Computer Physics Communications 71 (1992) 15–31
North-Holland

Computer Physics
Communications

Ariadne version 4 – A program for simulation of QCD cascades implementing the colour dipole model

Leif Lönnblad [1]

Deutsches Elektronen Synchrotron - DESY, Notkestraße 85, W-2000 Hamburg 52, Germany

Received 27 March 1992

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

The major changes from previous versions are the introduction of photon radiation from quarks and inclusion of interfaces to the LEPTO and PYTHIA programs.

1. Introduction

The colour dipole model (CDM) [1–3] as implemented in the Ariadne program has had considerable success in describing data from both e^+e^- [4] and lepto-production [5] 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 handle hard interactions, hadronization and particle decays. Standard interfaces to the JETSET [6], LEPTO [7] and PYTHIA [8] programs are included in the version presented in this paper.

1.1. Update history

The first version of Ariadne [9] only handled gluon radiation from primary quarks in e^+e^- collisions. Since then the program has evolved as follows:

- version 2 [10]: Included gluon emission from extended emitters to describe the QCD showers in lepto-production.
- version 3 [11]: 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 (this version): Completely rewritten, built around a new internal dipole-oriented event record. The preliminary features of subversions 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.

Correspondence to: L. Lönnblad, Deutsches Elektronen Synchrotron - DESY, Notkestraße 85, W-2000 Hamburg 52, Germany.

[1] E-mail: lonnblad@apollo3.desy.de (internet) and LONNBLAD@DESYVAX (bitnet).

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 is handled through the event record in the JETSET `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, explains 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

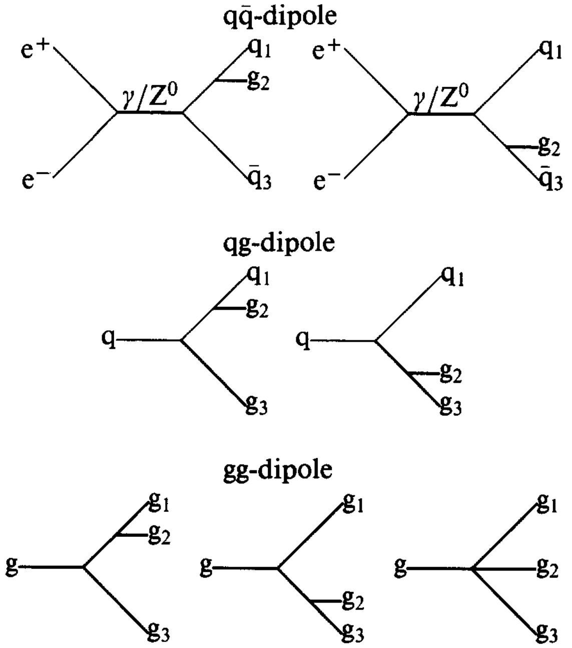

Fig. 1. The relevant Feynman diagrams for gluon emission from a q$\bar{q}$, qg and a gg dipole.

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 fig. 1 and can be written 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 $2E_i/\sqrt{S_{\text{dip}}}$ of the emitting partons in the dipoles center of mass system. Note that eqs. (1) and (2) are slightly modified when finite quark masses are taken into account (see description of `MSTA (19)` in section 3.5). It should also be noted that these cross-sections correctly reproduce 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

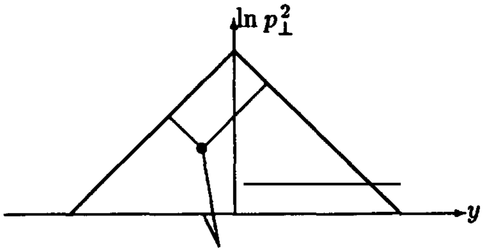

Fig. 2. The phase space limits for emission of a first gluon (thick lines) and a second gluon (thin lines).

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

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

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

$$\frac{\mathrm{d}P(p_\perp^2, y)}{\mathrm{d}p_\perp^2\,\mathrm{d}y} = \frac{\mathrm{d}\sigma(p_\perp^2, y)}{\mathrm{d}p_\perp^2\,\mathrm{d}y} \exp\left(-\int_{p_\perp^2}^{p_{\perp\max}^2} dk_\perp^2 \mathcal{I}(k_\perp^2)\right), \quad (5)$$

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

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

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

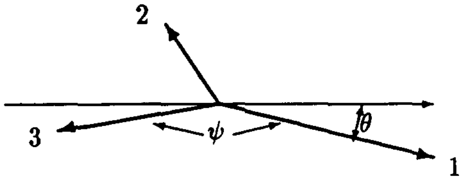

Fig. 3. The orientation of a dipole after emission.

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

$$y = \tfrac{1}{2}\ln\frac{1 - x_1}{1 - x_3} \quad (7)$$

all three cross-sections are well approximated by

$$\mathrm{d}\sigma \propto \frac{\mathrm{d}p_\perp^2}{p_\perp^2}\,\mathrm{d}y \quad (8)$$

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 fig. 2.

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

2.3. Recoils

The cross-sections in eqs. (1)–(3) do 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 fig. 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 exists a prescription [12] in the case of a $\mathrm{q\bar{q}}$ dipole derived

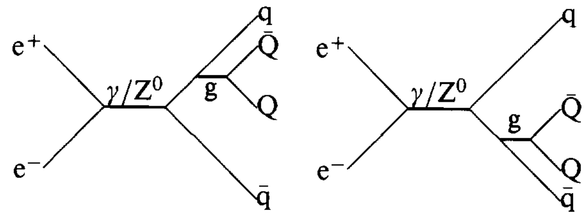

Fig. 4. The Feynman diagrams relevant for $e^+e^- \to q\bar{q}Q\bar{Q}$.

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 prescriptions 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

$$\theta = \frac{x_3^2}{x_1^2 \mp x_3^2}(\pi - \psi), \tag{9}$$

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 cannot be straightforwardly introduced into the dipole picture. Looking at the cross-section for the process $e^+e^- \to q\bar{q}Q\bar{Q}$, corresponding to the diagrams in fig. 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 eq. (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:

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

where 2 and 3 denote 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

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

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

Hence there is a competition between this process and the possibility to emit another gluon instead. It turns out [3] that the choice of $p_\perp^2$ as ordering variable tends to favor the $Q\bar{Q}$ production more than in conventional parton cascades (e.g. [6]) where the ordering is typically in Q^2.

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 pointlike, but for DIS only the struck quark is pointlike 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

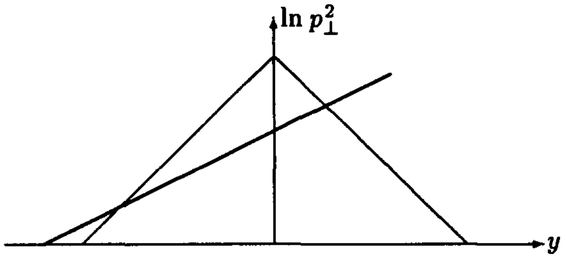

Fig. 5. The available phase space for gluon emission in DIS. Positive rapidity corresponds to the direction of the struck quark.

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 [2] by allowing only a fraction

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

of the hadron remnant 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 fig. 5, leading to a suppression of the radiation in the target region.

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. Analyzing the role of the recoil gluon, it can, however, be shown that it corresponds to a gluon radiated in the initial state in a parton shower scenario [13] and that in the low $p_\perp$ limit, the two approaches are basically equivalent if the recoil gluon is given a transverse recoil according to

$$p_{\perp 3}^2 \, e^{-y_3} = p_{\perp 2}^2 \, e^{-y_2}, \tag{14}$$

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 eq. (14).

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 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 [14]. The cross-section for emitting a photon from a $q\bar{q}$ pair is given by

$$\frac{d\sigma}{dx_1\, dx_3} = \frac{\alpha_{\rm EM}}{2\pi} e_q^2 \frac{x_1^2 + x_3^2}{(1-x_1)(1-x_3)}, \tag{15}$$

which is the same as eq. (1) substituting α_s with $\alpha_{\rm EM}$ 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

$$\frac{dP_\gamma(p_\perp^2, y)}{dp_\perp^2\, dy} = \frac{d\sigma_\gamma(p_\perp^2, y)}{dp_\perp^2\, dy} \times \exp\left(-\int_{p_\perp^2}^{p_{\perp,max}^2} dk_\perp^2 \left(\mathcal{I}_\gamma(k_\perp^2) + \mathcal{I}_g(k_\perp^2)\right)\right), \tag{16}$$

where $\mathcal{I}_i$ is given by eq. (12).

If at first a gluon is emitted from the q$\bar{\text{q}}$ 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 $\bar{\text{q}}$, emitting a gluon from each of the two colour dipoles qg and g$\bar{\text{q}}$ and splitting the gluon into a new Q$\bar{\text{Q}}$ 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$\bar{\text{q}}$ 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$\bar{\text{q}}$ 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 are possible.

2.8. Photon–gluon fusion in DIS

As the CDM only looks at radiation from the dipole between the struck quark and the hadron remnant in DIS, it does not give a satisfactory description of the photon–gluon fusion process, where it is possible that both a quark and an antiquark receive large momentum transfer, creating one dipole each with the remnant.

Picturing the hadron as a Lund-type string, there is, however, no clear distinction between the photon–gluon fusion process and ordinary sea–quark interaction. Both processes can be described in terms of resolving a kink on the string (a gluon) into a (virtual) q$\bar{\text{q}}$ pair. The main difference is the time scales involved. In the case of sea–quark interaction, the q$\bar{\text{q}}$ pair has lived long enough for the partons to be considered independent of each other, which is not the case in photon–gluon fusion.

In the interface to the LEPTO program, the default procedure is to let LEPTO generate the hard process according to first order α_s. In the case of a $\gamma\text{q} \to \text{qg}$ process, the gluon is simply removed as the gluon emission is assumed to be well treated by the dipole radiation. In the case of $\gamma\text{g} \to \text{q}\bar{\text{q}}$, the virtuality of the q$\bar{\text{q}}$ pair is taken to be their $p_{\perp}^2$. If this virtuality is small as compared to the virtuality of the photon probe Q^2, the q$\bar{\text{q}}$ pair has lived long enough to be treated as independent and the event is considered to be an ordinary sea–quark interaction according to the discussion above. Otherwise if $p_{\perp}^2$ is larger than Q^2 (multiplied by some factor, see description of `PARA(20)` in section 3.4) both quarks are given a momentum transfer, creating one dipole each with the hadron remnant which will radiate independently.

2.9. Hadron–hadron collisions

In principle it is straightforward 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 subprocess as point-like and all others as extended.

One problem arises when there is a gluon coming out of the hard subprocess. The invariant $p_{\perp}^2$ of this gluon must limit the subsequent gluon emission to avoid double counting.

Another problem arises in the case of Drell–Yan production. In the case of e.g. q$\bar{\text{q}}$ → 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 momentum from the initial state which has been observed in e.g. p$\bar{\text{p}}$ collisions[15].

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, however, the program uses 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 to 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 `'DELPHI'` or `'OPAL'` to use the tuning of refs. [16] and [4], respectively. Note that `ARTUNE` also changes some parameters and switches in JETSET (`PARJ(21)`, `PARJ(41)`, `PARJ(42)` and `MSTJ(11)`).

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

3.2. A jet clustering routine

In addition to the generation of dipole emission, Ariadne also provides a routine for jet-clustering called `ARCLUS`. It implements a CDM inspired jet algorithm 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 [18] or some mutual $p_\perp$ as in the LUCLUS algorithm [6]. 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.

The algorithm used in `ARCLUS` is different in the sense that it looks at all combinations of *three* jets, looking at the invariant $p_\perp^2$ of one with respect to 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 eq. (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 [17] 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/`.

Note that this algorithm is poorly optimized for speed. It will hopefully become faster in coming revisions.

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 /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 = 0.2 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 = 1.0 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 = 1.0 GeV) The cut-off in invariant $p_\perp$ for emissions from electro-magnetic dipoles.

`PARA(6)` (D = -1.0 GeV) If larger than zero this gives the maximum allowed invariant $p_\perp$, 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 = 1.0 GeV) Soft suppression parameter μ for partons with `K(I,4)=1`.

`PARA(12)` (D = 1.0 GeV) Soft suppression parameter μ for partons with `K(I,4)=2`.

`PARA(13)` (D = 1.0 GeV) Soft suppression parameter μ for partons with `K(I,4)=3`.

`PARA(14-19)` Not used.

`PARA(20)` (D = 1.0) 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(21-30)` Not used.

`PARA(31)` (D = 1.0 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 which Ariadne is run on.

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)` (R) Flag set by `ARINIT` to indicate that initialization has been done.

`MSTA(3)` (D=0) Setting of parameters in Ariadne, JETSET, PYTHIA and LEPTO to suitable values in `ARINIT`.
0 Off.
1 On.

`MSTA(4)` (R) Number of calls to `AREXEC` so far.

`MSTA(5)` (D=0) Performs 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)` (R) 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.

`MSTA(15)` (D=5) Maximum number of flavors allowed in $q\bar{q}$ emissions.

`MSTA(16)` (D=2) Recoil strategy:
0 Use eq. (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=2) 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 eq. (14).

`MSTA(18)` (D=1) $p_\perp$-ordering of recoil gluons.
0 Off.
1 On.

`MSTA(19)` (D=1) Treatment of emissions from heavy quarks.
0 Simple treatment, changing the denominator in eqs. 1, 2 and 15 to

$$\left(1-x_1+\frac{m_1^2-m_3^2}{S_{\text{dip}}}\right)\left(1-x_3+\frac{m_3^2-m_1^2}{S_{\text{dip}}}\right) \tag{17}$$

1 A more elaborate treatment taking into account the "dead-cone" effect in ref. [19].

`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 2.7).

`MSTA(21-23)` Not used.

`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(25-29)` Not used.

`MSTA(30)` (D=1) 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 $\mu = Q$.

`MSTA(31)` (D=1) Treatment of masses of extended partons.
0 Make extended partons massless (for compatibility with previous versions).
1 Extended partons allowed to be massive.

`MSTA(31-40)` Not used.

3.5. 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 has 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 does not 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.

3.5.1. 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.

3.5.2. 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(0,11,30.0,-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.

3.5.3. 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+',17000.0)

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 subprocesses available in PYTHIA, and attempts to use Ariadne for other subprocesses will result in a warning from Ariadne.

Appendix A. 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

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

which is by default commented out. Removing these comments (globally replacing "`CD    `" with "`      `") will avoid precision problems but may also severely reduce the speed of the program on some machines.

The internal event record of Ariadne has a maximum number of partons, dipoles and strings which it can handle. These numbers 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 7.1 or later of the JETSET program. In addition when run in `'LEPTO'` mode it should be loaded together with version 6.1 or later of the LEPTO program and when run in `'PYTHIA'` mode together with version 5.3 or later of the PYTHIA program.

A.1. Availability

The program is available on E-mail request from the author (`lonnblad@apollo3.desy.de`). The program will then be sent as an E-mail message containing the latest revision of the code together with the latest revision of this manual in LaTeX format.

The program is also available via anonymous ftp to `thep.lu.se` (IP number `130.235.92.57`). Here the program resides in the directory `pub/LundPrograms` as the compressed tar-archive file `ariadne- 4.01.tar.Z` (Forthcoming revisions will be numbered `4.02`, `4.03` etc.)

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 via anonymous ftp, the compressed tar-file should be "un-compressed" and "un-tarred" which will create a directory called `ariadne-4.01`. 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.tex`, `ariadne.man` and `ariadne.ps` containing the latest revision of this manual in LaTeX, ASCII and Postscript format, 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 10 000 events randomly distributed in center of mass energy and check their consistency with respect to 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 3.5 are also included as the files `jtest.f`, `ltest.f` and `ptest.f` including dummy routines for parameter settings and analysis.

Appendix B. Description of subroutines and common blocks

This is a 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 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 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 ARDUMP`
Copies a partonic state from the internal event record to `/LUJETS/`.

`SUBROUTINE ARDUPH`
Copies a photon radiated by Ariadne to `/LUJETS/`.

`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 subprocess in `PYTHIA` is not supported by Ariadne.
21 `ARCLUS` was not able to order jets in energy due to lack of space.

`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 massless if extended.

`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 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'`.

`REAL FUNCTION ARIPT2(I1,I2,I3)`
Returns the invariant $p_\perp^2$ of parton `I2` with respect to 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 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.

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

`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`.

`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 ARSPLG(IG,IFLAV)`
Splits the gluon entry `IG` into a quark and an antiquark 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.

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

`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 `'DELPHI'` or `'OPAL'` to use the tuning of refs. [16] and [4] respectively.

`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.

`REAL FUNCTION ARVET1(), ARVET2()` etc.
Different routines for calculating the veto factor to be used by `ARMCDI`.

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),
$       IEX(MAXPAR),QQ(MAXPAR),IDI(MAXPAR),
$       IDO(MAXPAR),INO(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`.

`IEX(I)` indicates if parton `I` is to be considered 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.

`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),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.

`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.

References

[1] G. Gustafson, Phys. Lett. B 175 (1986) 453.
G. Gustafson and U. Pettersson, Nucl. Phys. B 306 (1988) 746.

[2] B. Andersson, G. Gustafson, L. Lönnblad and U. Pettersson, Z. Phys. C 43 (1989) 621.
[3] B. Andersson, G. Gustafson and L. Lönnblad, Nucl. Phys. B 339 (1990) 393.
[4] OPAL collaboration, M.Z. Akrawy et. al., Phys. Lett. B 246 (1990) 285.
[5] N. Magnussen, Generators for deep inelastic scattering, in: Proc. Physics at HERA Workshop, Hamburg, October 1991, to be published.
[6] T. Sjöstrand, JETSET 7.3 program and manual, see e.g. B. Bambah et. al., QCD Generators for LEP, CERN-TH.5466/89.
T. Sjöstrand, Comput. Phys. Commun. 39 (1986) 347.
T. Sjöstrand and M. Bengtsson, Comput. Phys. Commun. 43 (1987) 367.
[7] G. Ingelman, LEPTO 6.1 Program and Manual, Private communication, to be published.
[8] H.-U. Bengtsson and T. Sjöstrand, PYTHIA 5.5 Program and Manual.
H.-U. Bengtsson and T. Sjöstrand, Comput. Phys. Commun. 46 (1987) 43.
[9] U. Pettersson, ARIADNE – A Monte Carlo for QCD Cascades in the Colour Dipole Formulation, Lund Preprint LU-TP 88-5 (1988).
[10] L. Lönnblad and U. Pettersson, ARIADNE 2, Lund Preprint LU-TP 88-15 (1988).
[11] L. Lönnblad, ARIADNE 3, Lund Preprint LU TP 89-10 (1989).
[12] R. Kleiss, Phys. Lett. B 180 (1986) 400.
[13] B. Andersson and L. Lönnblad, to be published.
[14] L. Lönnblad, Photon radiation in the dipole model and in the Ariadne program, Preprint DESY 92-032; in: Proc. Workshop Photon Radiation from Quarks, Annecy, France, December 1991, to be published.
[15] UA1 collaboration, G. Arnison et. al., Lett. Nuovo Cimento 44 (1985) 1.
UA2 collaboration, J.A. Appel et. al., Z. Phys. C 30 (1986) 1.
[16] W. de Boer and H. Fuerstenau, DELPHI data in comparison with QCD models..., IEKA-KA/91-07 (July 1991).
[17] B. Andersson and G. Gustafson, Z. Phys. C 3 (1980) 223.
B. Andersson, G. Gustafson, G. Ingelman and T. Sjöstrand, Phys. Rep. 97 (1983) 31.
[18] JADE Collaboration, W. Bartel et. al., Z. Phys. C 33 (1986) 23.
S. Bethke, Habilitation thesis, LBL 50-208 (1987).
[19] Yu.L. Dokshitzer, V.A. Khoze and S.I. Troyan, Proc. 6th Int. Conf. Physics in Collision, ed. M. Derrick (World Scientific, Singapore, 1987).