LU-TP 97-34
December 1997

LDCMC version 1.0

Hamid Kharraziha
Dept. of Theoretical Physics
Sölvegatan 14a
S-223 62 Lund, Sweden
hamid@thep.lu.se

Leif Lönnblad
NORDITA
Blegdamsvej 17
DK-2100 København Ø, Denmark
leif@nordita.dk

Abstract

We present the LDCMC program implementing the Linked Dipole Chain model for deeply inelastic ep scattering within the framework of the ARIADNE event generator.

1 Introduction

The large energies of the colliding positron and proton at the HERA accelerator enables a detailed study of the proton structure. Data is available for a wide kinematical region from very large virtualities Q^2 of the photon probe, to very small values of the Bjorken x-variable. We will here present a Monte Carlo event generator which covers the whole of this kinematical region and gives a theoretically well founded description of both the proton structure and the hadronic final state properties in the collision events.

Theoretical description of the proton structure has since long been available through the DGLAP evolution equations [1], valid for large values of Q^2, and the BFKL equations [2] which cover the very soft part of the proton structure (the $x \to 0$ limit). A third theoretical approach has been suggested by Ciafaloni, Catani, Fiorani and Marchesini and is referred to as the CCFM model [3]. This approach covers both the DGLAP and BFKL regions of validity and gives, unlike the DGLAP and BFKL equations, also a description of the final states produced in the collision process.

Recently, the CCFM model was reformulated into a new model, called the Linked Dipole Chain (LDC) model [4]. This reformulation leads to a considerable simplification of the formalism and provides a recipe for a Monte Carlo simulation. One assumes that some initial partons are produced with a weight distribution (given by the CCFM model) which provides the contribution to the total cross-section. These initial partons stretch up a chain of linked colour dipoles and more partons are emitted through a final-state Bremsstrahlung process which was initially developed for the description of parton cascades in hadronic e^+e^- events. The event generator presented here is based on the LDC model.

The CCFM model, and consequently also the LDC model, are approximate models valid to the leading log accuracy. One problem that was encountered in our development of the LDCMC event generator was to consistently include some sub leading effects into the formalism. Among these corrections are energy conservation and the description of quarks, both crucial for a realistic event generator. A set of assumptions has been made concerning these corrections, but the program provides some options to our default strategy, making it possible for the user to compare their effect on the results.

In section 2 we give a further description of the LDC model. The components of the program are presented in section 3 and finally a sample program is found in section 4.

You will find a more detailed description of the physics behind this event generator and some results in ref. [5]. The LDC model is presented in ref. [4] and some leading-log results and asymptotic solutions are described in refs. [6, 7]. For a full

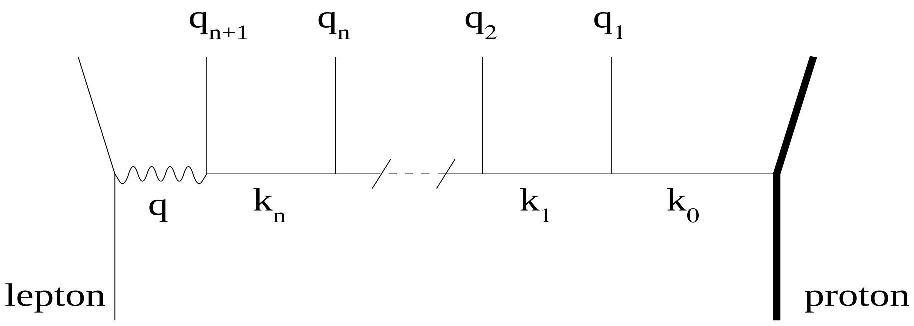

Figure 1: *Fan diagram for a DIS event with n initial emissions. q_i denote the emitted ISB partons, k_i denote the propagators.*

description of the CCFM model we refer to ref. [3], but a brief description of their results and the connection to the LDC model can be found in ref. [4].

2 The LDC Model

The Linked Dipole Chain model is a reformulation of the CCFM model. Possible parton emissions in DIS are separated into two sets which are denoted Initial State Bremsstrahlung (ISB) and Final State Bremsstrahlung (FSB), respectively. The total cross-section or the structure function is evaluated by providing a weight distribution $d\omega_I$ for all possible ISB-chains:

$$F_2 = \left(\sum d\omega_I\right) \otimes f_0, \tag{1}$$

where f_0 denotes the input parton distribution. The requirement on the FSB is that recoils on the initial chain is negligible and also that their contribution to the cross-section is of Sudakov type, that is, these emissions can be added without affecting the total cross-section.

In the fan-diagram in fig. 1, describing a DIS-event with n emissions, the ISB partons are denoted $\{q_i\}$ and the propagators are denoted $\{k_i\}$. Let further $z_{i\pm}$ be the fraction of the positive/negative light-cone momentum (positive is defined in the proton direction) carried by the continuing propagator in each emission, $z_{i+} \equiv k_{i+}/k_{(i-1)+}$ and $z_{i-} \equiv k_{i-}/k_{(i+1)-}$.

In the CCFM model, the ISB set is defined by requiring that the emissions are ordered in rapidity (angular ordering) and energy. A further requirement is introduced in the LDC model by a constraint on the transverse momenta in each emission

$$q_{\perp i} > \min(\mathrm{k}_{\perp \mathrm{i}}, \mathrm{k}_{\perp \mathrm{i}-1}) \tag{2}$$

This means a redefinition of the ISB-FSB sets by moving ISB emissions which violates this constraint into the FSB set.

The ISB kinematical regions is then determined by the constraint in eq. (2) plus ordering in positive and negative light-cone momenta

$$q_{-} > q_{n-} > \cdots > q_{1-} \qquad q_{n+} < \cdots < q_{1+} < P_{+} \tag{3}$$

with the outer limits set by the proton (P) and photon (q) momenta. The weight distribution for the ISB-chains factorizes into one factor for each emission and is given by

$$\begin{aligned} d\omega_I &= d\omega_1 d\omega_2 \cdots d\omega_n \\ d\omega_i &= \bar{\alpha}\frac{d^2\vec{q}_{\perp i}}{q_{\perp i}^2}\frac{dz_{i+}}{z_{i+}} \end{aligned} \tag{4}$$

with the effective coupling $\bar{\alpha} \equiv 3\alpha_s/\pi$.

This formalism is made symmetric with respect to the photon and proton directions and leads to evolution equations which work as a simultaneous evolution of the proton and the photon parton distributions. Consequently, it is within the LDC model, possible to, also describe resolved photon events in addition to normal DIS events where the probing photon is supposed to have the largest virtuality.

A further advantage with the reformulation of the ISB-FSB sets is that the FSB kinematical region, in the LDC model, is consistent with the colour Dipole model [8] for parton cascades. This means that, following the production of the ISB chain, the FSB partons can be emitted in a similar way as in e.g. hadronic e^+e^- events.

The emission phase space in the dipole model is usually plotted in a ($\ln p_\perp^2$, rapidity) plane and is given by a characteristic triangular area (see fig. 2a). In the LDC model one assumes that the initial ISB partons form a chain of linked colour dipoles which can radiate partons independently. For each dipole, the maximum possible transverse momentum is set by the virtuality of the corresponding propagator. The FSB kinematical region is then given by the shaded area in fig. 2b.

The CCFM model is correct only to leading log accuracy so some sub-leading corrections has been included in the event generator to be able to make more realistic predictions:

- Energy and momentum is conserved at every vertex.
- Quarks are introduced and some sub-leading terms are included by using standard Altarelli-Parisi splitting functions and full $2 \to 2$ matrix elements when appropriate.

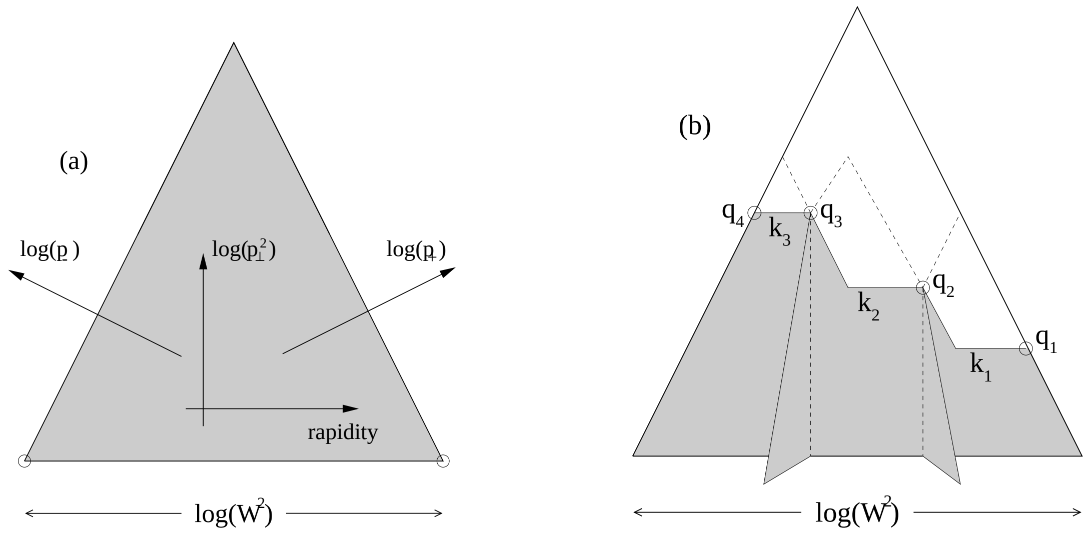

Figure 2: *(a) The dipole phase space for gluon emission from a $q-\bar{q}$ pair with central mass energy W. (b) The dipole phase space in a DIS event. The shaded are is the kinematical region for FSB.*

- The ISB emission closest to the electro-weak vertex is corrected with $\mathcal{O}(\alpha\alpha_s)$ matrix elements.

The momenta of the ISB partons $\{q_i\}$ are determined by generating the values of the positive $\{k_{i+}\}$ and negative $\{k_{i-}\}$ light-cone momenta for the propagators and the azimuthal angles $\{\phi_i\}$ of the emitted ISB partons. All other momentum components in the ISB chain are then determined by energy momentum conservation at each vertex.

The second correction is obtained by introducing generalised splitting functions $P^{k^2_{i+1},k^2_i,k^2_{i-1}}_{f_{i+1},f_i,f_{i-1}}(z)$. These are defined for every propagator link k_i and are determined by the flavours (f) and the ordering of the virtualities of the three successive propagators, k_{i-1}, k_i and k_{i+1}. The function variable is either $z = z_{i+}$ or $z = z_{i-}$.

In cases when the virtualities are uniformly ordered, $-k^2_{i-1} < -k^2_i < -k^2_{i+1}$ or $-k^2_{i-1} > -k^2_i > -k^2_{i+1}$, the generalised splitting functions are given by standard Altarelli- Parisi splitting functions with $z = z_{i+(-)}$ in the first (second) case. When the intermediate virtuality is larger than the others $(-k^2_{i-1} < -k^2_i > -k^2_{i+1})$, the process is viewed as a local sub-collision and the generalised splitting function is determined by the corresponding first order matrix element. Finally, if the intermediate virtuality is the smallest $(-k^2_{i-1} > -k^2_i < -k^2_{i+1})$, only the leading $1/z$ term of the standard Altarelli-Parisi splitting functions is used with $z = z_{i+}$ if $k^2_{i-1} < -k^2_{i+1}$ and $z = z_{i-}$ otherwise.

The colour factors are removed from the splitting functions. Instead, the colour factor for the whole ISB ladder is calculated separately, since it is (to leading order) independent of the kinematics.

The results of the CCFM model, as formulated in the LDC model, is intuitively quite easily comprehensible.

As we move in rapidity along the initial chain, the virtuality of the propagators can alternatively increase or decrease. In the DGLAP evolution formalism, the contribution to the total cross-section, or the structure function, is calculated with the assumption that the virtualities increase all the way to the photon end and that the photon has the largest virtuality. Unordered chains are shown to be suppressed. However, for events with very small x-values and moderately large Q^2, they are expected to have a significant contribution.

Generally one can subdivide the initial chain into regions with monotonically increasing or decreasing virtualities (see fig. 3). In ref. [7] it is shown that the structure function factorizes into one weight for each such region and that these weight factors are in principle given by the (leading order) DGLAP structure function. Regions with increasing virtuality, can be treated as an ordered (DGLAP) evolution of the proton parton distribution, while the regions with decreasing virtuality can be treated as ordered evolution of the photon parton distribution. These regions with ordered virtualities are then connected with propagators which have a virtuality which is either a local maximum, and can be treated as a local sub-collision, or a local minimum.

It then follows naturally that some sub-leading corrections can be included by using the Altarelli-Parisi splitting functions in the ordered regions and by using $2 \rightarrow 2$ matrix element for the local sub-collisions.

3 Program Components

The LDCMC program is logically divided into three parts. The first part generates chains to leading log accuracy given an x, Q^2, flavour of the struck quark and input parton density functions. The second part generates the exact kinematics taking into account quark masses and exact energy-momentum conservation in each branching. Finally, the third part handles the colour connections between partons and prepares the chains for subsequent final-state gluon emissions and hadronization. The detailed formulas used in the different parts are described in ref. [5], here we only describe some technical aspects of the implementation.

The first leading log part selects the flavour and x_0 of an input parton according

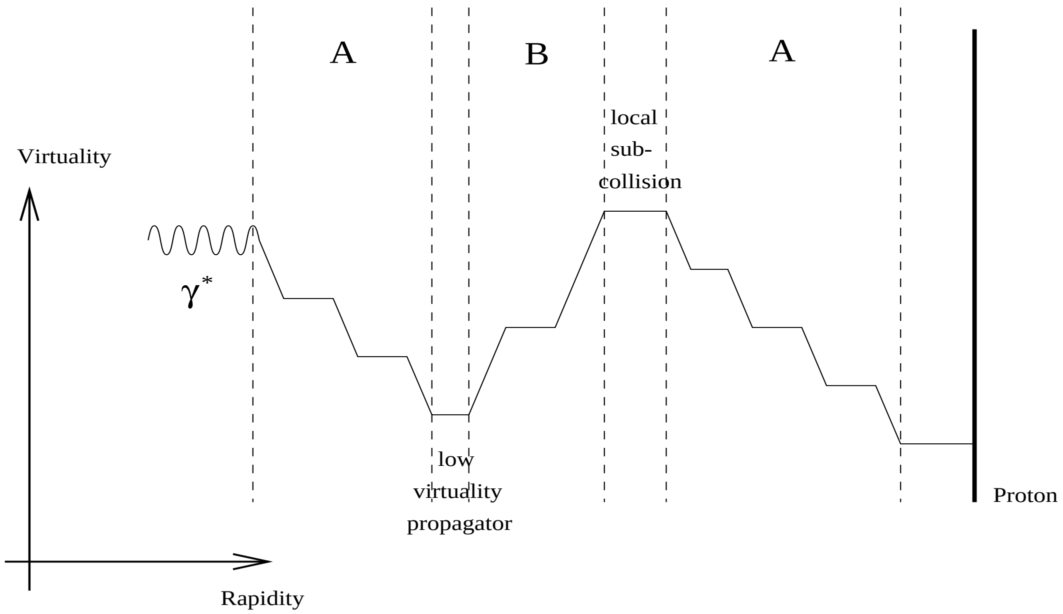

Figure 3: *A general DIS-event can be separated into regions with ordered virtualities. These regions are connected either with a local sub-collision or a propagator with low virtuality. The weight factor for the regions A (B) is given by a DGLAP evolution of the proton (photon) parton distribution.*

to a approximate upper limit of the evolved parton density

$$
\begin{aligned}
xf_i(x,Q^2) \;=\; \sum_j \int_x^1 \frac{dx_0}{x_0}\,[\; & G(Q^2/k_{\perp 0}^2, x/x_0) + \\
& S_j(Q^2,k_{\perp 0}^2)\delta_{ij}(\ln x - \ln x_0)]\, x_0 f_{0j}(x_0,k_{\perp 0}^2),
\end{aligned} \tag{5}
$$

with

$$
\begin{gathered}
G(Q^2/k_{\perp 0}^2, x/x_0) = \sqrt{\tfrac{a}{b}}I_1(2\sqrt{ab}) \\
a = \sqrt{\bar\alpha}(\ln Q^2/k_{\perp 0}^2 + \ln x_0/x), \quad b = \sqrt{\bar\alpha}\ln x_0/x.
\end{gathered} \tag{6}
$$

This is done numerically, and for speed considerations the actual integration is done only once per struck flavour and per bin in x and Q^2. Then the number of emissions and the z_+ and z_- of each splitting is generated according to

$$
\sqrt{\frac{a}{b}}I_1(2\sqrt{ab}) = \sum_{n=1}^{\infty}\frac{a^n b^{n-1}}{n!(n-1)!}, \tag{7}
$$

and

$$
\frac{a^n b^{n-1}}{n!(n-1)!} = \int \bar\alpha^n \Pi_j \frac{dz_{j+}}{z_{j+}}\frac{dz_{j-}}{z_{j-}}\delta(\ln x_0 + \sum_j \ln z_{j+} - \ln x). \tag{8}
$$

After this, the flavour combination in all splittings is done according to simplified splitting functions.

In the second part, the exact kinematics of each splitting is performed, generating the azimuth angle in each splitting according to a flat distribution and taking into account quark masses and ensuring that the condition in eq. (2) is satisfied. Then it is checked that all virtualities and transverse momenta are above the cutoff and a number of corrections are calculated to obtain a running α_s, correct splitting functions, the relevant Sudakov form factors and a correct matching to the full $\mathcal{O}(\alpha\alpha_s)$ matrix elements.

The produced chain has now a weight, which normally is below one, and a veto algorithm is used to obtain unweighted events. Sometimes, however, the weight may be above one, and in that case the chain is saved and may be reused the next time an event with the same flavour, is requested in the same x and Q^2 bin. Alternatively the weights can be rescaled with a common factor to avoid such situations. Finally the partons along the chain is colour connected and limits for the subsequent FSB is determined and passed together with the chain back to the calling Fortran routine in ARIADNE.

The whole procedure for generating a complete event is then as follows:

- Generate x, Q^2 and the flavour of the struck quark using the electro-weak matrix element and evolved structure functions in the LEPTO [9] program. The LDC distribution contains a replacement for the PYSTFU routine of PYTHIA [10] which is called from LEPTO to enable the usage of structure functions evolved with the LDC model.
- This information is passed to the ARIADNE [11] program which internally calls the LDCMC code to generate the the initial-state chains.
- ARIADNE then performs the final-state dipole cascade and optionally calls JETSET [10] to perform the subsequent hadronization.

There are a number of different options and parameters influencing the generation which are accessible to the user via Fortran common blocks. Most of these are in the `LDCDAT` common block defined as follows:

```
COMMON /LDCDAT/ MSTLDC(100),PARLDC(100)
```

`MSTLDC(1)` (Default = -1) Maximum number of flavours used in the generation. If negative, the corresponding ARIADNE number from `MSTA(15)` in common block `ARDAT1` is used.

`MSTLDC(2)` (D = -1) If positive, use running $\alpha_s(k_\perp^2) = \frac{\pi}{3}\bar{\alpha}/\log(k_\perp^2/\Lambda_{\mathrm{QCD}}^2)$ otherwise use constant $\alpha_s = \frac{\pi}{3}\bar{\alpha}$. $\bar{\alpha}$ and Λ_{QCD} are given by `PARLDC(1)` and `PARLDC(3)` below. If negative, the corresponding ARIADNE switch `MSTA(12)` in common block `ARDAT1` is used.

`MSTLDC(3)` (D = 1) If positive, use full $\mathcal{O}(\alpha\alpha_s)$ matrix elements from [12] for the splitting closest to the virtual photon.

`MSTLDC(4)` (D = 0) If positive, only allow chains with monotonically increasing virtualities from the proton side.

`MSTLDC(5)` (D = 0) If positive, only allow chains with monotonically increasing virtualities from the proton side, except the link closest to the virtual photon may have a virtuality larger than Q^2.

`MSTLDC(6)` (D = 0) If positive, do not allow any links with virtuality above Q^2.

`MSTLDC(7)` (D = 5) Number of bins per unit in $\log^{10}(x)$.

`MSTLDC(8)` (D = 5) Number of bins per unit in $\log^{10}(Q^2)$.

`MSTLDC(9)` (D = 1) If zero, do not allow $q \to q$ or $g \to q$ splittings (except $g \to q$ closest to the photon) in the initial chain.

`MSTLDC(11)` (D = 0) If positive, allow some links with virtuality below cutoff, as long as both neighbouring links are above.

`MSTLDC(12)` (D = 1) Choice of Sudakov form factor. If zero, no form factor is used. If non-zero, use a form factor of the form

$$\ln S = -\int_0^{z_{\mathrm{cut}}} dz \int \frac{dq_\perp^2}{q_\perp^2}\frac{\alpha_s(q_\perp^2)}{2\pi} P_{q\to q}(z) \tag{9}$$

for quarks and

$$\ln S = -\int_{0.5}^{z_{\mathrm{cut}}} dz \int \frac{dq_\perp^2}{q_\perp^2}\frac{\alpha_s(q_\perp^2)}{2\pi} P_{g\to g}(z) - \int_0^1 dz \int \frac{dq_\perp^2}{q_\perp^2}\frac{\alpha_s(q_\perp^2)}{2\pi} P_{g\to q}(z) \tag{10}$$

for gluons. If negative, the lower limit in the $q_\perp^2$ integration is just the virtuality of the parton, while if `MSTLDC(12)` is positive, this virtuality is multiplied by a factor $\exp(0.4)$ to approximate the phase space constraint in eq. (2). In both cases the upper limit is given by the highest virtuality in an emission step.

`MSTLDC(13)` (D = 0) If positive, do not allow gluon emission with $z > 0.5$ if the virtuality is increasing also in next step, even if `PARLDC(8)` is larger than 0.5.

`PARLDC(1)` (D = -1.0) $\bar{\alpha}$. If negative, the value used is 36/23.

`PARLDC(2)` (D = -1.0) Cutoff in $k_\perp$. If negative, the value is taken from the corresponding ARIADNE parameter (`PARA(3)` in `ARDAT1`).

`PARLDC(3)` (D = -1.0) Λ_{QCD}. If negative, the corresponding ARIADNE number from `PARA(1)` in common block `ARDAT1` is used.

`PARLDC(4)` (D = 1.0) The factor dividing the generated weights to ensure that all weights are below one (to allow for unweighted events).

`PARLDC(6)` (D = 0.2) Effective $\bar{\alpha}$ used in the leading-log part of the generation for speed considerations. The actual $\bar{\alpha}$ used is the one in `PARLDC(1)`.

`PARLDC(8)` (D = 0.5) The upper cut in z used in the $q \to q$ and $g \to g$ splitting functions to avoid divergences as $z \to 1$.

The following switches in the ARIADNE common block `ARDAT1` are relevant also for the LDCMC:

`MSTA(12)` (D = 1) If positive, use running α_s. Cf. `MSTLDC(2)`.

`MSTA(15)` (D = 1) Maximum number of flavours used in the generation. Cf. `MSTLDC(1)`.

`MSTA(32)` (D = 2) Handling of DIS events in ARIADNE when run with LEPTO. If set to -32, LDCMC is used.

`PARA(1)` (D = 0.22 GeV) The Λ_{QCD} used in the running of α_s. Cf. `PARLDC(3)`.

`PARA(3)` (D = 0.6 GeV) The cutoff in $k_\perp$. Cf. `PARLDC(2)`.

4 Sample Program

The main part of the LDCMC is written in C++ and therefore special care has to be taken when installing the program to ensure that the interface between the C++ and Fortran code is working properly. In the distribution there are detailed instructions on how to install the program in the `README` file and here we will only discuss the main points.

The C++ code relies heavily on the *Standard Template Library* so the installation requires a C++ compiler which supports this. The C++/Fortran interface assumes that the Fortran compiler appends an underscore to all external symbols and converts them to lower-case characters, ie. a Fortran subroutine declared as

`SUBROUTINE HERA` will be declared as `void hera_()` in the C++ code. The file `f77hack.h` contains *typedefs* for `Fint`, `Fdouble` and `Ffloat` and the user must ensure that these types corresponds to the Fortran equivalents `INTEGER`, `DOUBLE PRECISION` and `REAL` respectively.

The distribution includes a sample program to check that the installation was successful, and to serve as a template for user customized runs. It consists of a C++ main program which must be linked together with the Fortran steering routines and the LDCMC, Lepto, Pythia, Ariadne and Jetset libraries (in that order) as well as with the Fortran runtime libraries.

The C++ main program looks like this:

```
#include <fstream.h>
#include "EvolvedDensities.h"

// Declare the F77 subroutine as extern.
extern "C" {
  void hera_();
}

int main() {

  // First read in structure function parametrization from file.
  ifstream evolved("xf-LDC-97.A");
  EvolvedDensities::readInputFrom(evolved);
  EvolvedDensities::readFrom(evolved);
  evolved.close();

  // Call F77 subroutine.
  hera_();

  // Done.
  return 0;
}
```

and the corresponding Fortran steering routines looks like this:

```
      SUBROUTINE HERA

C...Initialize parameters in Ariadne and Jetset
      CALL ARTUNE('4.10')

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=1,10

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

      SUBROUTINE LEPSET

      COMMON /LEPTOU/ CUT(14),LST(40),PARL(30),X,Y,W2,XQ2,U
      SAVE /LEPTOU/
      COMMON /PYPARS/ MSTP(200),PARP(200),MSTI(200),PARI(200)
      SAVE /PYPARS/

C...Use structure functions from Pythia
      LST(15)=0

C...Or rather use the LDC version of pystfu giving the structure
C...functions evolved with LDC as read from the file in the C++
C...main program (Make a dummy call to pystfu to ensure correct
C...version of pystfu is linked.)
      IF (MSTP(51).EQ.940801) CALL PYSTFU
      MSTP(51)=0

C...Switch off hadronization
      LST(7)=0

C...Suppress printouts from LEPTO
      LST(3)=0

C...Set x, Q2 and W2 ranges in Lepto
      CUT(1) = 0.00008
      CUT(2) = 0.5
      CUT(5) = 5.0
      CUT(6) = 1280.0
      CUT(7) = 1000.0

      RETURN
      END

      SUBROUTINE ARISET

      COMMON /ARDAT1/ PARA(40),MSTA(40)
      SAVE /ARDAT1/
      COMMON /ARHIDE/ PHAR(400),MHAR(400)
      SAVE /ARHIDE/
      COMMON /LDCDAT/ MSTLDC(100),PARLDC(100)

C...Master switch for using LDC in Ariadne
      MSTA(32)=-32

C...Some parameters in LDCMC
      PARLDC(4)=1.0
      PARLDC(6)=0.2
```

```
      RETURN
      END

      SUBROUTINE HERANA

C...Trivial analysis - print the event
      CALL LULIST(1)

      RETURN
      END
```

If the installation was successful, running this program should print ten reasonable HERA events on the parton level (rather than producing a core dump).

The distribution also contains a selection of sample parton density function files containing input parton density parametrizations and tabulated densities evolved with the LDC model. The files are named `xf-LDC-97.A` to `xf-LDC-97.H`, where the last letter indicates the strategy used in the fitting in [5]:

A LDC default.

B DGLAP: As for A but only allow chains with monotonically increasing virtualities of the links from the proton side. `MSTLDC(4)=1`.

C DGLAP': As for B, but chains where the virtuality of the link closest to the virtual photon is larger than Q^2 are permitted. `MSTLDC(4)=0`, `MSTLDC(5)=1`.

D As for A but $k_{\perp 0} = 1$ GeV. `PARLDC(2)=1`.

E As for A but without the Sudakov form factors. Instead $P_{q\to q}(z)$ is set to zero and $P_{g\to q}(z)$ nonzero only in the splitting closest to the photon. `MSTLDC(9)=0`, `MSTLDC(12)=0`.

F As for A but $\beta_g = \beta_S = 5$ in the input density parametrization.

G As for A but only fitted to F_2 data with $x < 0.1$.

H As for A but allow the virtuality of some links to be below $k_{\perp 0}$ as long as the largest virtuality of two consecutive links always is above $k_{\perp 0}$. `MSTLDC(11)=1`.

The LDCMC program is available as a part of the Ariadne program from the Ariadne web pages `http://www.nordita.dk/~leif/ariadne/`.

References

[1] V.N. Gribov, L.N. Lipatov, *Sov. J. Nucl. Phys.* **15** (1972) 438 and 675;
G. Altarelli, G. Parisi, *Nucl. Phys.* **B126** (1977) 298;
Yu.L. Dokshitzer, *Sov. Phys. JETP* **46** (1977) 641.

[2] E.A. Kuraev, L.N. Lipatov and V.S. Fadin, *Zh. Eksp. Teor. Fiz.* **72** (1977) 373, *Sov. Phys. JETP* **45** (1977) 199;
Ya.Ya. Balitsky and L.N. Lipatov, *Yad. Fiz.* **28** (1978) 1597, *Sov. J. Nucl. Phys.* **28** (1978) 822.

[3] M. Ciafaloni, *Nucl. Phys.* **B269** (1988) 49;
S. Catani, F. Fiorani, G. Machesini, *Phys. Lett.* **B234** (1990) 339, *Nucl. Phys.* **B336** (1990) 18.

[4] B. Andersson, G. Gustafson, J. Samuelsson, *Nucl. Phys.* **B463** (1996) 217.

[5] H. Kharraziha, L. Lönnblad, LU-TP 97-21, NORDITA-97/54 P, hep-ph/9709424.

[6] B. Andersson et al., *Z. Phys.* **C71** (1996) 613.

[7] B. Andersson, G. Gustafson and H. Kharraziha,
LU-TP 97-29, aps1997nov19_001, hep-ph/9711403.

[8] G. Gustafson, *Phys. Lett.* **B175** (1986) 453;
G. Gustafson, U. Petterson, *Nucl. Phys.* **B306** (1988) 746;
B. Andersson, G. Gustafson, L. Lönnblad, *Nucl. Phys.* **B339** (1990) 393.

[9] G. Ingelman, A. Edin and J. Rathsman, *Comput. Phys. Comm.* **101** (1997) 108, hep-ph/9605286.

[10] T. Sjöstrand, *Comput. Phys. Comm.* **82** (1994) 74

[11] L. Lönnblad, *Comput. Phys. Comm.* **71** (1992) 15.

[12] R.D. Peccei, R. Rückl, *Nucl. Phys.* **B162** (1980) 125.