December 1997

Ariadne version 4

A Program for Simulation of QCD-Cascades
Implementing the Colour Dipole Model
Revision 10

Leif Lönnblad

NORDITA
Blegdamsvej 17
DK-2100 Copenhagen Ø, Denmark
E-mail: leif@nordita.dk

(original manual published in **Computer Physics Communication** [1])

Abstract

The fourth version of the Ariadne program for generating QCD cascades in the Colour Dipole Approximation is presented. The underlying physics issues is discussed and a manual for using the program is given together with a few sample programs.

1 Introduction

The Colour Dipole Model (CDM) [2–5] as implemented in the Ariadne program has had considerable success in describing data from both e^+e^- [6] and lepto-production [7] experiments.

The CDM differs from other QCD cascade models in that it in a natural way correctly treats most QCD coherence effects by describing the gluon bremsstrahlung in terms of radiation from colour dipoles between partons, instead of treating partons as independent emitters.

Ariadne is one of the "Lund family of Monte Carlo programs" and is not a complete event generator. It only generates the QCD cascade process and has to be interfaced to other programs which handles hard interactions, hadronization and particle decays. Standard interfaces to the JETSET [8–10], LEPTO [11] and PYTHIA [8, 9, 12] programs are included in the version presented in this paper.

1.1 Update history

The first version of Ariadne [13] only handled gluon radiation from primary quarks in e^+e^- collisions. Since then the program has evolved as follows

- version 2 [14]: Included gluon emission from extended emitters to describe the QCD showers in lepto-production.
- version 3 [15]: Included also production of $Q\bar{Q}$ pairs from gluon splitting.
- version 3.1: Adopted the new event record of JETSET version 7.1.
- version 3.2: Included preliminary facilities for generating dipole showers in hadron-hadron collisions.
- version 3.3: Included a preliminary treatment of electro-magnetic dipole radiation of photons.
- version 4.01: Completely rewritten, built around a new internal dipole oriented event record. The preliminary features of sub-versions 3.2 and 3.3 are properly included. Streamlined interfaces to the JETSET, LEPTO and PYTHIA are introduced as well as a new jet clustering routine, inspired by the CDM.
- version 4.02: Fixed bugs in connection with phase space restrictions (in 4.01 the switch `MSTA(11)` was inactive and effectively set to 4 - see section 3.4). Also a bug related to the option `MSTA(20) = 2`, and to the quark masses and charges in QED emissions was fixed.
- version 4.03: Added options for emissions from extended dipoles (see description of `MSTA(18)`, `MSTA(25)` and `PARA(25)` in section 3.4). Also introduced two new sets of tuned parameters (`'ALEPH'` and `'EMC'`) in subroutine `ARTUNE`. Please note that, in connection with this, a few parameters and switches were given new default values. Note especially that the default is to automatically call `ARTUNE('EMC')` in subroutine `ARINIT`, hence the switch `MSTA(3)` must be set to 0 before the call to `ARINIT` if the user wants to modify any parameters or switches.

  In previous releases, increased precision could be obtained by removing the comment of an implicit double precision statement. From this release on, increased precision is the default (see appendix A).
- version 4.04: Fixed minor bug in `ARCLUS` and speeded up the clustering algorithm. Added minor features to `ARCLUS` to make it more compatible with the `LUCLUS` routine.
- version 4.05: Fixed bug in matrix element for the final state $g \to q\bar{q}$ process (see `MSTA(23)`) and added a new option for this process (`MSTA(28)`).

  Introduced matching to the exact $\mathcal{O}(\alpha_S)$ matrix elements for the first emission in DIS (see `MSTA(32)` and `MSTA(33)`).

  Introduced pomeron-like scattering procedure for remnant handling in DIS and in hadron-hadron collisions (see `MSTA(34)`).

  Introduced a preliminary (and un-documented) treatment of Drell-Yan-like processes in hadron-hadron collisions (see `MSTA(22)`).

  Revised the PYTHIA interface. All PYTHIA sub-processes are now possible (although no guarantee is given for the physical relevance of adding a dipole shower to a given sub-process).

  Changed default parameters and switches to reflect the new features. In particular the default call to `ARTUNE('EMC')` has changed, and to retain the behavior of version 4.04 a call to `ARTUNE('4.04')` should be made instead.

  When run with LEPTO or PYTHIA, the subroutines dealing with structure functions in these programs are called from within Ariadne. This complicates the installation procedure somewhat (see section A.2).

- version 4.06: Fixed bug in the $\mathcal{O}(\alpha_S)$ matrix element for boson-gluon fusion diagram in DIS.

  Introduced initial state $g \to q\bar{q}$ splitting into the cascade (with first emission correced for the $\mathcal{O}(\alpha_S)$ matrix element in case of DIS).

  Changed default parameters and switches to reflect the new features. In particular the default call to `ARTUNE('EMC')` has changed, and to retain the behavior of version 4.05 (except for the bug above) a call to `ARTUNE('4.05')` should be made instead.

- version 4.07: Included correction for $\mathcal{O}(\alpha_S)$ matrix elements and improved the treatment of Drell-Yan production.

  Introduced a colour reconnection scheme governed by `MSTA(35)` and `PARA(26)` (switched off by default). This feature is so far not documented here.

  Changed default treatment of extendedness. For remnants, μ is given by the intrinsic $p_\perp$, which is now generated inside of Ariadne (see `MSTA(36)`, `MSTA(37)` and `PARA(27)`). By default the struck quark in DIS is now extended with $\mu = Q$.

  Changed default parameters and switches to reflect the new features. In particular, the default call to `ARTUNE('EMC')` has changed, and to retain the behavior of version 4.06 a call to `ARTUNE('4.06')` should be made instead.

- version 4.08: Fixed bug in orientation of emissions from $q\bar{q}$ dipoles. This mainly affects the general orientation of e^+e^- events i.e. the thrust-axis distribution, but may also affect prompt photon production in e^+e^-.

- version 4.09: A bug was found in the jetfinding routines ARCLUS. These routines have now been completely rewritten and is now a separate package called DICLUS, still distributed with Ariadne. The ARCLUS routine still works the same (apart from the fixed bug) and is a simple interface to DICLUS.

- version 4.10: Included simple model for fragmentation production of heavy quarkonium.

  Included hooks to run with the Linked Dipole Chain Monte Carlo for deeply inelastic electron-hadron scattering.

1.2 Programming philosophy

The program is a library of FORTRAN 77 subroutines to be called from a user supplied main program. Although there exist over fifty subroutines, only a handful are meant to be explicitly called by the user.

The interface to the main program as well as to other programs are handled through the event record in JETSET's `LUJETS` common block. The communication between different Ariadne routines is however handled by an internal dipole oriented event record in the `ARSTRS`, `ARDIPS` and `ARPART` common blocks, where the dipoles and partons are linked together to form "Lund-type" strings.

1.3 About this manual

This paper is divided into three parts. The first part, section 2, explain the underlying physics processes modeled in Ariadne. Section 3 describes the actual program components and in section 4 a couple of sample main programs are given to illustrate how Ariadne is used. In the appendix information on how to obtain, install and test the program is given.

2 The Colour Dipole Model

The CDM is based on the fact that a gluon emitted from a $q\bar{q}$ pair in an e^+e^- collision can be treated as radiation from the colour dipole between the q and $\bar{q}$, and that to a good approximation, the emission of a second softer gluon can be treated as radiation from two independent dipoles, one between the q and g and one between the g and $\bar{q}$.

In the CDM this is generalized so that the emission of a third, still softer gluon, is given by three independent dipoles etc.

2.1 Gluon emission

For gluon emission there are three different kinds of colour dipoles to be considered; $q\bar{q}$, qg (or $\bar{q}g$) and gg dipoles. The cross section for each of these can be calculated from the relevant Feynman diagrams in figure 1 and can be written

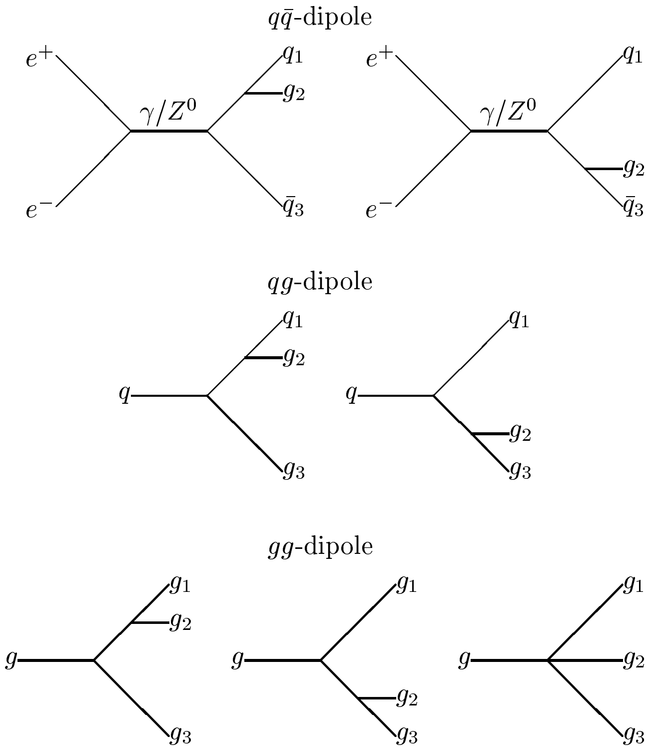

Figure 1: The relevant Feynman diagrams for gluon emission from a $q\bar{q}$, qg and a gg dipole

as

$$\frac{d\sigma_{q\bar{q}}}{dx_1 dx_3} = \frac{2\alpha_s}{3\pi}\frac{x_1^2+x_3^2}{(1-x_1)(1-x_3)} \tag{1}$$

$$\frac{d\sigma_{qg}}{dx_1 dx_3} = \frac{3\alpha_s}{4\pi}\frac{x_1^3+x_3^2}{(1-x_1)(1-x_3)} \tag{2}$$

$$\frac{d\sigma_{gg}}{dx_1 dx_3} = \frac{3\alpha_s}{4\pi}\frac{x_1^3+x_3^3}{(1-x_1)(1-x_3)} \tag{3}$$

where x_i are the final state energy fractions $= 2E_i/\sqrt{S_{dip}}$ of the emitting partons in the dipoles center of mass system. Note that equations 1 and 2 are slightly modified when finite quark masses are taken into account (see description of `MSTA(19)` in section 3.4). It should also be noted that these cross sections correctly reproduces the Altarelli-Parisi splitting kernels in the low $p_\perp$ limit.

The α_s is by default running with the scale taken to be the $p_\perp^2$ of the emission, defined invariantly as

$$p_\perp^2 = S_{dip}(1 - x_1 + \frac{m_1^2-(m_2+m_3)^2}{S_{dip}}) \times (1 - x_3 + \frac{m_3^2-(m_2+m_1)^2}{S_{dip}}) \tag{4}$$

where m_2, the mass of the gluon is always zero.

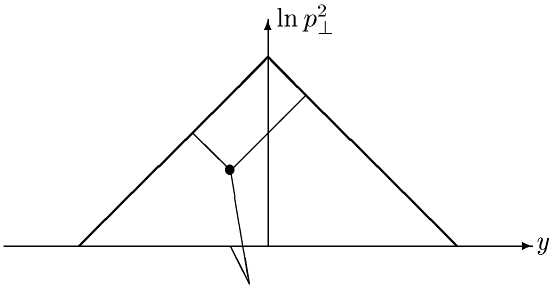

Figure 2: The phase space limits for emission of a first gluon (thick lines) and a second gluon (thin lines).

2.2 Ordering

The $p_\perp^2$ scale is also used for the ordering of the emissions. This means that an emission at a scale $p_{\perp 1}^2$ is performed "before" an emission at a lower scale $p_{\perp 2}^2 < p_{\perp 1}^2$. This is achieved by introducing a Sudakov form factor, giving the probability of emitting a gluon at some scale $p_\perp^2$ according to

$$\frac{dP(p_\perp^2, y)}{dp_\perp^2 dy} = \frac{d\sigma(p_\perp^2, y)}{dp_\perp^2 dy} e^{-\int_{p_\perp^2}^{p_{\perp max}^2} dk_\perp^2 \mathcal{I}(k_\perp^2)} \tag{5}$$

where the first factor is one of the cross sections in equations 1-3 and the second (the Sudakov form factor), with

$$\mathcal{I}(k_\perp^2) = \int_{y_{min}(k_\perp^2)}^{y_{max}(k_\perp^2)} dy' \frac{d\sigma(k_\perp^2, y')}{dk_\perp^2 dy'} \tag{6}$$

corresponds to the probability of not having any emissions at a higher scale.

The transformation of the cross sections in terms of $p_\perp^2$ and y (roughly the rapidity of the emitted gluon) is a very convenient one. With

$$y = \frac{1}{2}\ln\frac{1-x_1}{1-x_3} \tag{7}$$

all three cross sections are well approximated by

$$d\sigma \propto \alpha_S \frac{dp_\perp^2}{p_\perp^2} dy \tag{8}$$

and the available phase space can be approximately represented by the inside of a triangle in the ($\ln(p_\perp^2)$,y) plane as in figure 2.

The CDM can be proven to be a good approximation only in the limit where the emissions are strongly ordered in $p_\perp^2$; $p_{\perp 1}^2 >> p_{\perp 2}^2 >> p_{\perp 3}^2 >>$... But as seen in figure 2 where the thin lines corresponds to the available phase

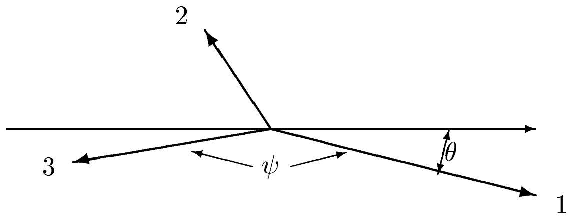

Figure 3: The orientation of a dipole after emission.

space left after emitting one gluon, it is possible to have a second emission at a higher scale than the first one, should the dipoles be considered completely independent. The default procedure in Ariadne is to have strictly ordered emissions so that $p^2_{\perp 1} > p^2_{\perp 2} > p^2_{\perp 3} >$...

2.3 Recoils

The cross sections in equations 1-3 does not completely specify the emission of a gluon. They only give the energy fractions of the partons, while there are two more degrees of freedom to be determined; the azimuth angle ϕ of the emitted gluon and the polar angle θ of parton 1 in figure 3 which determines the distribution of the transverse recoil among the emitting partons. The former is always taken to be evenly distributed between 0 and 2π. For the latter there exist a prescription [16] in the case of a $q\bar{q}$ dipole derived from spin considerations, where a correct treatment is achieved by letting one of the quarks retain its direction after the emission with a probability proportional to the square of its energy. For other kinds of dipoles no such prescription exist. Instead the CDM postulates that the transverse recoils are distributed in such a way that "the disturbance of the colour flow in neighboring dipoles is minimized". In Ariadne this is implemented so that for a qg dipole the gluon always retains its direction (as there is no neighboring dipole on the quark side to disturb) while for a gg dipole the recoil is distributed according to

$$\theta = \frac{x_3^2}{x_1^2 + x_3^2}(\pi - \psi) \qquad (9)$$

where ψ is the angle between partons 1 and 3.

2.4 $Q\bar{Q}$ production

The process of splitting a gluon into a $q\bar{q}$ pair is not straight forward to introduce into the dipole picture.

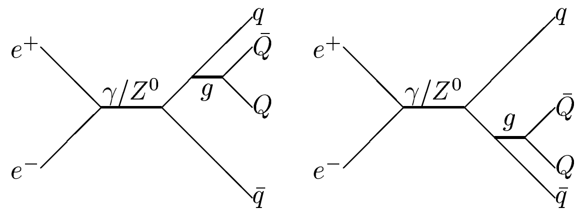

Figure 4: The Feynman diagrams relevant for $e^+e^- \to q\bar{q}Q\bar{Q}$.

Looking at the cross section for the process $e^+e^- \to q\bar{q}Q\bar{Q}$, corresponding to the diagrams in figure 4, in the limit of small $p_\perp^2$ of the gluon and small invariant mass of the produced $Q\bar{Q}$ pair, it can be shown that it factorizes into two parts; one corresponding to the process $e^+e^- \to qg\bar{q}$ and the other the process $qg\bar{q} \to q\bar{q}Q\bar{Q}$. The first part just gives the cross section in equation 1. In the CDM the latter is divided into two equal contributions from each of the two dipoles qg and $g\bar{q}$. This gives the cross section for splitting a gluon in a qg dipole into a $q\bar{q}$ pair:

$$\frac{d\sigma_{qg \to qQ\bar{Q}}}{dx_1 dx_3} = \frac{3\alpha_s}{8\pi}\frac{(1-x_2)^2 + (1-x_3)^2}{1-x_1} \qquad (10)$$

where 2 and 3 denotes the Q and $\bar{Q}$. Again this correctly reproduces the Altarelli-Parisi splitting kernels in the low $p_\perp$ limit.

Introducing ordering we get the probability for splitting a gluon in a qg dipole at a phase space point $(p_\perp^2, y)$ as

$$\frac{dP_{qQ\bar{Q}}(p_\perp^2, y)}{dp_\perp^2 dy} = \frac{d\sigma_{qQ\bar{Q}}(p_\perp^2, y)}{dp_\perp^2 dy} e^{-\int_{p_\perp^2}^{p^2_{\perp,max}} dk_\perp^2 (\mathcal{I}_{qQ\bar{Q}}(k_\perp^2) + \mathcal{I}_{qgg}(k_\perp^2))} \qquad (11)$$

$$\mathcal{I}_i(k_\perp^2) = \int_{y_{min}(k_\perp^2)}^{y_{max}(k_\perp^2)} dy' \frac{d\sigma_i(k_\perp^2, y')}{dk_\perp^2 dy'} \qquad (12)$$

Hence there is a competition between this process and the possibility to instead emit another gluon. It turns out [5] that the choice of $p_\perp^2$ as ordering variable, together with the handling of recoils in Ariadne [17], tends to favor the $Q\bar{Q}$ production more than in conventional parton cascades (e.g. [9]) where the ordering is typically in Q^2.

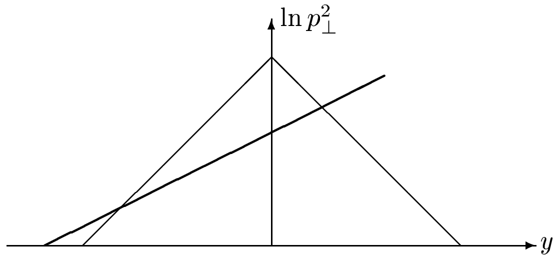

Figure 5: The available phase space for gluon emission in DIS. Positive rapidity corresponds to the direction of the struck quark.

2.5 Radiation from extended sources

In Deep Inelastic Scattering (DIS) of leptons on hadrons, the CDM does not divide the QCD cascade into a initial and final state as conventional parton cascades do. Instead it assumes that all radiation can be described as radiation from the colour dipole formed between the struck quark and the hadron remnant.

In this way the situation is very similar to the e^+e^- case with one important modification. In e^+e^- both the q and $\bar{q}$ can be considered point-like, but for DIS only the struck quark is point-like while the hadron remnant is an extended object. It is a well known fact that emissions of small wavelengths from an extended antenna is suppressed and that for an antenna of transverse size l, effectively only a fraction proportional to the emitted wavelength $\lambda \propto 1/p_\perp$ is participating in the emission. In the CDM this is taken into account [4] by allowing only a fraction

$$a = (\frac{\mu}{p_\perp})^\alpha \tag{13}$$

of the hadron remnants light-cone momentum P_- to take part in an emission with a transverse momentum $p_\perp$, where μ describes the inverse size and α the dimension of the remnant. This means that the available phase space is reduced as compared with the e^+e^- case in the way described in figure 5, leading to a suppression of the radiation in the target region.

An alternative to this sharp cut-off in phase-space is to allow a fraction $a'$of the hadron remnants light-cone momenta to take part in the emission with a probability

$$\frac{dP(a')}{da'} = \frac{b}{a}\frac{(\frac{a'}{a})^{b-1}}{(1+(\frac{a'}{a})^b)^2} \tag{14}$$

this corresponds to an exponential suppression in $\kappa = \ln p_\perp^2$ of emissions above the thick line in figure 5. (see description of `PARA(25)` and `MSTA(25)` in section 3.4.)

2.6 Recoil gluons

Also the distribution of transverse recoil is different in DIS as compared to the e^+e^- case. Since only a fraction of the hadron remnant takes part in the emission, only that part should be allowed to recoil. This is realized by introducing an extra "recoil" gluon taking a fraction a of the remnants energy. The recoil is then distributed between this recoil gluon and the other emitting parton.

This would mean that in the first emission of a gluon from the dipole between the struck quark and the remnant, the quark should take the full recoil according to the prescription for a qg dipole in section 2.3, leaving the recoil gluon collinear with the remnant. It can, however, be shown [18] that the emission of the recoil gluon is only consistent with ordinary gluon radiation if it is given a transverse recoil according to

$$p_{\perp 3}^2 e^{-y_3} = p_{\perp 2}^2 e^{-y_2} \tag{15}$$

where 2 denotes the emitted gluon and 3 the recoil gluon. Both this "recoil strategy" and the one implied by the rules in section 2.3 are implemented in Ariadne. By default however the strategy is given by equation 15.

In both cases the recoil gluon is submitted to $p_\perp^2$ ordering, i.e. it is not emitted if a normal emission of a gluon between the remnant and gluon 2 gives a larger $p_\perp^2$.

2.7 Photon-Gluon Fusion in DIS

In e^+e^- annihilation the first emission in the dipole cascade is governed by the exact $\mathcal{O}(\alpha_S)$ matrix element, a situation which is much better than in other QCD shower programs, where a more or less complicated matching procedure is introduced to reproduce the form of the matrix element.

In DIS the situation is more complicated. Although the gluon emission in the CDM is close to the form of the exact matrix element, the so-called boson-gluon fusion (BGF) process is not described at all. Therefore a matching procedure has been introduced for the first emission in a DIS event [19]. In this procedure, the initial dipole between the struck quark and the hadron remnant can either emit a gluon according to the $\mathcal{O}(\alpha_S)$ matrix element (modified to use the phase space suppression in section 2.5) or to "emit"the antipartner of the struck quark according to the BGF matrix element.

The selection of process is, as usual, made by the Sudakov form-factor prescription. In the case of BGF, there is then two dipoles, connecting each of the struck quarks with the hadron remnant, which continue to radiate independently.

An older procedure where the BGF procedure is generated separately by the LEPTO program is also included.

If in the first step a gluon is emitted, the antipartner of the struck quark can be emitted in a later stage by using the normal initial state $g \to q\bar{q}$ splitting functions and the input structure functions in a way similar to a conventional initial state parton shower. Again the Sudakov formfactor with $m_\perp^2$ as ordering variable is used to choose between this and emitting gluons from either of the existing dipoles, and if a $g \to q\bar{q}$ splitting is chosen a new dipole is formed between the emitted antipartner and the corresponding part of the hadron remnant.

2.8 Remnant treatment and rapidity gaps

The gluon emission in DIS is sensitive to the way the dipole connects the struck quark and the proton remnant. If only a fraction of the remnant is connected by the dipole, the total energy available for gluon radiation is reduced.

In [4] a procedure was introduced, in which a clear distinction was made between the case where the virtual photon couples to a valence quark and the case where it couples to an (anti) sea-quark. In the former case the colour-$\bar{3}$ charge is assumed to be carried by the whole remnant, which is treated as a simple diquark. In the case of a struck sea-quark, however, the remnant is not a simple diquark, but rather it contains all original flavour quantum numbers of the proton, plus the antiflavour corresponding to the struck sea-quark. It is therefore assumed that the dipole connects the stuck quark with a (valence) diquark leaving a colour-less meson containing the remaining valens quark and the anti-sea-quark. (In the case of a struck anti-sea-quark the dipole connects a valence quark and leaves a baryon.) The hadron is given a fraction z of the original proton momentum, where z is given by the Lund symmetric fragmentation function [20].

In the sea-quark case the available phase space for gluon emission is thus reduced either by scaling the total dipole mass with $W \to W\sqrt{1-z}$ or, equivalently, assuming that the transverse extension of the remnant is unchanged, by reducing $\mu \to \mu(1-z)$.

In the case of BGF there is also a need to split the remnant into two parts; and a diquark a quark connecting the struck quark and antiquark respectively. Here the strategy is to have the same distribution between the valence quark and diquark as the above procedure gives for the meson and baryon respectively.

Since the Lund string fragmentation governs the kinematics of the meson and baryon in the remnant treatment of sea-quark interaction events, large rapidity gaps (corresponding to z close to 1) will be very rare. It could however be assumed that the struck quark in some cases belonged to a colourless state taking only a small fraction of the proton momentum. This state could be a meson or, which will be assumed here, a pomeron produced diffractively by the proton in the spirit of [21] and the POMPYT program [22].

Assuming a "flux" of pomerons $f_{I\!P}^p(x_{I\!P}, t)$ taking a fraction $x_{I\!P}$ of the proton momentum at some small momentum transfer t, with the density of quarks $f_q^{I\!P}(z, Q^2)$ within the pomeron, the "pomeron-induced" part of the total parton density $f_q^p(x, Q^2)$ in the proton is given by a simple convolution:

$$f_q^{p(I\!P)}(x, Q^2) = \int dt \int dz \int dx_{I\!P}\; f_{I\!P}^p(x_{I\!P}, t) f_q^{I\!P}(z, Q^2)\delta(zx_{I\!P} - x). \tag{16}$$

For each event there is then a probability $f_q^{p(I\!P)}(x, Q^2)/f_q^p(x, Q^2) < 1$ that the struck quark be part of a pomeron, in which case the $x_{I\!P}$ and t are generated according to the distributions $f_{I\!P}^p$ and $f_q^{I\!P}$. This results in a dipole between the struck quark and the pomeron remnant with an invariant mass squared $M_x^2 = x_{I\!P}W^2$, which is small since $f_{I\!P}^p$ typically is proportional to $1/x_{I\!P}$.

The pomeron remnant in this scenario is always the antipartner of the struck quark, which is assumed to have a transverse extension $1/\mu_{I\!P}$.

Assuming that a pomeron, which by definition is a "gluonic" object, contains quarks might seem strange at first, but since the virtual photon cannot couple directly to a gluon, it always makes sense to define a quark density $f_q^{I\!P}$. It can of course be argued that the BGF diagram dominates and that the remnant therefore should be an extended gluon connecting both the quark and the corresponding antiquark with one dipole each. In the following, however, it is assumed that the energy sharing between the quark and antiquark in this case is sufficiently asymmetric, due to the $1/(z_q(1-z_q))$ pole in the cross section, that one of them always can be considered a part of the remnant. This should be a good approximation, as long as M_x is not too large.

2.9 The Linked Dipole Monte Carlo

As of release 4.10, the Ariadne distribution also includes the Linked Dipole Chain Monte Carlo (LDCMC) [23] for generating deeply inelastic electron-hadron collisions according to the Linked Dipole Chain model (LDC) [24,25]. By default, the LDCMC is disabled, but when linking with the LDCMC library, it can be called from Ariadne by simply setting the switch `MSTA(32)=-32`. If so, LDCMC will, for each requested event, generate an initial-state chain of dipoles according to the LDC model. Ariadne will then perform the final-state dipole radiation from these dipoles and return the produced event in the `/LUJETS/` common block in the usual way. A brief manual for the LDCMC program is included in the Ariadne distribution.

2.10 Photon radiation

A $q\bar{q}$ pair is, besides a colour dipole, of course also an Electro-Magnetic (EM) dipole, and the CDM can in a natural way be extended to also describe bremsstrahlung of photons from quarks [26]. The cross section for emitting a photon from a $q\bar{q}$ pair is given by

$$\frac{d\sigma}{dx_1 dx_3} = \frac{\alpha_{EM}}{2\pi} e_q^2 \frac{x_1^2 + x_3^2}{(1-x_1)(1-x_3)} \tag{17}$$

which is the same as equation 1 substituting α_s with α_{EM} and the colour factor 4/3 with the square of the quark charge. Introducing ordering in $p_\perp^2$ there will be a competition between photon and gluon emission from a $q\bar{q}$ dipole giving the probability to emit a photon

$$\frac{dP_\gamma(p_\perp^2, y)}{dp_\perp^2 dy} = \frac{d\sigma_\gamma(p_\perp^2, y)}{dp_\perp^2 dy} e^{-\int_{p_\perp^2}^{p_{\perp,max}^2} dk_\perp^2 (\mathcal{I}_\gamma(k_\perp^2) + \mathcal{I}_g(k_\perp^2))} \tag{18}$$

where $\mathcal{I}_i$ is given by equation 12.

If at first a gluon is emitted from the $q\bar{q}$ dipole, which is very likely due to the relative smallness of α_{EM}, there will be many competing processes in the next stage; emitting a photon from the EM dipole between the q and $\bar{q}$, emitting a gluon from each of the two colour dipoles qg and $g\bar{q}$ and splitting the gluon into a new $Q\bar{Q}$ pair.

As long as only gluons and photons are emitted, the CDM can treat EM bremsstrahlung in a well defined way using $p_\perp^2$ ordering. If, however, an additional $q\bar{q}$ pair is created in the cascade, the picture becomes very complicated. This would mean that further colour dipole emission would have to compete with EM quadrupole emission for which it is not possible to define a $p_\perp^2$ in the same way. Instead it may be argued that when an additional $q\bar{q}$ pair is created, the original EM dipole is screened and in Ariadne by default, the EM bremsstrahlung is simply switched off. Optionally it is possible to allow the original EM dipole to continue radiating. In any case the photons that are emitted at this late stage in the cascade are usually drowned in the background of photons from hadronic decays in the jets.

It should be noted that the Ariadne program only describes final state photon radiation in e^+e^- collisions. Initial state photon radiation must be handled by the program performing the hard interaction, and no interferences between initial and final state is possible.

2.11 Hadron-hadron collisions

In principle it is straight forward to use the formalism for DIS also to describe dipole radiation in hadron-hadron collisions. After the hard interaction the formed dipoles should be allowed to radiate, treating the partons that have taken part in the hard sub-process as point-like and all others as extended. When the hard sub-process is a QCD interaction, the $p_\perp^2$ of this interaction must limit the subsequent emissions to avoid double counting.

One problem arises in the case of Drell-Yan production. In the case of e.g. $q\bar{q} \to W$ there will be a dipole between the two hadron remnants which can radiate. Although the dipole in other cases describes correctly the initial state QCD-radiation, in this case it does not, as the W is already "disconnected" from the radiating dipole. This makes it impossible to give it any transverse momenta from the initial state which has been observed in e.g. pp collisions [27,28].

In ref. [29], a method was introduced to transfer the recoil of an emitted gluon to the W (see MSTA(22)). In initial state $g \to q\bar{q}$ radiation the $p_\perp$ of the emitted quark is always transferred to the struck system.

2.12 Heavy Quarkonium production

Ariadne includes a simple model for fragmentation production of heavy quarkonium as described in refs. [30] and [31]. In each step in the cascade a gluon or heavy quark can, in addition to the normal QCD radiation, be split into a heavy quarkonium and a spectator eg. $g \to \mathcal{O} + g$ and $Q \to \mathcal{O} + Q$. In the case of gluon splitting, the process is treated as in the $g \to Q\bar{Q}$ case where the splitting kernel is divided into two equal parts assigned to each of the connecting dipoles.

So far only three splitting kernels are implemented:

- Colour Octet gluon fragmentation, where a gluon couples to a quarkonium in an colour octet state which subsequently emits a soft gluon to form a colour singlet quarkonium. In this implementation the emission of the soft gluon is done immediately and continues to participate in the subsequent cascade, while the quarkonium is untouched by the continuing cascade. The soft gluon is given a fraction $\epsilon \propto v^2$ of the original gluons momentum, where v is the scaled velocity of the heavy quark inside the quarkonium.

- Colour Singlet gluon fragmentation, where a gluon perturbatively splits into a colour singlet quarkonium and a number of gluons. If more than one gluon is produced, the n-gluon system is first treated as one heavy gluon with the mass properly distributed, and this gluon is then decayed isotripocally into the n gluons.

- Colour Singlet heavy quark fragmentation, $Q \to \mathcal{O} + Q$, where the whole splitting kernel is assigned to the only dipole connected to the heavy quark.

To be able to study different quarkonium mechanisms separately, a subroutine ARADDO has been introduced to specify which mechanisms to use. This routine is described in section 3.1 below.

3 Program Components

The program consists of a large number of routines performing different, well defined, operations on an internal dipole oriented event record, consisting of the three common blocks /ARPART/, /ARDIPS/ and /ARSTRS/. Externally the program however use the event record of JETSET in the /LUJETS/ common block.

Most of these routines are only used internally by the program and are of no real interest for the ordinary user. They are however all described briefly in appendix B, mainly to give the user some idea of how the program works.

In short Ariadne works as follows: After it has been initialized with the ARINIT subroutine, it can be made to act on partonic states in the /LUJETS/ common block by calling AREXEC. AREXEC makes some initial modifications to /LUJETS/ depending on which program it is initialized to run with and then calls ARPARS which performs the translation to the internal event record. In ARCASC the main loop over emissions is found, where first ARGPT2 is called to generate a $p_\perp^2$ for a possible emission from each dipole. The dipole with largest $p_\perp^2$ is then allowed to radiate by a call to AREMIT. The loop is continued until all generated $p_\perp^2$:s are below the cut-off after which a call to ARDUMP translates the formed parton state back to /LUJETS/.

3.1 The main routines

The following routines are the ones normally called by the user:

- `SUBROUTINE ARINIT(MODE)`
  Before Ariadne can be used, it has to be initialized with ARINIT. ARINIT takes one argument which is a character string indicating which program Ariadne is used with; 'JETSET', 'LEPTO', 'PYTHIA' or by itself - 'ARIADNE'.

- `SUBROUTINE AREXEC`
  This is the main routine in Ariadne. Given a partonic state in /LUJETS/ it administers the dipole radiation according to the options and parameters set in /ARDAT1/. AREXEC assumes that the partonic state has been produced by the program for which Ariadne has been initialized, and reformats the event-record accordingly.

- `SUBROUTINE ARPRDA`
  Prints out the values of the parameters and switches used by Ariadne.

- `SUBROUTINE ARTUNE(SET)`
  Sets the parameters in Ariadne to the values tuned by different experimental collaborations. The argument is a character string and should be set to `'ALEPH'`, `'DELPHI'` or `'OPAL'` to use the tuning of references [32], [33] and [6] respectively. Giving the argument `'EMC'` will set parameters relevant for e^+e^--annihilation in the same way as for `'DELPHI'` but in addition parameters relevant for lepto-production will be set to fit EMC data. These settings have changed significantly between version 4.04 and 4.06. To retain the old tunings, the argument `4.04` or `4.05` should be given. Note that `ARTUNE` also changes some parameters and switches in JETSET (`PARJ(21)`, `PARJ(41)`, `PARJ(42)`, `PARJ(54)`, `PARJ(55)`and `MSTJ(11)`), and in LEPTO (`PARL(3)`). See also description of `MSTA(3)` below.

- `SUBROUTINE ARTEST(IPRINT)`
  A test program to check that Ariadne has been installed properly, disguised as a subroutine (see appendix A.3).

- `SUBROUTINE ARUPOM(KFT,KFSTR,X,XQ2, XPOM,TPOM,KFTF,KFPR,XFP,XFPOM)`
  This is a dummy routine which can be used to implement a user defined pomeron distribution and structure function. Given a target code `KFT`, the flavour `KFSTR` of the struck parton and the `X` and Q^2 (`XQ2`) of the incomming parton, the routine should return a generated momentum fraction `XPOM` and virtuality `TPOM` of the pomeron, the code `KFTF` of the diffracted target, the flavour `KFPR` of the pomeron remnant and, given `X` the total structure function `XFP` for the flavour `KFSTRF` the pomeron induced part of this function `XFPOM`.

- `ARADDO(IPO,IFLO,MEO,Q2G,P1,P2,P3,P4,P5)`
  This routine is used to switch on different mechanisms for fragmentation production of heavy quarkonia. It enables the fragmentation of a parton `IPO` into a quarkonium `IFLO` according to the mechanism `MEO` using parameters `P1` to `P5`. If `Q2G` is true, then more than one gluon is allowed in a splitting. The following mechanisms are included:

  - Colour Singlet $g \to \mathcal{O}(^3S_1^{(1)}) + 2g$. This corresponds to `MEO=1` and the parameter `P1` is the colour singlet parameter $|R(0)|^2$. If `Q2G` is false, the two gluons produced is approximated by one carrying the sum of their momenta.
  - Colour Singlet $Q \to \mathcal{O}(^3S_1^{(1)}) + Q$. This corresponds to `MEO=41` and the parameter `P1` is the colour singlet parameter $|R(0)|^2$.
  - Colour Octet $g \to \mathcal{O}(^3S_1^{(8)})$. This corresponds to `MEO=3` and the parameter `P1` is the colour octet matrix element and `P2` is the scaled velocity of the heavy quarks in the quarkonium.

  An example on how to use this subroutine is given in the sample program `ptest.f` included in the distribution.

3.2 A jet clustering routine

In addition to the generation of dipole emission, Ariadne also provides a routine for jet-clustering called `ARCLUS` [34]. It implements a CDM inspired jet algorithm, DICLUS, which is very different from conventional algorithms.

Conventional algorithms are typically based on some measure defining the distance between two jets. This measure can be the invariant mass as in the JADE algorithm [35] or some relative $p_\perp$ as in the LUCLUS algorithm [9]. The procedure would then be to find the two jets which are closest together according to this measure, replacing these with a new jet by summing their momenta. This would then be repeated until no two jets are closer together than some cut-off.

`DICLUS` is different in the sense that it looks at all combinations of *three* jets, looking at the invariant $p_\perp^2$ of one w.r.t. the two others. The combination which gives the smallest $p_\perp^2$ is then selected and these three jets are then clustered together into two, where the orientation of the two new jets are determined by equation 9. An inverse dipole emission if you like. The procedure is repeated until no $p_\perp^2$ is below a cut-off.

The algorithm is obviously inspired by the CDM but it also fits well into the Lund Sting Fragmentation [36, 37] picture where a hadron is not produced by one parton but rather in the string between two partons.

The algorithm is invoked by `CALL ARCLUS(NJET)` and is used in the same way as the `LUCLUS` algorithm and also uses some of the switches in JETSET's `/LUDAT1/` common block for compatibility. The cut-off in invariant $p_\perp^2$ is given by PARA(31) in /ARDAT1/. With MSTU(47) one can require a minimum number of jets to be reconstructed. MSTU(41) determines which particles are used if MSTU(48)=0 otherwise the clusters already in /LUJETS/ from an earlier cluster search are used.

After the call, NJET is equal to the number of jets found, or negative to indicate that something went wrong. The energy and momentum of the jets are stored in positions N+1 through N+MSTU(3) in /LUJETS/. In addition, after the call MSTU(62) will be set to the number and PARU(61) to the total invariant mass of particles used in the analysis, and PARU(63) will be equal to the smalles $p_\perp$ among the final jets.

There is also a stand-alone set of subroutines which can be used independently of JETSET common blocks etc. The main one of these are called DICLUS and it is used as followed:

```
SUBROUTINE DICLUS(MODE,NDIM,NP,PP,IC1,IC2,
PTCUT,PTNEXT,NJMAX,NJMIN,NJ,PJ,IERR)
```

- MODE The recoil strategy. If 0, both final jets will receive transverse recoil according to equation 9. If 1 only the jet closest to the clustered one receives recoil.
- NDIM The dimension of the PP and PJ vectors.
- NP Number of entries in the PP vector.
- PP(NDIM,*) Vector of 3- or 4- momenta of particles to be clustered.
- IC1 Pseudo particle info. IC1/10 is the index of a pseudo particle in the PP vector which is not allowed to be clustered. If MOD(IC1,10)>0, the direction of the pseudo particle must mot change. If MOD(IC1,10)>1, the magnitude of the pseudo particle must mot change.
- IC2 Info for a second pseudo particle as for IC1.
- PTCUT Maximum invariant $p_\perp$ allowed when clustering three particles into two.
- NJMAX Maximum number of jets allowed.
- NJMIN Minimum number of jets is given by ABS(NJMIN). If NJMIN is negative then contonue from a previous clustering.
- PTNEXT Minimum invariant $p_\perp$ among the remaining jets.
- NJ Number of jets produced.
- PJ(NDIM,*) Vector of 3- or 4- momenta of jets produced.
- IERR Indicates that an error has occurred if non-zero.

3.3 Main Common Blocks

In appendix B a full list and short description is given for all common blocks used in Ariadne, here only the main ones are described.

COMMON /ARDAT1/ PARA(40),MSTA(40)
The parameters and switches used by Ariadne. See below for a full description.

COMMON /ARDAT2/ PQMAS(10)
The quark masses used by Ariadne. These are by default set by ARINIT to the values of PARF(101) - PARF(108) in the /LUDAT2/ common block of JETSET. (See switch MSTA(24) below for more details.)

COMMON /ARPOPA/
TOTSIG,PPOW,CA(3),PB(3),CF(0:6),XA(0:6),NB(0:6)
The built in pomeron structure functions as defined below (see MSTA(34)). TOTSIG(D=2.3) is the parameter $\sigma_{\rm tot}$, PPOW (D=2) is the exponent p, CA(D=6.38,0.424,0.0) are the coefficients d_j and PB(D=8,3,0) the exponents e_j in the expression for the pomeron flux. CF(D=0,1.5,1.5,0,0,0,0) the coefficient c_i, XA(D=1,1,1,1,1,1,1) the exponent a_i and NB(D=1,1,1,1,1,1,1) the exponent b_i for flavour i in the expression for the parton density in the pomeron.

COMMON /LUJETS/ N,K(4000,5),P(4000,5),V(4000,5)
This is the standard JETSET event record used to communicate with the Ariadne program. When run together with the JETSET, LEPTO or PYTHIA programs AREXEC will automatically handle the encoding of the partonic states which should be treated by Ariadne. When Ariadne is used by itself the user must ensure that these partonic states are correctly encoded. Ariadne will then perform dipole radiation for all un-decayed strings of partons in /LUJETS/ where all partons have the code K(I,1)=2 except for the last one in a string which should have K(I,1)=1 (the standard JETSET encoding). In addition, all partons with K(I,4) between 1 and 3 will be treated extended with the μ given by PARA(10+K(I,4)). After the call to AREXEC the initial partons will have K(I,1)=12 or K(I,1)=11 to indicate that they have decayed, and K(I,4)

and `K(I,5)` will point to the first and last parton in the cascaded string. The partons in the produced string will be properly encoded for a subsequent call to `LUEXEC` for fragmentation. In addition `K(5,I)` will give information about in which order the partons were emitted in the cascade. A gluon produced in emission number `NO` will have `K(I,5)=NO`, a recoil gluon from the same emission will have `K(I,5)=-NO`. If in emission `NO` a gluon, previously produced in emission `NOG`, is split into a $q\bar{q}$ pair, both quarks will have `K(I,5)=NOG*1000+NO`.

3.4 Parameters and switches

The following parameters are used by Ariadne:

- `PARA(1)` (Default value = 0.22 GeV) The Λ_{QCD} used in the running coupling α_s.
- `PARA(2)` (D = 0.2) Value of constant α_s for `MSTA(12) = 0`.
- `PARA(3)` (D = 0.6 GeV) The cut-off in invariant $p_\perp$ for emissions from colour dipoles.
- `PARA(4)` (D = 1/137) Value of electro-weak coupling constant α_{EM} used for photon emissions.
- `PARA(5)` (D = 0.6 GeV) The cut-off in invariant $p_\perp$ for emissions from electro-magnetic dipoles.
- `PARA(6)` (D = -1.0 GeV^2) If larger than zero this gives the maximum allowed invariant $p_\perp^2$, otherwise the maximum is given by phase space limits.
- `PARA(7-9)` Not used.
- `PARA(10)` (D =1.0) Power in soft suppression - α (the dimensionality of the extended source).
- `PARA(11)` (D = 0.6 GeV) Soft suppression parameter μ in LEPTO, or in PYTHIA for remnant 1, or for partons with `K(I,4)=1`.
- `PARA(12)` (D = 0.6 GeV) Soft suppression parameter μ in PYTHIA for remnant 2, or for partons with `K(I,4)=2`.
- `PARA(13)` (D = 0.6 GeV) Soft suppression parameter μ for partons with `K(I,4)=3`.
- `PARA(14)` (D = 1.0) Factor to multiply $p_\perp$ of resolved photon in PYTHIA to get soft suppresion parameter μ
- `PARA(15)` (D = 1.0) Power in soft suppression - α for resolved photon in PYTHIA
- `PARA(16)` (D = -1.0) Mean of gaussian distributed intrinsic $p_\perp$ of struck quark in pomeron (if less than 0 this is instead taken from `PARL(3)` in LEPTO or `PARP(91)` in PYTHIA
- `PARA(17)` (D = 2.0) Maximum value of intrinsic $p_\perp$ of struck quark in pomeron if given by `PARA(16)`
- `PARA(18)` (D = 1.0) Maximum fraction of pomeron induced structure function of total structure function.
- `PARA(19)` (D=0.0001) Minimum value of structure function in denominator in initial state $g \to q\bar{q}$.
- `PARA(20)` (D = 0.0) For `MSTA(32) = 0` when used together with LEPTO 6.1 - the minimum value of $p_\perp^2/Q^2$ of a $q\bar{q}$ pair in a boson-gluon fusion event. If below, the event is treated as a sea-quark interaction.
- `PARA(21)` (D = 1.0) Factor multiplying a virtuality Q^2 before comparing with an invariant $p_\perp^2$.
- `PARA(22-24)` Not used.
- `PARA(25)` (D = 2.0) For `MSTA(25)` > 0 gives the power $\beta = 1/b$ in equation 14.
- `PARA(26)` (D = 9.0) The number of differently coloured dipoles possible.
- `PARA(27)` (D = 0.6) The square of this is the mean value of the primordial $p_\perp^2$ for `MSTA(37)` > 0.
- `PARA(28)` (D = 0.0) If larger than zero, this is the minimum allowed gluon energy. If less than zero, the absolute value is the maximum allowed gluon energy.
- `PARA(29-30)` Not used.
- `PARA(31)` (D = 25.0 GeV^2) The maximum invariant $p_\perp^2$ for clustering three jets into two in `ARCLUS`.
- `PARA(32-38)` Not used.
- `PARA(39)` (D = 0.001) Tolerance factor for momentum conservation. If any component of the total energy and momentum for a partonic state has changed more than this factor times the total invariant mass of the state during the cascade, a warning is produced.
- `PARA(40)` (D = 10^{32}) Maximum floating point number allowed by the machine on which Ariadne is run.

The following switches are used by Ariadne:

- `MSTA(1)` The mode set by `ARINIT` for correct treatment of incoming partons.
  0 No special treatment.
  1 The incoming partons are treated as if produced by JETSET.
  2 The incoming partons are treated as if produced by PYTHIA.
  3 The incoming partons are treated as if produced by LEPTO.
- `MSTA(2)` Flag set by `ARINIT` to indicate that initialization has been done.
- `MSTA(3)` (D=1) Setting of parameters in Ariadne, JETSET, PYTHIA and LEPTO to suitable values in `ARINIT`.
  0 Off.
  1 On.
- `MSTA(4)` Number of calls to `AREXEC` so far.
- `MSTA(5)` (D=0) Perform fragmentation at the end of each call to `AREXEC`. When running with JETSET, LEPTO or PYTHIA, this switch is set by `ARINIT` to the value of the corresponding switch in these programs.
  0 Off.
  1 On.
- `MSTA(6)` (D=-1) If larger than zero, sets the maximum number of emissions allowed per string in a `AREXEC` call.
- `MSTA(7)` (D=6) File number for output from Ariadne. (Is set by `ARINIT` to the value of `MSTU(11)` in the `/LUDAT1/` common block of JETSET.)
- `MSTA(8)` (D=6) File number for error messages and warnings from Ariadne.
- `MSTA(9)` (D=1) Determines how carefully Ariadne checks momentum conservation etc.
  0 No checking of momentum conservation. Only serious errors are reported by Ariadne.
  1 Momentum conservation is checked after each call to `AREXEC`.
  2 Momentum conservation is checked after each emission.
  3 As for 2 but in addition the current parton state is copied into the `/LUJETS/` event record after each emission.
- `MSTA(10)` (D=5) Maximum number of warnings (of each kind) issued by Ariadne.
- `MSTA(11)` (D=0) Phase space restrictions; The maximum $p_\perp^2$ of an emission is set to the $p_\perp^2$ of the last emission for:
  0 all emissions
  1 all emissions from colour dipoles
  2 only for gluon and photon emissions
  3 only for gluon emissions
  4 no restriction.
- `MSTA(12)` (D=1) Treatment of α_s.
  0 Constant α_s given by `PARA(2)`.
  1 Running $\alpha_s = 12\pi/(33-2n_f)\ln p_\perp^2/\lambda^2$.
- `MSTA(13)` If non-zero, a warning was issued in the last call to `AREXEC` or `ARCLUS`. (See description of subroutine `ARERRM` in appendix B)
- `MSTA(14)` (D=1) Setting of the maximum invariant $p_\perp^2$ to the minimum $p_\perp^2$ of all incoming gluons in a string.
  0 Off.
  1 On. Except when run together with PYTHIA where instead the limit is set to the $p_\perp^2$ of the hard interaction for the relevant sub-process.
  2 On also for PYTHIA.
- `MSTA(15)` (D=5) Maximum number of flavors allowed in $q\bar{q}$ emissions.
- `MSTA(16)` (D=2) Recoil Strategy:
  0 Use equation 9 for all emissions.
  1 As 0, but point-like quarks take full recoil.
  2 As 1, but also extended quarks takes full recoil if $a > 1$.
- `MSTA(17)` (D=3) Treatment of recoil gluons:
  0 No recoil gluons are emitted.
  1 Emit recoil gluon except if other dipole end is a point-like quark for `MSTA(16)=1`.
  2 Emit recoil gluon according to equation 15 except if other dipole end is a point-like quark for `MSTA(16)=1`.
  3 Always emit recoil gluon according to equation 15.

- `MSTA(18)` (D=3) $p_\perp$-ordering of recoil gluons:
  - 0 Off.
  - 1 On and require $p_\perp > \max(\mu, p_{\perp cut})$
  - 2 as 1 but allow $p_\perp < \mu$
  - 3 as 2 but allow $p_\perp < p_{\perp cut}$
- `MSTA(19)` (D=2) Treatment of emissions from heavy quarks.
  - 0 Simple treatment, changing the denominator in equations 1, 2 and 17 to

$$(1 - x_1 + \frac{m_1^2 - m_3^2}{S_{dip}})(1 - x_3 + \frac{m_3^2 - m_1^2}{S_{dip}}) \quad (19)$$

  - 1 A more elaborate treatment taking into account the "dead-cone" effect in reference [38].
  - 2 As for 1 but also use $\max(p_\perp^2, Q^2)$ in argument for α_s in DIS
- `MSTA(20)` (D=0) Electro-magnetic dipole radiation:
  - 0 Off.
  - 1 On.
  - 2 On, but turned off at the first occurrence of $q\bar{q}$ emission in a string (c.f. section 2.10).
- `MSTA(21)` Not used.
- `MSTA(22)` (D=1)Transfer of recoils in Drell-Yan processes
  - 0 Off.
  - 1 On.
  - 2 On but no transfer if $a > 1$
  - 3 On but modified phase space.
  - <0 As for > 0 but only transfer recoil from recoil gluons
- `MSTA(23)` (D=1) Matrix element for $g \to q\bar{q}$.
  - 0 Use wrong matrix element for backward compatibility with version 4.01-4.04.
  - 1 Correct matrix element with $p_\perp^2$ as ordering variable.
  - 2 Correct matrix element with $m_\perp^2$ as ordering variable.
- `MSTA(24)` (D=2) Quark masses to be used in $q\bar{q}$ emissions
  - 0 as specified in `PQMAS(1-8)` in `/ARDAT2/`.
  - 1 "bare" quark masses as specified in `PMAS(1-8)` in `/LUDAT2/`.
  - 2 "constituent" quark masses as specified in `PARF (101 - 108)` in `/LUDAT2/`.
- `MSTA(25)` (D=1) Phase space treatment for extended dipoles
  - 0 using a restricted phase space according to equation 13.
  - 1 using full phase space with suppression according to equation 14.
  - 2 as 1 but with $p_\perp^2$ defined as

$$S_{dip}\frac{(1 - x_1)(1 - x_3)}{1 - x_2} \quad (20)$$

  - 4 as 1 but use real $p_\perp$ of gluon w.r.t. remnant
  - 5 as 1 but use real $p_\perp$ of gluon w.r.t. struck quark
- `MSTA(26)` Not used.
- `MSTA(27)` (D=0) Normalization of pomeron structure function
  - 0 Off.
  - 1 On.
- `MSTA(28)` (D=0) Extra phase space limits on final-state $g \to q\bar{q}$
  - 0 Normal $p_\perp^2$-ordering according to `MSTA(11)`
  - 1 Additional requirement that $m_{Q\bar{Q}} < p_{\perp g}$
  - 2 Additional requirement that $m_{Q\bar{Q}} - 2m_Q < p_{\perp g}$
  - <0 as > 0 but do not limit $p_{\perp Q\bar{Q}}$ in `MSTA(11)`.
- `MSTA(29)` (D=0) Allow for gluonic "rings" in event record
  - 0 Yes.
  - 1 No.
- `MSTA(30)` (D=3) Options when running with LEPTO
  - 0 Struck quark point-like, remnant extended with μ=`PARA(11)`.
  - 1 Struck quark point-like, remnant extended with μ=`PARA(11)`$/(1 - x)$.
  - 2 as 1, but also struck quark extended with $\mu = Q$.
  - 3 as 2, but emitted quark in initial-state $g \to q\bar{q}$ not extended.

- `MSTA(31)` (D=1) Treatment of masses of extended partons
  - 0 Make extended partons mass-less (for compatibility with previous versions).
  - 1 Extended partons allowed to be massive.
- `MSTA(32)` (D=2) Options for running with LEPTO.
  - 0 Use DIS matrix elements for BGF as given by LEPTO.
  - 1 Add BGF as extra process in first emission using exact matrix element.
  - 2 Include initial state $g \to q\bar{q}$ splitting as an extra process and correct the first emission for $\mathcal{O}(\alpha_S)$ BGF matrix element if `MSTA(33)`>0.
  - -32 If linked together with LDCMC, generate initial-state chain of dipoles according to the LDC model and continue with final-state dipole radiation from these.
- `MSTA(33)` (D=1) Treatment of DIS matrix elements for QCD-Compton
  - 0 Approximated by ordinary dipole matrix element (also use leading log approximation for BGF if `MSTA(32)`=1).
  - 1 Use exact matrix element.
- `MSTA(34)` (D=2) Inclusion of pomeron scattering
  - 0 Not included.
  - 1 Included using built-in structure functions with $zf_i^{I\!P}(z) = c_i z^{a_i}(1-z)^{b_i}$ and $f_{I\!P}^p(x_{I\!P}, t) = \frac{1}{\sigma_{\text{tot}} x_{I\!P}^{p/2}} \sum_{j=1}^3 d_j \exp(e_j t)$, with parameters defined in `/ARPOPA/`.
  - 2 Included using built-in structure functions with $zf_i^{I\!P}(z) = c_i z(1-z)$ and $f_{I\!P}^p(x_{I\!P}, t) = \frac{(1-x_{I\!P})^3}{\sigma_{\text{tot}} x_{I\!P}} \sum_{j=1}^3 d_j \exp(e_j t)$
  - 3 Included using built-in structure functions with $zf_i^{I\!P}(z) = c_i z(1-z)$ and $f_{I\!P}^p(x_{I\!P}, t) = \frac{(1-x_{I\!P})^5}{\sigma_{\text{tot}} x_{I\!P}} \sum_{j=1}^3 d_j \exp(e_j t)$
  - -1 Included using external structure functions as supplied in subroutine `ARUPOM`.
- `MSTA(35)` (D=0) Colour reconnections of dipoles in the cascade
  - 0 Not included.
  - -1 Included after the cascade is done.
  - 1 Included within each initial string seperately.
  - 2 Global reconnections allowed but only below gluon energies given by `PARA(28)`.
  - 3 Global reconnections allowed from start.
- `MSTA(36)` (D=2) Transverse extension μ of the remnant (overrides other switches)
  - 0 μ = `PARA(11/12)` or as defined by other switches.
  - 1 μ = `PARA(11/12)`/(1-z).
  - 2 μ = intrinsic $p_\perp$ times `PARA(14)`.
  - 3 μ = intrinsic $p_\perp$ times `PARA(14)`/(1-x).
  - 4 μ = intrinsic $p_\perp$ times `PARA(14)`/((1-x)(1-z)) in case of sea-quark interaction.
- `MSTA(37)` (D=1) Handeling of primordial $p_\perp$ in remnant
  - 0 Given by Pythia or Lepto
  - 1 Gaussian with $\langle p_\perp^2 \rangle$ = `PARA(27)`^{2}.
  - 2 Exponential with $\langle p_\perp^2 \rangle$ = `PARA(27)`^{2}.
- `MSTA(38)` (D=0) Allow Fragmentation of partons into heavy quarkonia.
  - 0 Not allowed
  - 1 Allowed but in case of $g \to \mathcal{O} + ng$ only emit one gluon.
  - 2 As for 1 but emit all gluons.
- `MSTA(39-40)` Not used.

4 Sample Programs

The easiest way to learn how to use Ariadne is of course by looking at examples. In the following three sample programs are given illustrating how to use Ariadne together with the JETSET, LEPTO and PYTHIA programs. For simplicity they are all assuming that the user have supplied routines for setting parameters and switches and for analyzing the produced events.

The general strategy is to first set all parameters and switches in Ariadne and the program it is running with and then initialize Ariadne with `ARINIT` before initializing the other program. In this way Ariadne can set up the other program so that after it has generated an event the dipole shower can be applied with a simple call to `AREXEC`. This of course is relying on that the user doesn't change any parameters and switches in the other program which influence the way its events are produced, after the call to ARINIT.

Note that these sample programs are included in the distribution as described in appendix A.3.

4.1 Generating LEP events with JETSET

```
      PROGRAM LEP

C...Call a user supplied routine setting
C...the parameters and switches in JETSET
      CALL SETJET

C...Call a user supplied routine setting
C...the parameters and switches in Ariadne
      CALL ARISET

C...Initialize Ariadne to run with JETSET
      CALL ARINIT('JETSET')

C...Loop over a number of events
      DO 100 IEVE=1,10000

C...Generate an LEP event with JETSET
        CALL LUEEVT(0,91.0)

C...Apply the Dipole Cascade
        CALL AREXEC

C...Call a user supplied analysis routine
        CALL LEPANA

 100  CONTINUE

      END
```

In the call to ARINIT the parton evolution is completely switched off in JETSET and so is the fragmentation. If fragmentation previously was switched on in JETSET it will instead be switched on in Ariadne so that AREXEC will end with a call to LUEXEC.

Note that by commenting out the calls to ARINIT and AREXEC, this program will produce events with JETSET as set up in SETJET.

4.2 Generating HERA events with LEPTO

```
      PROGRAM HERA

C...Call a user supplied routine setting
C...the parameters and switches in LEPTO
      CALL LEPSET

C...Call a user supplied routine setting
C...the parameters and switches in Ariadne
      CALL ARISET

C...Initialize Ariadne to run with LEPTO
      CALL ARINIT('LEPTO')

C...Initialize LEPTO for HERA
      CALL LINIT(0,11,-26.5,820.0,4)

C...Loop over a number of events
      DO 100 IEVE=1,10000

C...Call generate an event with LEPTO
        CALL LEPTO

C...Apply the Dipole Cascade
        CALL AREXEC

C...Call a user supplied analysis routine
        CALL HERANA

 100  CONTINUE

      END
```

The comments made for the JETSET case also applies here.

4.3 Generating LHC events with PYTHIA

```
      PROGRAM LHC

C...Call a user supplied routine setting
C...the parameters and switches in PYTHIA
      CALL PYTSET

C...Call a user supplied routine setting
C...the parameters and switches in Ariadne
      CALL ARISET

C...Initialize Ariadne to run with PYTHIA
      CALL ARINIT('PYTHIA')

C...Initialize PYTHIA for LHC
      CALL PYINIT('CMS','p+','p+',16000.0)

C...Loop over a number of events
      DO 100 IEVE=1,10000

C...Call generate an event with PYTHIA
        CALL PYEVNT

C...Apply the Dipole Cascade
```

```
      CALL AREXEC

C...Call a user supplied analysis routine
      CALL LHCANA

 100  CONTINUE

      END
```

Again the comments in the JETSET case also applies here. Note however that Ariadne presently only can handle a small subset of the sub-processes available in PYTHIA, and attempts to use Ariadne for other sub-processes will result in a warning from Ariadne.

References

[1] L. Lönnblad, *Comput. Phys. Commun.* **71** (1992) 15.

[2] G. Gustafson, *Phys. Lett. B* **175** (1986) 453.

[3] G. Gustafson, U. Pettersson, *Nucl. Phys. B* **306** (1988) 746.

[4] B. Andersson et al., *Z. Phys. C* **43** (1989) 625.

[5] B Andersson, G. Gustafson, L. Lönnblad, *Nucl. Phys. B* **339** (1990) 393.

[6] OPAL Coll., M.Z. Akrawy et al., *Z. Phys. C* **47** (1990) 505.

[7] N. Magnussen et al., Generators for Deep Inelastic Scattering, in *Physics at HERA*, edited by W. Buchmüller, G. Ingelman, volume 3, p. 1167, DESY, 1991.

[8] T. Sjöstrand, *Comput. Phys. Commun.* **82** (1994) 74.

[9] T. Sjöstrand, *Comput. Phys. Commun.* **82** (1994) 74.

[10] T. Sjöstrand, *Comput. Phys. Commun.* **39** (1986) 347.

[11] G. Ingelman, LEPTO version 6.1, in *Physics at HERA*, edited by W. Buchmüller, G. Ingelman, volume 3, p. 1366, DESY, 1991.

[12] H-U. Bengtsson, T. Sjöstrand, *Comput. Phys. Commun.* **46** (1987) 43.

[13] U. Pettersson, ARIADNE – A Monte Carlo for QCD Cascades in the Colour Dipole Formulation, Lund Preprint LU–TP 88–5, 1988.

[14] ARIADNE 2, Lund Preprint LU–TP 88–15, 1988.

[15] ARIADNE 3, Lund Preprint LU TP 89–10, 1989.

[16] R. Kleiss, *Phys. Lett. B* **180** (1986) 400.

[17] M.H. Seymour, *Nucl. Phys. B* **436** (1995) 163.

[18] The dipole model structure functions, DESY 92–098 and Lund preprint LU–TP 92–21, 1992.

[19] L. Lönnblad, *Z. Phys. C* **65** (1995) 285.

[20] B. Andersson, G. Gustafson, B. Söderberg, *Z. Phys. C* **20** (1983) 317.

[21] G. Ingelman, P. Schlein, *Phys. Lett. B* **152** (1985) 256.

[22] P. Bruni, G. Ingelman, POMPYT version 1.0 program and manual, DESY preprint in preparation.

[23] H. Kharraziha, L. Lönnblad, The Linked Dipole Chain Monte Carlo, preprint LU-TP 97-21, NORDITA-97/54 P, hep-ph/9709424, December 1997.

[24] B. Andersson, G. Gustafson, J. Samuelsson, *Nucl. Phys. B* **463** (1996) 217.

[25] B. Andersson, et al., *Z. Phys. C* **71** (1996) 613.

[26] L. Lönnblad, Photon Radiation in the Dipole Model and in the Ariadne Program, in *Workshop on photon radiation from quarks*, edited by S. Cartwright, p. 109, CERN, 1992.

[27] UA1 Coll., G. Arnison et al., *Nuovo Cim. Lett.* **44** (1985) 1.

[28] UA2 Coll., J.A. Appel et al., *Z. Phys. C* **30** (1986).

[29] L. Lönnblad, *Nucl. Phys. B* **458** (1996) 215.

[30] P. Ernström, L. Lönnblad, *Z. Phys. C* **75** (1997) 51.

[31] P. Ernström, L. Lönnblad, M. Vänttinen, *Z. Phys. C* **76** (1997) 515.

[32] G. Cowan, private communication.

[33] W. de Boer, H. Fuerstenau, DELPHI data in comparison with QCD models and grand unified theories, in *MC91 Workshop on detector and event simulation in high energy physics*, edited by K. Bos, B. van Eijk, p. 616, NIKHEF–H, 1991.

[34] L. Lönnblad, *Z. Phys. C* **58** (93) 471.

[35] JADE Coll., W. Bartel et al., *Z. Phys. C* **33** (1986) 23.

[36] B. Andersson, G. Gustafson, *Z. Phys. C* **3** (1980) 223.

[37] B. Andersson, G. Gustafson, G. Ingelman, T. Sjöstrand, *Phys. Rep.* **97** (1983) 31.

[38] Yu.L. Dokshitzer, V.A. Khoze, S.I. Troyan, New perturbative results in hadron jet physics, in *6th Int. Conf. on Physics in Collision*, edited by M. Derrick, World Scientific, Singapore, 1987.

A Appendix: Technical Information

The Ariadne program is written according to the FORTRAN 77 standard and should work on any platform with a FORTRAN 77 compiler.

To avoid name clashes when run together with other programs, all external names in Ariadne begins with the two character `AR`. All internal identifiers conforms to the standard FORTRAN 77 implicit declarations except for double precision and logical variables which are declared in all subroutines as

```
IMPLICIT DOUBLE PRECISION (D)
IMPLICIT LOGICAL (Q)
```

Ariadne performs a large amount of boost to and from the center of mass frames of the radiating dipoles which may give rise to precision problem when the program is used for simulations at very high energies. To avoid these Ariadne has an additional declaration of double precision variables

```
IMPLICIT DOUBLE PRECISION (B)
```

On some machines double precision arithmetic is much slower than single precision, and if you feel that speed is more important than precision this latter double precision declaration may be commented out (in every routine). Note that possible warnings about non-conservation of four-momentum may be avoided by adjusting `MSTA(9)` and `PARA(39)` (see section 3.4).

The internal event record of Ariadne has a maximum number of partons, dipoles and strings which it can handle. These number are defined in the parameter statement in each routine and are by default set to:

```
PARAMETER(MAXDIP=500,MAXPAR=500,MAXSTR=100)
```

These limits can of course be changed by the user, but it should be noted that generation of more than 500 partons in Ariadne most probably is an indication that an error has occurred.

Ariadne contains a block data routine `ARDATA` for setting the default values of the parameters and switches used. When compiling Ariadne in separate modules, this block data routine should be compiled in the same module as `ARINIT`. Otherwise, since `ARDATA` is never actually referenced, it will not be linked and Ariadne will not be properly initialized.

Ariadne has to be loaded together with version 7.4 or later of the JETSET program. In addition, when run in 'LEPTO' mode, it should be loaded together with version 6.3 or later of the LEPTO program and, when run in 'PYTHIA' mode, together with version 5.7 or later of the PYTHIA program.

A.1 Availability

The program is available from the web page `http://www.thep.lu.se/tf2/staff/leif/ariadne/`

A.2 Installation

If the program has been obtained through E-mail correspondence it should simply be extracted into a file and be compiled.

To install the program obtained the web, the compressed tar-file should be "un-compressed" and "un-tarred" which will create a directory called `ariadne-4.10`. This directory will contain, besides the actual code, a file called `README` containing all instructions needed to install the program. In addition there will be the files `ariadne.man` and `ariadne.ps` containing the latest revision of this manual in ASCII and Postscript format respectively.

Note that the distribution contains two dummy subroutines `PYSTFU` and `LNSTRF`. These are needed to avoid linking problems when *not* running with PYTHIA or LEPTO, and should be removed when running with PYTHIA or LEPTO respectively.

A.3 Test Programs

Ariadne contains a subroutine called `ARTEST` intended to be used for confirming that the installation has been successful. To use it, write a small program calling the routine (this program is included in the `tar`-distribution as the file `atest.f`):

```
PROGRAM TEST

CALL ARTEST(0)

END
```

When run, `ARTEST` will generate 10000 events randomly distributed in center of mass energy and check their consistency w.r.t. momentum conservation and colour flow. If Ariadne was successfully installed, a message

```
No errors experienced by Ariadne.
```

will be printed. If anything else is printed, such as

```
2 errors occurred in Ariadne.
```

please consult the author.

In the `tar`-distribution the sample programs described in section 4 are also included as the files `jtest.f`, `ltest.f` and `ptest.f` including dummy routines for parameter settings and analysis.

B Appendix: Description of Subroutines and Common Blocks

This is a (no longer) complete list of the subroutines in Ariadne.

- `SUBROUTINE ARRADG(ID)`
  Administers the emission of a gluon from dipole `ID`.
- `REAL FUNCTION ARANGL(I1,I2)`
  Returns the angle between partons `I1` and `I2` in radians.
- `SUBROUTINE ARBOCM(ID)`
  Boosts the partons in dipole `ID` to their center of mass frame.
- `SUBROUTINE ARBOLE(THEL,PHI1,PHI2, DBXL,DBYL,DBZL)`
  Boost partons to the hadronic c.m.s. of a LEPTO event. The reverse transform is given by first rotation `PHI2` around the z-axis, then `THEL` around the y-axis, `PHI1` around the x-axis and last boosting with `DBXL`, `DBYL` and `DBZL`.
- `SUBROUTINE ARBOPY(THEPY,PHIPY, DBXPY,DBYPY,DBZPY)`
  Boost partons to the total c.m.s. of a PYTHIA event. The reversed transformation is achieved by calling `LUDBRB(1,N,THEPY,PHIPY,DBXPY,DBYPY,DBZPY)`.
- `SUBROUTINE ARCASC`
  Contains the main loop over dipole emissions.
- `SUBROUTINE ARCHEM(IMOD)`
  Checks that energy and momentum is conserved in Ariadne.
- `SUBROUTINE ARCHKI(ID)`
  Checks that the emission generated for dipole `ID` is kinematically allowed.

- SUBROUTINE ARCLUS(NJET)
  Jet-clustering routine implementing the "inverse dipole radiation" algorithm.
- SUBROUTINE ARCONT
  Continues a dipole cascade peviously started with ARCASC.
- SUBROUTINE ARCOPA(IJ,IP,ITYP)
  Copies a parton from position IJ in /LUJETS/ to position IP in /ARPART/.
- SUBROUTINE ARCOPJ
  Copies particles to be considered jet-initiators to the end of /LUJETS/.
- SUBROUTINE ARCRDI(ID,IPA1,IPA3,IS,QED)
  Creates a dipole entry in ID /ARDIPS/ from the partons at position IPA1 and IPA3 in /ARPART/.
- SUBROUTINE ARDBRB(DTHE,DPHI, DBEX,DBEY,DBEZ,NI,I)
  Same as ARROBO below but with double precision arguments also for the rotation angles.
- SUBROUTINE ARDUMP
  Copies a partonic state from the internal event record to /LUJETS/.
- SUBROUTINE ARDUPH
  Copies a photon radiated by Ariadne to /LUJETS/.
- REAL FUNCTION ARDYRE(IDE,BW,QRG1,QRG3)
  Transfers the recoil of an emission to a Drell-Yan produced particle if the emission and the particle are in the same phase space region. Returns a negative number if no transfer was possible. IDE is the position of the emitting dipole, BW the total invariant mass, and QRG1 and QRG3 are .TRUE. if the corresponding dipole ends are recoil gluons.
- REAL FUNCTION ARECOI(BE0,DE1,DE2,DE3, BP0,DP1,DP2,DP3,BALP,PT12)
  Calculates the recoil angle in the strategy of MSTA(17) = 3. BE0, BE1, BE2, BE3, BP0, BP1, BP2 and BP3 are the energy and momentum of the incolved partons, BALP is the angle between partons 1 and 3 and PT12 is the invarioant $p_\perp$ of the emission.
- SUBROUTINE AREMIT
  Administers the actual emission from dipole ID.
- SUBROUTINE ARERRM
  Prints out an error message and optionally stops the execution. If the execution is allowed to continue the value of MSTA(13) will be set to a value corresponding to the warning produced:

  3 /LUJETS/ event record was not properly formatted.

  9 Total four-momentum was not conserved in Ariadne.

  10 A particle was found to have inconsistent four-momentum.

  13 A dipole was found to have inconsistent invariant mass.

  20 Selected sub-process in PYTHIA is not supported by Ariadne.

  21 ARCLUS was not performed due to too many jet-initiators.
- SUBROUTINE AREVOL(PTMAX,PTMIN)
  Perform a dipole shower evolution allowing emissions between PTMAX and PTMIN.
- SUBROUTINE AREVST(ISIN)
  Reverse the colour flow of string entry ISIN.
- SUBROUTINE AREXEC
  This is the main routine in Ariadne. Given a partonic state in /LUJETS/ it administers the dipole radiation according to the options and parameters set in /ARDAT1/.
- SUBROUTINE AREXMA(I1,I3)
  Makes partons I1 and I3 mass-less if extended.
- SUBROUTINE ARFIDY(NSAVE) Fix up event record after dipole shower in Drell-Yan events.
- SUBROUTINE ARGDIG(ID)
  Calculates the $p_\perp^2$ of a possible gluon emission from the colour dipole ID assuming it is the first emission of a DIS event.
- SUBROUTINE ARGDIS(ID)
  Prepare for the calculation of the $p_\perp^2$ of a possible gluon emission in ARGDIG.
- REAL FUNCTION ARGPT2(ID)
  Returns the generated $p_\perp^2$ for a possible emission from dipole ID. If necessary it calls the relevant procedure to generate this $p_\perp^2$.
- SUBROUTINE ARGQCD(ID)
  Calculates the $p_\perp^2$ of a possible emission from the colour dipole ID.
- SUBROUTINE ARGQCG(ID)
  The gluon emission part of ARGQCD.
- SUBROUTINE ARGQCQ(ID)
  The gluon splitting part of ARGQCD.

- SUBROUTINE ARGQED(ID)
  Calculates the $p_\perp^2$ of a possible emission from the electro-magnetic dipole ID.

- SUBROUTINE ARGTYP(I,ITYP)
  Determines the colour state of a particle in /LUJETS/.

- SUBROUTINE ARINIT(MODE)
  Initializes the Ariadne program. The argument is a character string indicating which program Ariadne is used with; 'JETSET', 'LEPTO', 'PYTHIA' or by itself - 'ARIADNE'.

- SUBROUTINE ARINQQ(IT,KQ,IRP, PT2,ZQ,PHI,QFAIL)
  Try to perform an initial-state $g \to q\bar{q}$ splitting given which target side IT, the flavour of the struck quark KQ, the position of the chopped-off remnant particle, the $p_\perp^2$ (PT2), momentum fraction ZQ and azimuthal angle PHI of the struck quark. If the emission fails QFAIL is set to .TRUE.

- REAL FUNCTION ARIPSF(A,N,X)
  Return the integral from X to 1 of the function $x^{\mathtt{A}}(1-x)^{\mathtt{N}}$.

- REAL FUNCTION ARIPT2(I1,I2,I3)
  Returns the invariant $p_\perp^2$ of parton I2 w.r.t. the partons I1 and I3.

- SUBROUTINE ARJOIN(J1,J2,J3)
  Clusters the three jet-entries J1, J2 and J3 in /LUJETS/ into two according to a "reversed" dipole emission scenario.

- SUBROUTINE ARJOQQ(IQ1,IQ2)
  Join the quark and anti-quark entries IQ1 and IQ2 into a gluon entry located at MIN(IQ1,IQ2).

- SUBROUTINE ARJOST(IS1,IS2,IPA1,IPA2)
  Join the string entries IS1 and IS2 in the ends IP1 and IP2.

- SUBROUTINE ARLEPT
  The interface to LEPTO used by AREXEC for MSTA(32) = 1.

- SUBROUTINE ARMADE
  Determined some mass-dependent factors for use in the veto-algorithm.

- REAL FUNCTION ARMASS(N,I)
  Returns the square of the invariant mass of the N partons pointed to in the vector I(N).

- SUBROUTINE ARMCDI(ARRNDX,ARRNDY,ARVETO)
  Implements the veto-algorithm for generating a $p_\perp^2$ for any dipole given the functions ARRNDX, ARRNDY and ARVETO.

- REAL FUNCTION ARMIPT(IF,IL)
  Returns the minimum invariant $p_\perp^2$ of the partons between positions IF and IL in /ARPART/.

- REAL FUNCTION ARNOFL(W,MNOFW)
  Returns the number of flavors to be used to calculate α_s at a scale W.

- SUBROUTINE ARORDJ
  Orders the jet entries in /LUJETS/ according to their energy.

- SUBROUTINE ARORIE(I1,I2,I3, BS,B1,B3,QR1,QR3,PT21,PT23)
  Orients the partons I1, I2 and I3 in their center of mass frame, given their energy fractions and their total invariant mass.

- SUBROUTINE ARPARS(NSTART,NEND)
  Parses the /LUJETS/ common block between positions NSTART and NEND, copying partons to be cascaded into the internal event record.

- REAL FUNCTION ARPCMS(S,SM1,SM2)
  Returns the positive light-cone component of a particle momentum when placed in the c.m.s. system of itself and an other particle, given the two particle masses SM1 and SM2 and the total energy squared S.

- SUBROUTINE ARPHAS(IFIRST) Rotate event begining from entry IFIRST to correctly take into account azimuthal asymmetries in gluon emission in DIS.

- SUBROUTINE ARPOKI(IT,IQR,IDR,IRP,IDIR, KFTF,KFPR,XPOM,TPOM,QFAIL)
  Redistribute remnants assuming an incoming hadron IT, outgoing hadron KFTF with a pomeron (or other colourless sub-object) with momentum fraction XPOM and viruality -TPOM giving a remnant of flavour KFPR. IQR, IDR and IRP as in subroutine ARREMN.

- SUBROUTINE ARPOSF(KFT,KFSTR,X,XQ2, XPOM,TPOM,KFTF,KFPR,XFP,XFPOM)
  Returns some information about possible pomeron interaction. The arguments are the same as for ARUPOM in section 3.1.

- SUBROUTINE ARPRDA
  Prints out the values of the parameters and switches used by Ariadne.

- `SUBROUTINE ARPRDY`
  Make special preparations of the output from PYTHIA in case of a Drell-Yan event.

- `ARPTQQ(KF,KQ,PT2MAX,PT2MIN, X,XQ2,XY,XP,ZQ,PHI)`
  Generate kinematical variables describing an initial state $g \to q\bar{q}$ splitting. `KF` is the target code, `KQ` the flavour of the struck quark, `PT2MAX` and `PT2MIN` limits the possible $p_\perp^2$ of the emission, `X`, `XQ2` and `XY` are the usual DIS parameters x, Q^2 and y. `XP` is the ratio between x and the momentum fraction of the incoming gluon, `ZQ` is the momentum fraction and `PHI` the azimuthal angle of the struck quark.

- `SUBROUTINE ARPYTH`
  The interface to PYTHIA used by `ARECEX`.

- `SUBROUTINE ARRADG(ID,NREM,SNR)`
  Performs the emission of a gluon from dipole `ID`.

- `SUBROUTINE ARRADP(ID)`
  Performs the emission of a photon from dipole `ID`.

- `SUBROUTINE ARRADQ(ID)`
  Performs the splitting of a gluon into a $q\bar{q}$ pair in dipole `ID`.

- `SUBROUTINE ARRECA(ID,IDS,IS1,IS3)`
  Recalls a full dipole entry to the internal event record, previously stored away by `ARSTOR`.

- `SUBROUTINE ARREMD(IREM)`
  Remove dipole entry `IREM`.

- `SUBROUTINE ARREMG(IGI)`
  Remove gluon entry `IGI`.

- `SUBROUTINE ARREMN(IT,IQR,IDR,IRP,IDIR)`
  Redistribute remnants on side `IT` given a single quark remnant in position `IQR` and/or a di-quark remnant in position `IDR` and/or a chopped off particle remnant in position `IPR`. `IDIR` indicates the direction along the z-axis of the incomming parton.

- `SUBROUTINE ARREMP(IREM)`
  Remove parton entry `IREM`.

- `REAL FUNCTION ARNDX1(), ARNDX2()` etc.
  Different functions for generating a $p_\perp^2$ according to a Sudakov-suppressed suppression, to be used by `ARMCDI`.

- `REAL FUNCTION ARNDY1(), ARNDY2()` etc.
  Different functions for generating a rapidity according to a flat distribution, to be used by `ARMCDI`.

- `SUBROUTINE ARROBO(THE,PHI, DBEX,DBEY,DBEZ,N,I)`
  Rotates and boost the `N` partons pointed to by the vector `I(N)`. The polar rotation is performed first (`THE`) followed by the azimuth rotation (`PHI`) and the boost.

- `SUBROUTINE ARSCAN(NSTART,NEND,NR,IR)`
  Same as `ARPARS` above but don't perfom any cascade. Instead put a special marker any of the entries copied which are listed in the array `IR(NR)`.

- `SUBROUTINE ARSPLG(IG,IFLAV)`
  Splits the gluon entry `IG` into a quark and an anti-quark entry with flavors determined by `IFLAV`.

- `SUBROUTINE ARSTOR(ID,IDS,IS1,IS3)`
  Stores away a full dipole entry in the internal event record for later use.

- `REAL FUNCTION ARSTRA(KF,KQ,X,XP,XQ2)` Return the ratio of the structure function of flavour `KQ` in target `KF` evaluated at `X/XP` and `XQ2` and at `X` and `XQ2`.

- `SUBROUTINE ARSUME(NULL,BSX,BSY,BSZ,BSE,BSM,NI,I)`
  Sum up energy and momentum of all entries listed in the array `I(NI)`. `BSE`, `BSX`, `BSY` and `BSZ` are the sum of the energy and momentum and are set to zero befor summation if `NULL = 0`. `BSM` is the total invariant mass.

- `SUBROUTINE ARTEST(IPRINT)`
  A test program to check that Ariadne has been installed properly, disguised as a subroutine.

- `REAL FUNCTION ARTPT2(ID,SN,BX1IN,BX3IN, Y1IN,Y2IN,Y3IN)`
  Calculate the minimum $p_\perp$ of a radiated particle w.r.t. any of the original dipole ends for a given dipole `ID`. If `ID` is zero, use total invariant mass squared in `SN`, x_1 and x_3 in `BX1IN` and `BX3IN` together with the scaled mass ssquared of the partons `Y1IN`, `Y2IN` and `Y3IN`.

- `SUBROUTINE ARTUNE(SET)`
  Sets the parameters in Ariadne as described in section 3.1.

- `SUBROUTINE ARUPDJ(I2,I1,I3)`
  Calculates the minimum invariant $p_\perp^2$ of a jet entry in `/LUJETS/` with respect to any other pair of jet-entries.

- SUBROUTINE ARUPOM(KFT,KFSTR,X,XQ2, XPOM,TPOM,KFTF,KFPR,XFP,XFPOM)
  This is a dummy routine which can be used to implement a user defined pomeron distribution and structure function. See section 3.1 for details.
- REAL FUNCTION ARVET1(), ARVET2() etc.
  Different routines for calculating the veto factor to be used by ARMCDI.
- REAL FUNCTION ARZCMS(S,SM1,SM2)
  Returns the z component of a particle momentum when placed in the c.m.s. system of itself and an other particle given the two particle masses SM1 and SM2 and the total energy squared S.

The following common blocks are used in Ariadne:

```
      COMMON /ARDAT1/ PARA(40),MSTA(40)
```

The parameters and switches used in Ariadne as explained in section 3.4.

```
      COMMON /ARDAT2/ PQMAS(10)
```

The quark masses used in Ariadne as described in section 3.3.

```
      COMMON /ARDAT3/ IWRN(40)
```

The number of errors and warnings of each kind experienced by Ariadne.

```
      COMMON /ARPART/ BP(MAXPAR,5),IFL(MAXPAR),QEX(MAXPAR),
     $                QQ(MAXPAR),IDI(MAXPAR),IDO(MAXPAR),
     $                INO(MAXPAR),INQ(MAXPAR),XPMU(MAXPAR),
     $                XPA(MAXPAR),PT2GG(MAXPAR),IPART
```

The internal representation of partons in Ariadne:

- BP(I,1) x-component of the momentum of parton I.
- BP(I,2) y-component of the momentum of parton I.
- BP(I,3) z-component of the momentum of parton I.
- BP(I,4) energy of parton I.
- BP(I,5) mass of parton I.
- IFL(I) flavor code of parton I.
- QEX(i) is .TRUE. if parton I is extended
- QQ(I) is .TRUE. if parton I is in a colour-3 or $\bar{3}$ state.
- IDI(I) position of "incoming" dipole in /ARDIPS/.
- IDO(I) position of "outgoing" dipole in /ARDIPS/.
- INO(I) The number of the emission in which parton I was produced.
- INQ(I) For a quark produced in gluon splitting, the position of the corresponding anti quark, otherwisw zero.
- XPMU(I) Soft suppression parameter μ for parton I.
- XPA(I) Power in soft suppression - α - for parton I.
- IPART The number of partons presently in /ARPART/.

```
      COMMON /ARDIPS/ BX1(MAXDIP),BX3(MAXDIP),
     $       PT2IN(MAXDIP),SDIP(MAXDIP),IP1(MAXDIP),
     $       IP3(MAXDIP),AEX1(MAXDIP),AEX3(MAXDIP),
     $       QDONE(MAXDIP),QEM(MAXDIP),IRAD(MAXDIP),
     $       ISTR(MAXDIP),ICOLI(MAXDIP),IDIPS
```

The internal representation of dipoles in Ariadne.

- BX1(ID) value of x_1 generated for dipole ID.
- BX3(ID) value of x_3 generated for dipole ID.
- PT2IN(ID) invariant $p_\perp^2$ generated for dipole ID.
- SDIP(ID) invariant mass squared of dipole ID.
- IP1(ID) position of parton 1 in /ARPART/.
- IP3(ID) position of parton 3 in /ARPART/.
- AEX1(ID) value of $a = (\mu/p_\perp)^\alpha$ for parton 1.
- AEX3(ID) value of $a = (\mu/p_\perp)^\alpha$ for parton 3.
- QDONE(ID) is .TRUE. if a $p_\perp$has been generated for dipole ID.
- QEM(ID) is .TRUE. if ID corresponds to an EM-dipole.
- IRAD(ID) the type of emission generated for dipole ID. 0: gluon radiation (or photon radiation for EM dipole). $(-)n$: $q\bar{q}$ radiation of flavor n splitting gluon 1 (3).
- ISTR(ID) The string entry in common block /ARSTRS/ to which dipole ID belongs.
- ICOLI(ID) The colour index of dipole ID.
- IDIPS The number of dipoles currently in /ARDIPS/.

```
      COMMON /ARSTRS/ IPF(MAXSTR),IPL(MAXSTR),
     $       IFLOW(MAXSTR),PT2LST,IMF,IML,IO,QDUMP,ISTRS
```

The internal representation of strings in Ariadne

- IPF(IS) position of the first parton in /ARPART/.
- IPL(IS) position of the last parton in /ARPART/.
- IFLOW(IS) the direction of colour flow in string IS. A positive value corresponds to IPF(ID) being a colour-3 parton.
- PT2LST $p_\perp^2$ of the last emission in Ariadne
- IMF The position of the first parton in the parent string in /LUJETS/.
- IML The position of the first parton in the parent string in /LUJETS/.
- IO The number of emissions performed for the parent string.
- QDUMP is .TRUE. if current event information has been copied into the /LUJETS/ common block.
- ISTRS The number of strings currently in /ARSTRS/.

```
      COMMON /ARINT1/ BC1,BC3,BZM,BZP,BP1,BM1,BP3,BM3,
     $       B1,B2,B3,XT2,XT,Y,QQ1,QQ3,NE1,NE3,
     $       S,W,C,CN,ALPHA0,XLAM2,IFLG,
     $       XT2MP,XT2ME,XT2M,XT2C,XTS,XT3,XT1,
     $       YINT,YMAX,YMIN,
     $       Y1,Y2,Y3,SY1,SY2,SY3,SSY,
     $       AE1,AE3,NXP1,NXP3,FQ1,FQ3
```

The common variables needed for the veto-algorithm in subroutine ARMCDI.

```
      COMMON /ARINT2/ DBEX,DBEY,DBEZ,PHI,THE
```

Information of the boost vector and rotation angles for transformation of the radiating dipole back to the original Lorenz-frame.

```
      COMMON /ARINT3/ DPTOT(5)
```

The total energy and momentum of the parton state being considered by Ariadne.

```
      COMMON /ARINT4/ BASS(5),BASSX1,BASSX3,IFLASS
```

Information to fix azimuthal asymmetry for the first gluon emission in DIS. BASS(5) is the four-momentum an mass of the emitted gluon, BASSX1 and BASSX3 are the x_1 and x_3 of the corresponding emission and IFLASS is the flavour of the gluon or zero if a foson-gluon fusion event.

```
      COMMON /ARSTRF/ KFSAVE(2),XSAVE(2),XQ2SAV(2),XPQSAV(2,-6:6)
```

Structure function information in the current PYTHIA event. KFSAVE(IT) is the code for the target on side IT. XSAVE(IT) and XQ2SAV(IT) is the x and Q^2 of the incoming partons to the hard interaction on the corresponding sides and XPQSAV(IT,IFL) is x times the structure function of flavour IFL.