^2}$$

$$B: \frac{\pi \alpha_s^2}{\hat{s}^2} \frac{4}{9} \frac{\hat{s}^2 + \hat{t}^2}{\hat{u}^2}$$

TABLE 1, cont.

Process: $q_i \bar{q}_j \rightarrow q_i \bar{q}_j$; $q_i \bar{q}_i \rightarrow q_k \bar{q}_k$

Diagrams:

Amplitudes:

$$t: ig^2 T_{\alpha\delta}^a T_{\gamma\beta}^a \frac{1}{\hat{t}} \bar{u}_i^\delta(q_4) \gamma_\kappa u_i^\alpha(q_1) \bar{v}_j^\beta(q_2) \gamma^\kappa v_j^\gamma(q_3)$$

$$s: -ig^2 T_{\gamma\delta}^a T_{\alpha\beta}^a \frac{1}{\hat{s}} \bar{u}_k^\delta(q_4) \gamma_\kappa v_k^\gamma(q_3) \bar{v}_i^\beta(q_2) \gamma^\kappa u_i^\alpha(q_1)$$

Colour flows:

String configurations:

Colour factors: A: $T_{\gamma\beta}^a T_{\alpha\delta}^a$ B: $T_{\gamma\delta}^a T_{\alpha\beta}^a$

Amplitudes:

$$A: ig^2 \frac{1}{\hat{t}} \bar{u}_i^\delta(q_4) \gamma_\kappa u_i^\alpha(q_1) \bar{v}_j^\beta(q_2) \gamma^\kappa v_j^\gamma(q_3)$$

$$B: -ig^2 \frac{1}{\hat{s}} \bar{u}_k^\delta(q_4) \gamma_\kappa v_k^\gamma(q_3) \bar{v}_i^\beta(q_2) \gamma^\kappa u_i^\alpha(q_1)$$

Cross-sections:

$$A: \frac{\pi \alpha_s^2}{\hat{s}^2} \frac{4}{9} \frac{\hat{s}^2 + \hat{u}^2}{\hat{t}^2}$$

$$B: \frac{\pi \alpha_s^2}{\hat{s}^2} \frac{4}{9} \frac{\hat{t}^2 + \hat{u}^2}{\hat{s}^2}$$

TABLE 1, cont.

Process: $q_i \bar{q}_i \rightarrow gg$

Diagrams:

Amplitudes:

$$s: g^2 f^{abc} T_{\alpha\beta}^c \bar{v}_i^\beta(q_2) \frac{\epsilon_3^\kappa \epsilon_4^\lambda \gamma^\mu}{\hat{s}} C_{\kappa\lambda\mu}(q_3, q_4, -q_3 - q_4) u_i^\alpha(q_1)$$

$$t: -ig^2 T_{\alpha\gamma}^b T_{\gamma\beta}^a \bar{v}_i^\beta(q_2) \not{\epsilon}_3 \frac{\not{q}_1 - \not{q}_4}{\hat{t}} \not{\epsilon}_4 u_i^\alpha(q_1)$$

$$u: -ig^2 T_{\alpha\gamma}^a T_{\gamma\beta}^b \bar{v}_i^\beta(q_2) \not{\epsilon}_4 \frac{\not{q}_1 - \not{q}_3}{\hat{u}} \not{\epsilon}_3 u_i^\alpha(q_1)$$

Colour flows:

String configurations:

Colour factors: A: $T_{\alpha\gamma}^b T_{\gamma\beta}^a$ B: $T_{\alpha\gamma}^a T_{\gamma\beta}^b$

Amplitudes:

$$A: -ig^2 \bar{v}_i^\beta(q_2) \left[\not{\epsilon}_3 \frac{\not{q}_1 - \not{q}_4}{\hat{t}} \not{\epsilon}_4 - \frac{\epsilon_3^\kappa \epsilon_4^\lambda \gamma^\mu}{\hat{s}} C_{\kappa\lambda\mu}(q_3, q_4, -q_3 - q_4) \right] \times u_i^\alpha(q_1)$$

$$B: -ig^2 \bar{v}_i^\beta(q_2) \left[\not{\epsilon}_4 \frac{\not{q}_1 - \not{q}_3}{\hat{u}} \not{\epsilon}_3 + \frac{\epsilon_3^\kappa \epsilon_4^\lambda \gamma^\mu}{\hat{s}} C_{\kappa\lambda\mu}(q_3, q_4, -q_3 - q_4) \right] \times u_i^\alpha(q_1)$$

Cross-sections:

$$A: -\frac{\pi \alpha_s^2}{\hat{s}^2} \frac{32}{27} \left(\frac{\hat{u}}{\hat{t}} - 2 \frac{\hat{u}^2}{\hat{s}^2} \right)$$

$$B: -\frac{\pi \alpha_s^2}{\hat{s}^2} \frac{32}{27} \left(\frac{\hat{t}}{\hat{u}} - 2 \frac{\hat{t}^2}{\hat{s}^2} \right)$$

TABLE 1, cont.

Process: $q_i g \rightarrow q_i g$ ($\bar{q}_i g \rightarrow \bar{q}_i g$)

Diagrams:

Amplitudes:

$$s: -ig^2 T_{\alpha\gamma}^a T_{\gamma\beta}^b \bar{u}_i^\beta(q_4) \not{\epsilon}_3 \frac{\not{q}_1 + \not{q}_2}{\hat{s}} \not{\epsilon}_2 u_i^\alpha(q_1)$$

$$t: g^2 f^{abc} T_{\alpha\beta}^c \bar{u}_i^\beta(q_4) \frac{\epsilon_2^\kappa \epsilon_3^\lambda \gamma^\mu}{\hat{t}} C_{\kappa\lambda\mu}(-q_2, q_3, q_2 - q_3) u_i^\alpha(q_1)$$

$$u: -ig^2 T_{\alpha\gamma}^b T_{\gamma\beta}^a \bar{u}_i^\beta(q_4) \not{\epsilon}_2 \frac{\not{q}_1 - \not{q}_3}{\hat{u}} \not{\epsilon}_3 u_i^\alpha(q_1)$$

$$(T_{\alpha\gamma}^a T_{\gamma\beta}^b \leftrightarrow T_{\beta\gamma}^b T_{\gamma\alpha}^a; T_{\alpha\gamma}^b T_{\gamma\beta}^a \leftrightarrow T_{\beta\gamma}^a T_{\gamma\alpha}^b; u \leftrightarrow \bar{v})$$

Colour flows:

String configurations:

Colour factors: A: $T_{\alpha\gamma}^a T_{\gamma\beta}^b (T_{\beta\gamma}^b T_{\gamma\alpha}^a)$ B: $T_{\alpha\gamma}^b T_{\gamma\beta}^a (T_{\beta\gamma}^a T_{\gamma\alpha}^b)$

Amplitudes:

$$A: -ig^2 \bar{u}_i^\beta(q_4) [\not{\epsilon}_3 \frac{\not{q}_1 + \not{q}_2}{\hat{s}} \not{\epsilon}_2 + \frac{\epsilon_2^\kappa \epsilon_3^\lambda \gamma^\mu}{\hat{t}} C_{\kappa\lambda\mu}(-q_2, q_3, q_2 - q_3)] \times$$

$$xu_i^\alpha(q_1)$$

$$B: -ig^2 \bar{u}_i(q_4) [\not{\epsilon}_2 \frac{\not{q}_1 - \not{q}_3}{\hat{u}} \not{\epsilon}_3 - \frac{\epsilon_2^\kappa \epsilon_3^\lambda \gamma^\mu}{\hat{t}} c_{\kappa\lambda\mu}(-q_2, q_3, q_2 - q_3)] \times$$

$$xu_i^\alpha(q_1)$$

$$(u \leftrightarrow \bar{v})$$

Cross-sections:

$$A: \frac{\pi \alpha_s^2}{\hat{s}^2} \frac{4}{9} \left(2 \frac{\hat{u}^2}{\hat{t}^2} - \frac{\hat{u}}{\hat{s}} \right)$$

$$B: \frac{\pi \alpha_s^2}{\hat{s}^2} \frac{4}{9} \left(2 \frac{\hat{s}^2}{\hat{t}^2} - \frac{\hat{s}}{\hat{u}} \right)$$

TABLE 1, cont.

Process: $gg \rightarrow q_i \bar{q}_i$

Diagrams:

Amplitudes:

$$s: g^2 f^{abc} T_{\alpha\beta}^c \bar{u}_i^\beta(q_4) \frac{\epsilon_1^\kappa \epsilon_2^\lambda \gamma^\mu}{\hat{s}} C_{\kappa\lambda\mu}(q_1, q_2, -q_1 - q_2) v_i^\alpha(q_3)$$

$$t: -ig^2 T_{\alpha\gamma}^b T_{\gamma\beta}^a \bar{u}_i^\beta(q_4) \not{\epsilon}_1 \frac{\not{q}_1 - \not{q}_4}{\hat{t}} \not{\epsilon}_2 v_i^\alpha(q_3)$$

$$u: -ig^2 T_{\alpha\gamma}^a T_{\gamma\beta}^b \bar{u}_i^\beta(q_4) \not{\epsilon}_2 \frac{\not{q}_1 - \not{q}_3}{\hat{u}} \not{\epsilon}_1 v_i^\alpha(q_3)$$

Colour flows:

String configurations:

Colour factors: A: $T_{\alpha\gamma}^b T_{\gamma\beta}^a$ B: $T_{\alpha\gamma}^a T_{\gamma\beta}^b$

Amplitudes:

$$A: -ig^2 \bar{u}_i^\beta(q_4) \left[\not{\epsilon}_1 \frac{\not{q}_1 - \not{q}_4}{\hat{t}} \not{\epsilon}_2 - \frac{\epsilon_1^\kappa \epsilon_2^\lambda \gamma^\mu}{\hat{s}} C_{\kappa\lambda\mu}(q_1, q_2, -q_1 - q_2) \right] \times$$

$$x v_i^\alpha(q_3)$$

$$B: -ig^2 \bar{u}_i^\beta(q_4) \left[\not{\epsilon}_2 \frac{\not{q}_1 - \not{q}_3}{\hat{u}} \not{\epsilon}_1 + \frac{\epsilon_1^\kappa \epsilon_2^\lambda \gamma^\mu}{\hat{s}} C_{\kappa\lambda\mu}(q_1, q_2, -q_1 - q_2) \right] \times$$

$$x v_i^\alpha(q_3)$$

Cross-sections:

$$A: -\frac{\pi \alpha_s^2}{\hat{s}^2} \frac{1}{6} \left(\frac{\hat{u}}{\hat{t}} - 2 \frac{\hat{u}^2}{\hat{s}^2} \right)$$

$$B: -\frac{\pi \alpha_s^2}{\hat{s}^2} \frac{1}{6} \left(\frac{\hat{t}}{\hat{u}} - 2 \frac{\hat{t}^2}{\hat{s}^2} \right)$$

TABLE 1, cont.

Process: $gg \rightarrow gg$

Amplitudes:

$$s: -ig^2 \frac{1}{\hat{s}} f^{abe} f^{cde} \epsilon_1^\kappa \epsilon_2^\lambda \epsilon_3^\mu \epsilon_4^\nu C_{\kappa\lambda\tau}(-q_1, -q_2, q_1+q_2) \times$$

$$\times C_{\mu\nu\tau}(q_3, q_4, -q_3-q_4)$$

$$t: -ig^2 \frac{1}{\hat{t}} f^{dae} f^{bce} \epsilon_1^\kappa \epsilon_2^\lambda \epsilon_3^\mu \epsilon_4^\nu C_{\nu\kappa\tau}(q_4, -q_1, q_1-q_4) \times$$

$$\times C_{\lambda\mu\tau}(-q_2, q_3, q_2-q_3)$$

$$u: -ig^2 \frac{1}{\hat{u}} f^{cae} f^{bde} \epsilon_1^\kappa \epsilon_2^\lambda \epsilon_3^\mu \epsilon_4^\nu C_{\mu\kappa\tau}(q_3, -q_1, q_1-q_3) \times$$

$$\times C_{\lambda\nu\tau}(-q_2, q_4, q_2-q_4)$$

$$4: -ig^2 f^{abe} f^{cde} (g_{\kappa\mu} g_{\lambda\nu} - g_{\kappa\nu} g_{\lambda\mu}) -ig^2 f^{dae} f^{bce} \times$$

$$\times (g_{\kappa\mu} g_{\lambda\nu} - g_{\kappa\lambda} g_{\nu\mu}) +ig^2 f^{cae} f^{bde} (g_{\kappa\lambda} g_{\mu\nu} - g_{\kappa\nu} g_{\lambda\mu})$$

Colour flows:

String configurations:

Colour factors: $A_1: T_{\delta\alpha}^a T_{\alpha\beta}^b T_{\beta\gamma}^c T_{\gamma\delta}^d$ $A_2: T_{\delta\gamma}^d T_{\gamma\beta}^c T_{\beta\alpha}^b T_{\alpha\delta}^a$

$$B_1: T_{\delta\alpha}^a T_{\alpha\beta}^b T_{\beta\gamma}^d T_{\gamma\delta}^c$$
 $B_2: T_{\delta\gamma}^c T_{\gamma\beta}^d T_{\beta\alpha}^b T_{\alpha\delta}^a$

$$C_1: T_{\delta\alpha}^a T_{\alpha\beta}^c T_{\beta\gamma}^b T_{\gamma\delta}^d$$
 $C_2: T_{\delta\gamma}^d T_{\gamma\beta}^b T_{\beta\alpha}^c T_{\alpha\delta}^a$

Amplitudes:

$$\begin{aligned}
 A_1, A_2: & 2ig^2 \epsilon_1^\kappa \epsilon_2^\lambda \epsilon_3^\mu \epsilon_4^\nu \left[\frac{1}{\hat{s}} C_{\kappa\lambda\tau}(-q_1, -q_2, q_1+q_2) \times \right. \\
 & \quad \times C_{\mu\nu}{}^\tau(q_3, q_4, -q_3-q_4) + \frac{1}{\hat{t}} C_{\nu\kappa\tau}(q_4, -q_1, q_1-q_4) \times \\
 & \quad \times C_{\lambda\mu}{}^\tau(-q_2, q_3, q_2-q_3) + 2g_{\kappa\mu} g_{\lambda\nu} - g_{\kappa\nu} g_{\lambda\mu} - g_{\kappa\lambda} g_{\nu\mu} \left. \right] \\
 B_1, B_2: & -2ig^2 \epsilon_1^\kappa \epsilon_2^\lambda \epsilon_3^\mu \epsilon_4^\nu \left[\frac{1}{\hat{s}} C_{\kappa\lambda\tau}(-q_1, -q_2, q_1+q_2) \times \right. \\
 & \quad \times C_{\mu\nu}{}^\tau(q_3, q_4, -q_3-q_4) - \frac{1}{\hat{u}} C_{\mu\kappa\tau}(q_3, -q_1, q_1-q_3) \times \\
 & \quad \times C_{\lambda\nu}{}^\tau(-q_2, q_4, q_2-q_4) - 2g_{\kappa\nu} g_{\lambda\mu} + g_{\kappa\mu} g_{\lambda\nu} + g_{\kappa\lambda} g_{\nu\mu} \left. \right] \\
 C_1, C_2: & -2ig^2 \epsilon_1^\kappa \epsilon_2^\lambda \epsilon_3^\mu \epsilon_4^\nu \left[\frac{1}{\hat{u}} C_{\mu\kappa\tau}(q_3, -q_1, q_1-q_3) \times \right. \\
 & \quad \times C_{\lambda\nu}{}^\tau(-q_2, q_4, q_2-q_4) + \frac{1}{\hat{t}} C_{\nu\kappa\tau}(q_4, -q_1, q_1-q_4) \times \\
 & \quad \times C_{\lambda\mu}{}^\tau(-q_2, q_3, q_2-q_3) - 2g_{\kappa\lambda} g_{\nu\mu} + g_{\kappa\mu} g_{\lambda\nu} + g_{\kappa\nu} g_{\lambda\mu} \left. \right]
 \end{aligned}$$

Cross-sections:

$$\begin{aligned}
 A_1 + A_2: & \frac{\pi \alpha_s^2}{\hat{s}^2} \frac{9}{4} \left(\frac{\hat{s}^2}{\hat{t}^2} + 2 \frac{\hat{s}}{\hat{t}} + 3 + 2 \frac{\hat{t}}{\hat{s}} + \frac{\hat{t}^2}{\hat{s}^2} \right) \\
 B_1 + B_2: & \frac{\pi \alpha_s^2}{\hat{s}^2} \frac{9}{4} \left(\frac{\hat{u}^2}{\hat{s}^2} + 2 \frac{\hat{u}}{\hat{s}} + 3 + 2 \frac{\hat{s}}{\hat{u}} + \frac{\hat{s}^2}{\hat{u}^2} \right) \\
 C_1 + C_2: & \frac{\pi \alpha_s^2}{\hat{s}^2} \frac{9}{4} \left(\frac{\hat{t}^2}{\hat{u}^2} + 2 \frac{\hat{t}}{\hat{u}} + 3 + 2 \frac{\hat{u}}{\hat{t}} + \frac{\hat{u}^2}{\hat{t}^2} \right)
 \end{aligned}$$

TABLE 2: Elements of INCLU and Corresponding Subprocesses

The general structure of INCLU can be described as follows
(refer also to Table 3):

<u>Element</u>	<u>Remarks</u>	<u>Subprocess</u>	<u>Configuration</u>
INCLU(i,j,1)	$i \neq j, i \neq -j, i \neq 0$	$q_i q_j \rightarrow q_i q_j$	A
INCLU(i,j,2)	void		
INCLU(i,j,3)	void		
INCLU(i,j,4)	void		
INCLU(i,i,1)	$i \neq 0$	$q_i q_i \rightarrow q_i q_i$	A+B
INCLU(i,i,2)	void		
INCLU(i,i,3)	void		
INCLU(i,i,4)	void		
INCLU(i,-i,1)	$i \neq 0$	$q_i \bar{q}_i \rightarrow q_i \bar{q}_i$	A
INCLU(i,-i,2)	$i \neq 0$	$q_i \bar{q}_i \rightarrow q_k \bar{q}_k$	B
INCLU(i,-i,3)	$i \neq 0$	$q_i \bar{q}_i \rightarrow gg$	A+B
INCLU(i,-i,4)	$i \neq 0$	$q_i \bar{q}_i \rightarrow \gamma g$	
INCLU(0,i,1)	$i \neq 0$	$g q_i \rightarrow g q_i$	A
INCLU(0,i,2)	$i \neq 0$	$g q_i \rightarrow g q_i$	B
INCLU(0,i,3)	$i \neq 0$	$g q_i \rightarrow \gamma q_i$	
INCLU(0,i,4)	void		
INCLU(i,0,1)	$i \neq 0$	$q_i g \rightarrow q_i g$	A
INCLU(i,0,2)	$i \neq 0$	$q_i g \rightarrow q_i g$	B
INCLU(i,0,3)	$i \neq 0$	$q_i g \rightarrow q_i \gamma$	
INCLU(i,0,4)	void		
INCLU(0,0,1)		$gg \rightarrow q_k \bar{q}_k$	B
INCLU(0,0,2)		$gg \rightarrow q_k \bar{q}_k$	A
INCLU(0,0,3)		$gg \rightarrow gg$	A1+A2+B1+B2
INCLU(0,0,4)		$gg \rightarrow gg$	C1+C2

TABLE 3: Synoptical Guide to the Elements of INCLU

The first two arguments of INCLU(I,J,K) are given by the row and column index, respectively (incoming partons); the third argument by the number representing the corresponding matrix entry (resulting partons). Note, that I and J actually cover the range -4 to 4, i.e. anticharm to charm.

	-2 ($\bar{d}$)	-1 ($\bar{u}$)	0 (g)	1 (u)	2 (d)
-2 ($\bar{d}$)	1: $\bar{d}\bar{d}$ 2: $\bar{d}g$ 3: $\bar{d}\gamma$	1: $\bar{d}\bar{u}$	1: $\bar{d}g$ 2: $\bar{d}g$ 3: $\bar{d}\gamma$	1: $\bar{d}u$	1: $\bar{d}d$ 2: $\bar{q}_i q_i$ 3: gg 4: $g\gamma$
-1 ($\bar{u}$)	1: $\bar{u}\bar{d}$	1: $\bar{u}\bar{u}$	1: $\bar{u}g$ 2: $\bar{u}g$ 3: $\bar{u}\gamma$	1: $\bar{u}u$ 2: $\bar{q}_i q_i$ 3: gg 4: $g\gamma$	1: $\bar{u}d$
0 (g)	1: $g\bar{d}$ 2: $g\bar{d}$ 3: $\gamma\bar{d}$	1: $g\bar{u}$ 2: $g\bar{u}$ 3: $\gamma\bar{u}$	1: $q_i \bar{q}_i$ 2: $q_i \bar{q}_i$ 3: gg 4: gg	1: gu 2: gu 3: γu	1: gd 2: gd 3: γd
1 (u)	1: $u\bar{d}$	1: $u\bar{u}$ 2: $q_i \bar{q}_i$ 3: gg 4: $g\gamma$	1: ug 2: ug 3: $u\gamma$	1: uu	1: ud
2 (d)	1: $d\bar{d}$ 2: $q_i \bar{q}_i$ 3: gg 4: $g\gamma$	1: $d\bar{u}$	1: dg 2: dg 3: $d\gamma$	1: du	1: dd

TABLE 4: ICONF, CROSS and Corresponding Subprocesses

ICONF	Reaction	Cross-section (in mbarn) given by
-----	Sum of all reactions	CROSS(0)
1	$q_i \bar{q}_j \rightarrow q_i \bar{q}_j$ $q_i \bar{q}_j \rightarrow q_i \bar{q}_j$ $\bar{q}_i \bar{q}_j \rightarrow \bar{q}_i \bar{q}_j$	CROSS(1)
2	$q_i \bar{q}_i \rightarrow q_j \bar{q}_j$	CROSS(2)
3	$q_i \bar{q}_i \rightarrow gg$	CROSS(3)
4	$q_i g \rightarrow q_i g$ $\bar{q}_i g \rightarrow \bar{q}_i g$	CROSS(4)
5	$gg \rightarrow q_i \bar{q}_i$	CROSS(5)
6	$gg \rightarrow gg$ (configs. A1+A2+B1+B2)	CROSS(6)
7	$gg \rightarrow gg$ (configs. C1+C2)	CROSS(7)
8	$q_i \bar{q}_i \rightarrow g\gamma$	CROSS(8)
9	$q_i g \rightarrow q_i \gamma$ $\bar{q}_i g \rightarrow \bar{q}_i \gamma$	CROSS(9)

Figure captions

Fig.1 Two different possible string configurations corresponding to the same set of triplets and antitriplets.

Fig.2 The s-, t- and u-channel diagrams contributing to the process $q_i g \rightarrow q_i g$.

Fig.3 The different colour flows in $q_i g \rightarrow q_i g$.

Fig.4 The final string configurations corresponding to the colour flows in fig.3.

Fig.5 A more realistic picture of the final state in the subprocess $q_i g \rightarrow q_i g$, configuration B. The triplet and antitriplet of the hadron remnant are connected with triplet strings to the other triplet and antitriplet partaking of the reaction (cf. fig.4b). The numbers indicate the colour charges.

Fig.6 Two different concepts of the string. (a) Ordinary Lund model: the (triplet) strings are stretched all the way out to the endpoint triplets (antitriplets). (b) Strings with 'junctions': the strings belonging to the octet are stretched only to a 'junction' at a distance from the endpoint triplets (antitriplets).

Fig. 1

Fig. 2

Fig. 3

Fig. 4

Fig. 5

Fig. 6