Computer Physics Communications 70 (1992) 97–119
North-Holland

Computer Physics
Communications

KRONOS – A Monte Carlo event generator for higher order electromagnetic radiative corrections to deep inelastic scattering at HERA *

Harald Anlauf [a,1], Hans D. Dahmen [c], Panagiotis Manakos [a], Thomas Mannel [a,b,2] and Thorsten Ohl [a,3]

[a] *Technische Hochschule Darmstadt, W-6100 Darmstadt, Germany*
[b] *Deutsches Elektronensynchrotron, W-2000 Hamburg, Germany*
[c] *Universität Siegen, W-5900 Siegen, Germany*

Received 1 October 1991

We present the Monte Carlo even generator KRONOS for deep inelastic lepton–hadron scattering at HERA. KRONOS focusses on the description of electromagnetic corrections beyond the existing fixed order calculations.

PROGRAM SUMMARY

Title of program: KRONOS

Catalogue number: ACHC

Program obtainable from: CPC Program Library, Queen's University of Belfast, N. Ireland (see application form in this issue)

Licensing provisions: none

Computer: (i) IBM RS/6000, (ii) IBM 3090, (iii) IBM 9000

Operating system: (i) AIX, (ii) MVS/XA, (iii) MVS/ESA

Programming language used: FORTRAN-77

Memory required to execute with typical data: 800k words

No. of bits in a word: 32

Number of lines in distributed program, including test data, etc.: 5307

CPC Program Library subprograms used: PAKPDF (parton distribution functions) [1]; catalogue number: ACGU.

Keywords: radiative corrections, deep inelastic scattering, Monte Carlo event generator, multiphoton radiation

Nature of physical problem
Higher order leading logarithmic QED radiative corrections to high energy scattering of leptons off protons.

Method of solution
Monte Carlo event generation.

Restrictions on the complexity of the problem
Final state radiation and radiation from the incoming quarks are not included.

Typical running time
≈ 1 ms/event, depending on the chosen cuts.

Reference
[1] K. Charchuła, Comput. Phys. Commun. 69 (1992) 360.

* Supported by Bundesministerium für Forschung und Technologie, Germany.
1 Supported in part by Studienstiftung des Deutschen Volkes. E-mail: td15@ddagsi3.bitnet.
2 E-mail: t00man@dhhdesy3.bitnet.
3 Present address: DESY, W-2000 Hamburg, Germany. E-mail: t00ohl@dhhdesy3.bitnet.

LONG WRITE-UP

1. Introduction

At the new ep collider HERA [1,2], a new kinematic range for deep inelastic lepton–hadron scattering can be explored. One of the main tasks for HERA will be a detailed measurement of the structure functions of the proton in a wide range of momentum transfers up to $Q^2 \approx 10^4$ GeV^2 and for very small values of x down to 10^{-4} [3].

In order to extract the information on the structure functions, radiative corrections have to be taken into account. The electroweak corrections to deep inelastic ep scattering have been calculated completely to one loop order [4,5], and the leading logarithms of the next order have been given [6]. The order α corrections have been implemented in the HERACLES event generator [7].

However, it is well known that radiative corrections may become large at very high momentum transfers and a partial summation of higher order terms becomes mandatory. The leading contributions in processes involving the scattering of high energy electrons originate from logarithms of the form

$$\frac{\alpha}{\pi}\ln\left(\frac{Q^2}{m_e^2}\right) \approx 6.2\% \quad \text{for} \quad Q^2 \approx 10^5 \text{GeV}^2, \tag{1}$$

where m_e is the mass of the electron. In an appropriately chosen gauge [8], these terms may be expressed as collinear photon radiation off the incoming electron.

To sum leading logarithms of the form (1) one may use renormalization group techniques well known in QCD. These methods are directly applicable to QED inclusive radiative corrections where the outgoing photons are not individually detected. In the QED context this idea was first used by ref. [9] to estimate higher order radiative corrections in e^+e^- annihilation at LEP energies and later refined by ref. [10]. In this approach the cross-section with inclusive radiative corrections (i.e. none of the radiated photons is resolved) is obtained by folding the Born cross-section (the cross-section without radiative corrections) with a so-called radiator $e(\xi, Q^2)$,

$$\sigma(p) = \int_0^1 \mathrm{d}\xi\, e(\xi, Q^2)\, \sigma_{\mathrm{Born}}(\xi p), \tag{2}$$

where p is the momentum of the incoming electron, Q^2 is the hadronic momentum transfer in the process and ξ is the fraction of the initial electron momentum left after photon radiation.

The radiator $e(\xi, Q^2)$ is obtained as the solution of the evolution equation [8]

$$Q^2\frac{\partial}{\partial Q^2}e(\xi, Q^2) = -\frac{\alpha}{2\pi}\left[\int_0^{1-\epsilon}\mathrm{d}\zeta\, P(\zeta)\right]e(\xi, Q^2) + \frac{\alpha}{2\pi}\int_\xi^{1-\epsilon}\frac{\mathrm{d}\zeta}{\zeta}P(\zeta)\, e\left(\frac{\xi}{\zeta}, Q^2\right), \tag{3}$$

with the initial condition

$$e(\xi, m_e^2) = \delta(1-\xi) \tag{4}$$

and the splitting function

$$P(\zeta) = \frac{1+\zeta^2}{1-\zeta}. \tag{5}$$

The infrared regulator ϵ in eq. (3) is understood to have been chosen well below the experimental resolution for resolved photons.

Equation (3) is an explicitely regularized form of the familiar evolution equation

$$Q^2 \frac{\partial}{\partial Q^2} e(\xi, Q^2) = \frac{\alpha}{2\pi} \int_\xi^1 \frac{d\zeta}{\zeta} [P(\zeta)]_+ e\left(\frac{\xi}{\zeta}, Q^2\right), \tag{6}$$

where the "+"-distributions are defined by

$$\int_0^1 d\xi \, [P(\xi)]_+ f(\xi) = \int_0^1 d\xi \, P(\xi) \, (f(\xi) - f(1)). \tag{7}$$

However, aiming at the construction of an event generator for the analysis of HERA experiments, it is necessary to describe the individual photons. Since the renormalization group sums the leading logarithms (1), arbitrarily high orders in α are involved and thus a corresponding event generator must be capable of generating multiphoton final states.

One method of extending the renormalization group approach to exclusive photons uses the so called unitary approximation [11], treating each photon individually. It was shown to reproduce the renormalization group results for the inclusive quantities [12]. Furthermore, the unitary approximation yields the same result as in ref. [10] using a different approach. These ideas have already been used to construct the event generator UNIBAB [12,13] which simulates Bhabha events at LEP energies, including multiphoton radiation.

This paper describes the implementation of the event generator KRONOS, which simulates ep scattering at HERA energies. It is based on the leading logarithmic approximation and is capable to treat multiphoton emission in ep processes for HERA.

In section 2 we describe the features of the first release KRONOS 1.0. In subsection 2.1 the algorithm for the generation of the multiphoton final states is described. Subsection 2.2 elaborates on the implementation of the hard Born cross-sections and in subsection 2.3 the important question of specifying kinematical cuts is discussed.

The structure of KRONOS is presented in detail in section 3. Technical details regarding the parameters and the interface to other packages are discussed in sections 4 and 5, respectively.

2. Features of KRONOS 1.0

The event generator KRONOS simulates ep scattering at HERA energies including leading logarithmic QED corrections to all orders in α. It generates exclusive multiphoton events using a cascade algorithm (cf. section 2.1) which implements the renormalization group summation of leading logarithms.

The electromagnetic radiation from the incoming quark can be treated in the same spirit. However, this contribution is significantly smaller, since the leading collinear QED singularity has to be cut off at the QCD factorization scale. Furthermore it can be separated in a gauge invariant way, therefore version 1.0 of KRONOS does not include it.

Final state radiation may also be treated using a similar cascade algorithm (cf. last paper of ref. [10]). However, this corresponds to a resummation of logarithms of the form

$$\frac{\alpha}{\pi} \ln\left(\frac{\Delta E}{\sqrt{Q^2}}\right) \quad \text{or} \quad \frac{\alpha}{\pi} \ln \delta, \tag{8}$$

where ΔE and δ are the energy and angular resolutions of the detectors; for the present application these logarithms are considerably smaller than the ones from the initial state radiation.

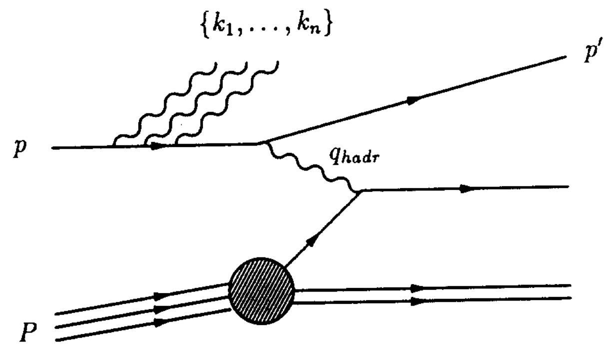

Fig. 1. Deep inelastic scattering kinematics.

Thus, in the version 1.0 of KRONOS only the initial state radiation off the incoming electron is implemented by the algorithm described in section 2.1. It generates the momenta of the radiated photons and the resulting electron momentum after multiple radiation. This electron momentum is then used as input for the hard scattering process.

The kinematical variables for the hard process are shown in fig. 1. The leptonic momentum transfer is $Q^2_{\text{lept}} = -q^2_{\text{lept}}$, with $q_{\text{lept.}} = p - p'$. Therefore, the leptonic x is given by $x_{\text{lept}} = Q^2_{\text{lept}}/2Pq_{\text{lept}}$. The hadronic momentum transfer differs from the leptonic by the photon momenta: $q_{\text{hadr}} = q_{\text{lept}} - \sum_i k_i$.

In version 1.0 of KRONOS the most important contributions have been split into two channels which may be switched on and off separately. The first channel is the neutral current reaction

$$\mathrm{e} + \mathrm{p} \to \mathrm{e}' + \mathrm{X}, \tag{9}$$

where X denotes some hadronic final state.

The other channel implements the Compton events

$$\mathrm{e} + \mathrm{p} \to \mathrm{e}' + \gamma + \mathrm{X}, \tag{10}$$

which are characterized by a high p_{T} electron and photon with little hadronic activity. This contribution is handled separately to allow a better treatment of its $1/Q^2$ singularity.

The initial state proton is described in terms of quark distributions or structure functions F_i, for which the present release contains a set of commonly used parametrizations. In the more detailed description below, we indicate which subroutines must be replaced if different parametrizations are to be used.

Version 1.0 of the program does not contain QCD parton showers and hadronization. Thus it is applicable to studies where the final state hadrons are treated inclusively (e.g. structure function measurements).

2.1. The algorithm for multiphoton radiation

In this subsection we describe the algorithm generating multiphoton events. It is similar to algorithms for quark fragmentation in QCD [14]. A more detailed discussion and a derivation using the unitary approximation may be found in ref. [12].

The Poissonian distribution of the photon numbers is generated using a standard algorithm. The mean value $\bar{n}$ of this distribution is given by the probability that no emission occurs,

$$\bar{n} = \frac{\alpha}{2\pi} \ln\left(\frac{Q^2}{m^2}\right) \int_0^{1-\epsilon} \mathrm{d}z\, P(z), \tag{11}$$

where the parameter ϵ defines the minimal energy fraction of a photon generated by the algorithm. This parameter should be chosen well below the experimental resolution of the detectors. Thus the program also generates many soft photons explicitly, which will be eliminated later by the experimental cuts. In the region of $1-\epsilon < x < 1$ a x-distribution according to

$$P_{\text{soft}}(x) \sim \exp\left(-\left(1-\frac{\alpha}{\pi}\ln\left(\frac{Q^2}{m^2}\right)\right)\ln(1-x)\right) \tag{12}$$

is chosen. Consequently, the inclusive x-distribution will not depend on the infrared cutoff parameter ϵ.

Once a photon is generated, its energy k_0 is selected according to the splitting function $P(z)$ given in eq. (5). The energy of the first photon generated is

$$k_0^{(1)} = z^{(1)}|\boldsymbol{p}| = z^{(1)}p_0,$$

where $\boldsymbol{p}$ is the momentum of the incoming electron.

The transverse component of the photon momentum is determined by a probability distribution according to the pole in the electron propagator

$$P \propto \frac{1}{2p\cdot k} = \frac{1}{-t}. \tag{13}$$

Since

$$p\cdot k = p_0 k_0(1-\cos\theta^{(1)}) = p_0{}^2 z^{(1)}(1-\cos\theta^{(1)}),$$

the polar angle of the photon is chosen accordingly to

$$P(\cos\theta^{(1)}) \propto \frac{1}{1-\cos\theta^{(1)}}, \quad -1 \le \cos\theta^{(1)} \le 1-\frac{2m^2}{Q^2}. \tag{14}$$

The lower limit on $\theta^{(1)}$ correctly reproduces the mass dependence of the leading logarithm $\ln(Q^2/m^2)$. The azimuthal angle is generated independently using a uniform distribution.

After emitting a photon, the electron momentum is determined by three momentum conservation and the energy from the mass shell condition.

The leading logarithms are obtained from the phase space region of ordered virtualities of the electron:

$$(-t^{(1)}) < (-t^{(2)}) < \cdots < (-t^{(n)}). \tag{15}$$

The photon angles $\theta^{(i)}$ are measured relative to the direction of the incoming electron, their energy fractions $z^{(i)}$ relative to the electron energy after emission of $i-1$ photons. For the first two photons, the ordering condition (15) translates into

$$z^{(1)}(1-\cos\theta^{(1)}) < \left[1+(z^{(1)})^2 - 2z^{(1)}\cos\theta^{(1)}\right] z^{(2)}(1-\cos\theta^{(2)}). \tag{16}$$

For both photons being hard ($z^{(1)}$, $z^{(2)} \approx 1$) and both being emitted under large and similar angles, this equation is not a unique ordering prescription. In this case the collinear approximation is not valid anyway and we shall replace the conditions (16) by

$$z^{(1)}(1-\cos\theta^{(1)}) < z^{(2)}(1-\cos\theta^{(2)}). \tag{17}$$

The variables for the second photon are generated in the same way as the ones for the first. Then the variables $z^{(i)}$, $\cos\theta^{(i)}$, $i=1, 2$, are ordered so that condition (17) holds.

In the case of n photons the same prescription is iterated; this means that the variables $z^{(i)}$, $\cos\theta^{(i)}$ are generated independently and ordered subsequently. For n photons, the modified ordering condition (17) implies

$$\frac{1}{p\cdot k^{(1)}} > \frac{1}{p\cdot k^{(2)}} > \cdots > \frac{1}{p\cdot k^{(n)}}, \tag{18}$$

where p is the momentum of the incoming electron.

The final electron momentum is determined by the conservation of the spatial momentum, while the energy is then fixed by putting the electron four momentum on the mass shell.

The electron momentum determined this way is then used as the input for one of the channels of hard scattering described in section 2.2.

2.2. *Implementation of the Born cross-sections*

In the following subsections we shall describe some technical details of our implementation of the hard scattering processes.

2.2.1. *Neutral current scattering*

The basic variables we use for the implementation of the Born cross-section are the scaling variable $x = -q^2/2Pq$ and the hadronic momentum transfer $Q^2 = -q^2$. Assuming a parton model for the proton, the doubly differential cross-section for scattering of unpolarized electrons reads:

$$\frac{d^2\sigma}{dx\, dQ^2} = \frac{\pi\alpha^2}{xQ^4}\sum_q \left[xq(x, Q^2) + x\bar{q}(x, Q^2)\right]\left[A_q + (1-y)^2 B_q\right]. \tag{19}$$

Here $q(x, Q^2)$ denotes the distribution of quarks of flavor q inside the proton, $y = Q^2/xs$, and

$$\begin{aligned} A_q &= \left(-e_q + g_{L,q}g_{L,e}\frac{Q^2}{Q^2+M_Z^2}\right)^2 + \left(-e_q + g_{R,q}g_{R,e}\frac{Q^2}{Q^2+M_Z^2}\right)^2, \\ B_q &= \left(-e_q + g_{L,q}g_{R,e}\frac{Q^2}{Q^2+M_Z^2}\right)^2 + \left(-e_q + g_{R,q}g_{L,e}\frac{Q^2}{Q^2+M_Z^2}\right)^2, \end{aligned} \tag{20}$$

where e_q is the charge of flavor q, and g_L and g_R are the left- and right-handed couplings to the Z^0:

$$g_{L,q} = \frac{T_{3,q} - e_q \sin^2\theta_W}{\sin\theta_W \cos\theta_W}, \qquad g_{R,q} = -\frac{e_q \sin^2\theta_W}{\sin\theta_W \cos\theta_W}, \tag{21}$$

$$g_{L,e} = \frac{-\frac{1}{2} + \sin^2\theta_W}{\sin\theta_W \cos\theta_W}, \qquad g_{R,e} = \frac{\sin^2\theta_W}{\sin\theta_W \cos\theta_W}. \tag{22}$$

The cross-sections (19) are strongly varying functions of x and Q^2. For reasons of efficiency and flexibility, we apply a combination of mapping and numerical sampling techniques.

At low x, the x-dependence of the cross-sections is governed by the distributions of sea quarks, which is roughly $\sim x^{-1}$. The Q^2-dependence is dominated by the photon exchange. Therefore we use the following mapping:

$$\xi = \log\frac{1}{x}, \qquad \eta = \frac{1}{Q^2}. \tag{23}$$

The doubly differential cross-sections $d^2\sigma/d\xi\, d\eta$ are sufficiently smooth for a numerical sampling.

Numerical sampling is implemented in the following way. During initialization of a subgenerator, the phase space for the Born process, expressed in terms of the variables ξ and η, is divided into small rectangles. In each rectangle an upper bound on the differential cross-section $d^2\sigma/d\xi\, d\eta$ is estimated, and the rectangle is assigned a weight which is just its area times the estimate of the function. In the generation phase, first a rectangle is chosen with a probability according to its weight, and the variables ξ and η are generated uniformly in this rectangle. Then a rejection weight is calculated, which is just the ratio of the true differential cross-section to its estimate from the initialization step. The estimate is sufficiently conservative to avoid events with a rejection weight exceeding unity. Therefore no special precautions have to be taken to handle such events.

This "raw event" will be accepted according to the rejection weight, only if it lies within the specified cuts, otherwise its weight equals zero, and it will be discarded. After all events have been generated, the integrated cross-section is calculated from the accumulated weights by the formula

$$\sigma = \frac{\sigma_{\text{est}}}{N_{\text{total}}} \sum_{\text{accepted}} \left(\frac{d^2\sigma}{d\xi\, d\eta}(\xi_i, \eta_i) \Big/ \frac{d^2\sigma_{\text{est}}}{d\xi\, d\eta} \right), \tag{24}$$

where σ_{est} is the integral over the sampling function, N_{total} is the count of all raw events, and the sum runs only over those events that lie within the specified phase space boundary.

For semi-inclusive measurements in the region of very low hadronic momentum transfers $Q^2 \lesssim \mathscr{O}(5\ \text{GeV}^2)$, the use of the electroproduction structure functions is more appropriate, since the parton model is not applicable for such small momentum transfers. Therefore we have implemented the possibility to use parametrizations of the neutral current structure functions as well. The cross-section in this case reads:

$$\frac{d^2\sigma^{\text{NC}}}{dx\, dQ^2} = \frac{4\pi\alpha^2}{xQ^4}\left[xy^2 F_1^{\text{NC}}(x, Q^2) + (1-y) F_2^{\text{NC}}(x, Q^2) + xy\left(1 - \frac{y}{2}\right) F_3^{\text{NC}}(x, Q^2) \right]. \tag{25}$$

The distribution of KRONOS version 1.0 contains the parametrizations by Bodek et al. [15] and alternatively the parametrization by Brasse and Stein et al. [16]. These parametrizations can be used for $Q^2 < 6\ \text{GeV}^2$, together with the parton model predictions for higher momentum transfer.

2.2.2. Compton scattering

There is a particular class of scattering processes, where the transverse momenta of the outgoing electron and one hard photon are nearly balanced, all other particles having small transverse momenta. In this case the hadronic momentum transfer and the hadronic mass are small. This class of events has been called Compton events [17]. The lowest non-vanishing contribution to the Compton scattering process is contained in the logarithmically non-leading $\mathscr{O}(\alpha)$ radiative corrections. However, for studies of radiative corrections at small x and high y, its inclusion is indispensable. Therefore we have implemented it as a separate channel. The kinematics of the Compton process is shown in fig. 2.

To lowest order, the differential cross-section for Compton scattering reads:

$$\frac{d^2\sigma_c}{dx\, dy} = \frac{2\pi\alpha^2}{s} \frac{1 + (1-y)^2}{x(1-y)} \gamma(x, Q^2). \tag{26}$$

Here $\gamma(x, Q^2)$ is the distribution of photons inside the proton, and $Q^2 = x(1-y)s$ is the maximal virtuality of the photon exchanged between the electron and the proton.

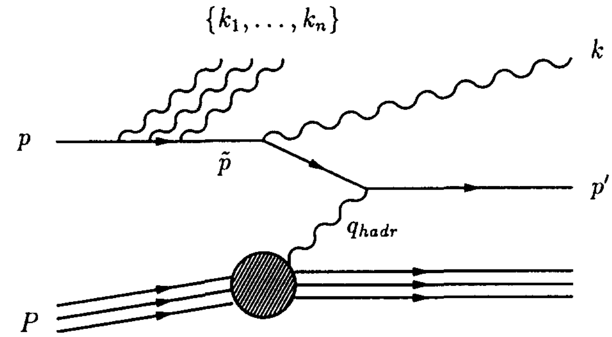

Fig. 2. Compton process kinematics.

If electromagnetic structure functions F_1^{NC} and F_2^{NC} are given explicitly, we can factorize the cross-section for the Compton contribution and define a photon distribution function for use in eq. (26).

$$\gamma(x, Q^2) = \frac{\alpha}{2\pi}\int_{\Lambda^2}^{Q^2}\frac{\mathrm{d}Q_{\mathrm{hadr}}^2}{Q_{\mathrm{hadr}}^2}\int_x^1\frac{\mathrm{d}z}{xz^3}\left\{\left[z^2+(x-z)^2-2x^2z^2\frac{M^2}{Q_{\mathrm{hadr}}^2}\right]F_2(z, Q_{\mathrm{hadr}}^2)\right.$$
$$\left. +x^2\left[2zF_1(z, Q_{\mathrm{hadr}}^2)-F_2(z, Q_{\mathrm{hadr}}^2)\right]\right\}. \tag{27}$$

If parton distributions are used, the situation is different. As there are no parametrizations of the photon distribution in the proton available, we use a Weizsäcker–Williams approximation to relate it to the quark distributions:

$$\gamma(x, Q^2) = \frac{\alpha}{2\pi}\int_{\Lambda^2}^{Q^2}\frac{\mathrm{d}Q_{\mathrm{hadr}}^2}{Q_{\mathrm{hadr}}^2}\int_x^1\frac{\mathrm{d}z}{z}\,\frac{1+(1-x/z)^2}{x/z}\sum_q e_q^2 q(z, Q_{\mathrm{hadr}}^2). \tag{28}$$

Here e_q denotes the charge of the quark q, and Q_{hadr}^2 is the hadronic momentum transfer. Λ^2 is a cutoff which regularizes the singularity arising from collinear emission of the photon from the quark line. However, since $q(x, Q_{\mathrm{hadr.}}^2)$ must vanish linearily with $Q_{\mathrm{hadr.}}^2$ [18,19], this quark mass singularity is fictitious. In the Monte Carlo implementation, Λ^2 plays a role as a regularization parameter and thus must be kept small but finite.

Strictly speaking, eq. (28) uses the quark distributions for values of Q_{hadr}^2 for which they are no longer applicable. Thus we follow refs. [18,19] and use a phenomenological correction factor $(1-\exp(3.37\, Q_{\mathrm{hadr}}^2/\mathrm{GeV}^2))$ for the quark distributions.

2.3. Internal cuts

As already mentioned, the basic internal variables of the subgenerators we use are the hadronic scaling variable x and the hard momentum transfer Q^2. In the initialization step, the subgenerators derive internal cuts on these variables from the given physical cuts on the Bjorken variables x and y and on the minimal hadronic momentum transfer Q_{hadr}^2, as described below.

Radiative corrections from collinear photon emission off the electron line tend to enhance the cross-section at low x and high y. This effect may be easily understood. Emission of photons shifts the momentum transfer at the hadronic vertex to smaller values and thus enhances the cross-section due to the $1/Q_{\mathrm{hadr}}^2$ behavior of the photon propagator.

In the standard deep inelastic scattering channel the smallest hadronic momentum transfer for given x and y is

$$Q^2_{\mathrm{hadr,min}}(x, y) = \frac{1-y}{1-xy} xys. \tag{29}$$

From this relation it is clear that one has to specify a lower cut on x and both a lower and an upper cut on y, in order to keep the photon propagator bounded.

The choice of these cuts influences the CPU time needed to generate an event within the specified region of phase space. If one allows for y-values very close to 1, i.e. for events with a large amount of energy being radiated into the beam pipe, a large difference between the electronic and the hadronic momentum transfer occurs. In this case the implementation of the Born cross-section used in KRONOS version 1.0 may become less efficient.

For sufficient hadronic activity, one can directly calculate the hadronic momentum transfer from the measured transverse hadronic momenta by the method of Jaquet and Blondel [20]. In the Monte Carlo event generator KRONOS we have therefore implemented the possibility to specify a cut on the smallest hadronic momentum transfer. In the region of high y this may be used to restrict the available phase space and improve the performance of the generator.

In the Compton channel, the smallest characteristic momentum transfer in the electron–photon subsystem is

$$Q^2_{\mathrm{lept,min}}(x, y) = x(1-y)s. \tag{30}$$

From eq. (26) it is obvious that the Compton cross-section gets large if one allows for y values close to 1. Thus specifying a lower cut on x and an upper cut on y will be sufficient. On the other hand, we use a regularization parameter Λ to cut off the integration over the hadronic momentum transfer. We have decided to treat this parameter as unrelated to the (optional) cut on the hadronic momentum transfer which may be given for the neutral current cross-section.

3. Structure of the program

KRONOS has been designed to be as modular as possible, with information flowing only through well-defined channels (cf. fig. 3).

A listing of all external symbols in KRONOS is given in the appendix.

3.1. The implementation of radiative corrections

The flow of control for event generation in KRONOS is depicted in fig. 4.

As a first step, the routine `krbini` performs the usual initial state branching, as described in section 2.1. However, the generated momentum fractions are stored in a common block (`/krcmsc/`) and only the effective center of mass energy $zs = (\prod_i z_i)s$ is passed to the hard subprocess generator `krgen`.

After `krgen` has generated the invariants (x, y), these are combined with the previously saved $\{(z_i, t_i)\}$ for construction of particle four momenta in `krecon`.

3.2. Parameter management

Controlling physics and Monte Carlo parameters in a FORTRAN-77 Monte Carlo library from within the application program can be a difficult task. Allowing direct manipulation of common block data by

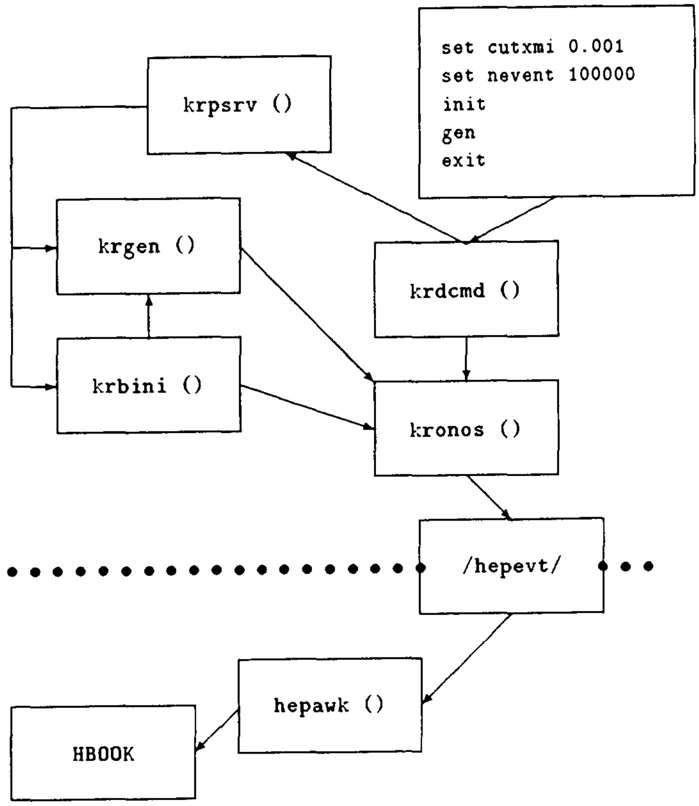

Fig. 3. Top level structure of KRONOS. The arrows connecting the building blocks illustrate the flow of data in the program. All requests from the application program should be passed to KRONOS via the krdcmd subroutine. The generated events are returned to the application program in the standard /hepevt/ common block [25]. (In the example the events are analyzed on the parton level by hepawk [23] and HBOOK [24].)

the application program requires that this program "knows" whether a particular change to a parameter should trigger a reinitialization of the Monte Carlo program.

To overcome this problem we have implemented a simple client–server model for parameter

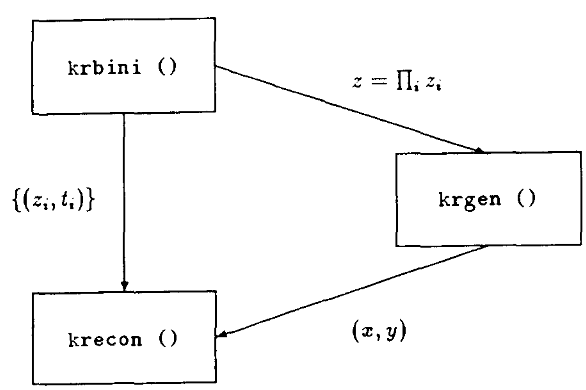

Fig. 4. Implementation of radiative corrections. The initial state branching is performed by the subroutine krbini. The total energy loss z of the electron is passed to the hard subgenerators (krgen). The output of krgen is combined with the photon energies and their transverse momenta by krecon to form the four vectors of the final state particles.

handling. In this approach the application program (client) sends requests to the Monte Carlo, or a module within the Monte Carlo (server), which keeps complete control over its parameters. In this model the application program will not need to reference the common blocks at all.

The server is structured in two layers. The outer layer is handled by the command interpreter `krdcmd`. `krdcmd` honors requests in form of FORTRAN-77 character strings, see section 3.3 for details. The inner layer is implemented by the subroutine `krpsrv`.

Each parameter may have three attributes: `'r'` (readable), `'w'` (writable), and `'x'` (force reinitialization). All parameters will be readable, of course, the `'r'` attribute was added for symmetry.

These attributes are best explained by the following typical examples:

1. `nevent: 'rw'`
`nevent`, the number of events to be generated by the next ``gen'' command, can be changed by the application program and does not imply a reinitialization of the Monte Carlo.

2. `mass1z: 'rwx'`
`mass1z`, the Z^0 mass (M_Z), can be changed by the application program, but requires a reinitialization of the Monte Carlo before the next event.

3. `mass2z: 'r'`
`mass2z`, the squared Z^0 mass (M_Z^2), should not be changed by the application program, as it is considered a derived parameter (derived from M_Z).

Of course, changing a parameter with the `'x'` attribute does not cause an immediate reinitialization, but a flag is set which will force the reinitialization before the next event is generated.

If the application program tries to modify a read-only parameter, an error message is printed and the parameter is left unchanged.

3.3. *The command interpreter* `krdcmd`

All communication of the user program with KRONOS should use the subroutine `krdcmd`, which takes a FORTRAN-77-string as argument and interprets it as a command: a simple example is shown in fig. 5. This approach is clearly preferable to changing internal parameters in KRONOS "by hand", since it allows clean control over necessary reinitializations.

It is of course possible to read these command strings from a file, in the manner of the sample driver `krdriv`.

`krdcmd` internally converts all upper case letters to lower case so the commands can be entered in either case.

The simple commands understood by `krdcmd` are (here keywords are typeset in typewriter font and variables in italics; vertical bars denote alternatives):

```
* kronos.examples
      ...
      call krdcmd ( ``set cutxmi 0.001 '')          set the x cuts
      call krdcmd ( ``set cutxma 0.1 '')
      call krdcmd ( ``print all '')                 print the value of all parameters
      ...
```

Fig. 5. `krdcmd`.

`init`
Force initialization of `KRONOS` and write an initialization record into `/hepevt/`, which should trigger the necessary initializations in the analyzer.

`gen`
Generate `nevent` events and call `hepawk` to analyze them.

`close`
Write a deinitialization record to `/hepevt/`, which should trigger the necessary cleanups in the analyzer.

`quit|q`
Terminate `KRONOS` without writing a deinitialization record.

`exit|e`
Write a deinitialization record and terminate `KRONOS`.

`structure` *parametrization set*
Switch to another parametrization of parton distributions. The following are available in the PAKPDF package [21] (see ref. [21] for references and a detailed description):
1. `do` (sets 1, 2, 3) – Duke/Owens;
2. `ehlq` (sets, 1, 2) – Eichten/Hinchliffe/Lane/Quigg;
3. `dflm` (sets, 1, 2) – Diemoz/Ferroni/Longo/Martinelli;
4. `grv` (sets 1, 2) – Glück/Reya/Vogt;
5. `mt` (sets −5, −4,...,5) – Morfin/Tung;
6. `hmrs` (sets 1, 2) – Harriman/Martin/Roberts/Stirling;
7. `kmrs` (sets 1, 2, 3, 4) – Kwiecinski/Martin/Roberts/Stirling;
8. `mrs` (sets 1, 2, 3, 4) – Martin/Roberts/Stirling.

(The first set of KMRS has been chosen as default).

`structure` *parametrization*
This alternate form of the structure command is used to switch from parton distributions to the structure function parametrizations for low Q^2. For large Q^2 the current parton distribution stays in effect.
1. `brasse` – Use the parametrizations of Brasse and Stein et al. [16].
2. `bodek` – Use the parametrization of Bodek et al. [15].

`channel` *channel* `on|off`
Switch hard subchannels on or off, currently available are `nc` and `compton`, where `nc` is the neutral current deep inelastic scattering channel and `compton` is the Compton channel. The latter is switched off by default.

Table 1
Changeable physical parameters

Variable name	Physical significance	Unit	Default value
`ahpla`	$1/\alpha_{QED}$		137.0359895
`lambd`	Λ_{QCD}	GeV	0.2
`mass1e`	$m_{e^\pm}$	GeV	$0.51099906 \times 10^{-3}$
`mass1p`	m_p	GeV	0.93827231
`mass1z`	M_{Z^0}	GeV	91.17
`mass1w`	$M_{W^\pm}$	GeV	80.0
`sin2w`	$\sin^2\theta_W$		0.23

`set` *variable ival* | *rval*
Set physical or internal parameters. See the tables 1 and 3 for a comprehensive listing of all variables. For example, the command `set ahpla 128.0` will set the QED fine structure constant to 1/128.

`print` *variable* |`all`|`channels` |`pdf`
Print the value of physical or internal variables. Specifying the special variable `all` causes a listing of all variables known to KRONOS. The other special variables `channels` and `pdf` can be used to query information about the currently active channels and the currently selected parametrization of the parton distribution respectively.

`debug`|`nodebug` *flag*
Switch debugging flags.

`testran`
Test the portability of the random number generator. We use the generator RANMAR [22] which should give identical results on almost all machines.

`banner`
Print a string identifying this version of KRONOS.

`echo` *message*
Print *message* on standard output.

The tokens are separated by blanks. Blank lines and lines starting with a # are ignored and may be used for comments. For portability, only the first 72 characters of each line are considered.

4. Physical and technical parameters

Table 1 shows the physical input parameters which can be changed by the application program. Changing these will trigger a reinitialization of KRONOS after updating the derived parameters displayed in table 2.

In version 1.0 of KRONOS the algorithms for updating the derived parameters are obviously trivial, because renormalizations by the weak interaction are ignored. After the inclusion of a library for weak loop corrections, these algorithms will become more involved.

The parameters in table 3 are controlling the event generation.

Table 2
Derived physical parameters

Variable name	Physical significance	Unit	Algorithm
alpha	α_{QED}		1/ahpla
lambd2	Λ^2_{QCD}	GeV^2	lambd**2
mass2e	$m^2_{e^\pm}$	GeV^2	mass1e**2
mass2p	m^2_p	GeV^2	mass1p**2
mass2z	$M^2_{Z^0}$	GeV^2	mass1z**2
mass2w	$M^2_{W^\pm}$	GeV^2	mass1w**2
sin1w	$\sin\theta_W$		sqrt (sin2w)
cos2w	$\cos^2\theta_W$		1.0-sin2w
cos1w	$\cos\theta_W$		sqrt (cos2w)

Table 3
Monte Carlo parameters

Variable	Semantics	Default value
elecen	e^- energy	30 GeV
proten	p energy	820 GeV
nevent	number of events	1000
epsiln	internal infrared cutoff	10^{-4}
cutxmi	lower cut in x	0.001
cutxma	upper cut in x	0.5
cutymi	lower cut in y	0.1
cutyma	upper cut in y	0.9
cutqmi	lower cut on hadronic Q^2	100
cutqma	upper cut on hadronic Q^2	s
rseed	random number seed	54217137
errmax	maximum error count	100
verbos	verbosity	0
runid	run identification	
stdin	standard input	5
stdout	standard output	6
stderr	standard error	6

5. FORTRAN-77 interface

Figure 6 shows a sample application of KRONOS. In this example we generate 100000 events. The call to krdcmd with the argument 'gen ' will execute the loop in fig. 7, where each event in the common block /hepevt/ will be analyzed by the user routine hepawk (e.g. ref. [23]).

```
* kronosappl.f
      ...
      call krdcmd ( 'init ')                          initialize the generator
      ...
      call krdcmd ( 'set nevent 100000 ')
      call krdcmd ( 'gen ')                           generate 100000 events
      ...
      call krdcmd ( 'close ')                         cleanup
      ...
```

Fig. 6. FORTRAN-77 interface.

```
* krdcmd.f
      subroutine krdcmd (cdmlin)
      character*(*) cdmlin
      ...
      else if (cmdlin.eq. 'gen ')
          do 10 n=1, nevent
              call kronos (1)                         generate an event
              call hepawk ( 'scan ')                  analyze the event
10        continue
      else
      ...
      end
```

Fig. 7. Event generation loop.

6. Conclusions

In this paper we have described the Monte Carlo event generator KRONOS for QED radiative corrections at HERA. It focusses on higher orders (i.e. multi-photon radiation) and uses the leading logarithmic approximation.

KRONOS has been extensively tested on two very different machines. Thus we have experiences either for conventional mainframes such as IBM 3090 and IBM 9000 and for more modern UNIX workstations such as an IBM RS/6000 (Model 520). Typically KRONOS needs about one millisecond per event on either machine. This number gives a rough estimate, since timings depend on the region of phase space in which events are generated.

Since the FORTRAN-77 source conforms to ANSI X3.9-1978, it should run without modifications on all platforms.

Acknowledgements

The authors are indebted to H. Spiesberger for valuable discussions.

Appendix

External symbols: Common blocks and subroutines

To avoid possible name clashes with other packages, all external symbols exported by KRONOS begin with the two letters KR.

Common blocks

The following common blocks are used by KRONOS:

/hepevt/, /hepspn/: standard common blocks for passing generated events [25].
/krpcom/: overall common block, providing parameters, etc.
/krcgen/: common block used by the hard subgenerators.
/krcsig/: common block for the lowest order cross-sections.
/krcmsc/: miscellaneous parameters.
/krcqua/: parton distibutions.
/krcabc/, /krcrdc/, /krcrdc/, /krcrdf/, /krcsin/: structure function parametrizations.

Driver program:

krdriv: a sample main program, which reads commands from standard input and feeds them into krdcmd.
kronos: the low level entry point into the generator for application programs.
krdcmd: KRONOS' command interpreter, the preferred entry point for application programs.
krdlxi, krdlxd, krdlxs: utility routines: tokenization of input.

Parameter management

These routines are used to control the parameters common block /krpcom/:

krpsrv: server handling parameter changing requests.
krpini: block data supplying default values.
krpprn: print parameters.

Branching:
krbini: generates the initial state photon radiation.

Event construction and kinematics:
krecon: construct four momenta from the generated invariants.

Hard subevent generation:
krgen: main entry point for hard subevents.
krgcut: calculate internal cuts from physical cuts.
krgini: initializations.
krgraw: generation of raw (weighted) events.
krgwgt: calculate rejection weight.
krgsig: differential cross-sections.
krgbd: block data initializations.

Parton cross-sections:
krxco, krxcos: QED Compton scattering
krxnc, krxncs: neutral current scattering
krxcpl: standard model coupling constants.
krxini: initialization.

Structure functions:
krpdis: calculate parton distributions.
krstrf: calculate structure functions (presently they are derived from parton distributions for large momentum transfers).
krsbs, krslac: parametrizations of structure functions at low Q^2.
krfbck, krfbra, krff2, krfres, krfrf, krfste, krfsvt, krfw12: functions used by the above parametrizations.

Importance sampling:
krsami: initialization, perform sampling for a subgenerator.
krsamg: generate raw event according to sampled cross-sections.
krgues: optimization of sampling domain and a priori weights.

Random numbers:
krrgen: returns a double precision uniform deviate.
krrtst: test the portability of the random number generator.

Utilities:
krumsg: messages and error exit.
krulwr: convert input to lower case.

References

[1] R. Peccei, ed., Proc. HERA Workshop, DESY, Hamburg (1988).
[2] Proc. Workshop on Physics at HERA, DESY, 1991, to appear.
[3] F. Eisele, in: Small-x Behaviour of Deep Inelastic Structure Functions in QCD, eds. A. Ali and J. Bartels (North-Holland, Amsterdam, 1990) p. 1.

[4] M. Böhm and H. Spiesberger, Nucl. Phys. B 294 (1987) 1081.
[5] D.Y. Bardin, Č. Burdik, P.C. Christova and T. Riemann, Z. Phys. C 42 (1989) 679.
[6] J. Kripfganz, H.-J. Möhring and H. Spiesberger, Z. Phys. C 49 (1991) 501.
[7] A. Kwiatkowski, H. Spiesberger and H.-J. Möhring, Comput. Phys. Commun. 69 (1992) 155.
[8] V. Gribov and L. Lipatov, Sov. J. Nucl. Phys. 15 (1972) 438.
L. Lipatov, Sov. J. Nucl. Phys. 20 (1974) 94.
G. Altarelli and G. Parisi, Nucl. Phys. B 126 (1977) 298.
[9] E. Kuraev and V. Fadin, Sov. J. Nucl. Phys. 43 (1985) 466.
[10] G. Altarelli and G. Martinelli, in: Physics at LEP, eds. J. Ellis and R. Peccei, CERN 86-02, Geneva (1986).
J.P. Alexander et al., Phys. Rev. D 37 (1988) 56.
O. Nicrosini and L. Trentadue, Nucl. Phys. B 318 (1989) 1.
G. Bonvicini and L. Trentadue, Nucl. Phys. B 323 (1989) 253.
W. Beenakker, F.A. Berends and W.L. van Neerven, in: Radiative Corrections for e^+e^- Collisions, ed. J.H. Kühn (Springer, Berlin, 1989) p. 3.
[11] H.D. Dahmen, T. Mannel and P. Manakos, Z. Phys. C 40 (1988) 425, 435; Phys. Rev. D 38 (1988) 1176.
H.D. Dahmen, P. Manakos, T. Mannel and T. Ohl, Z. Phys. C 44 (1989) 139.
[12] H.D. Dahmen, P. Manakos, T. Mannel and T. Ohl, Z. Phys. C 50 (1991) 75.
[13] H.D. Dahmen, P. Manakos, T. Mannel and T. Ohl, UNIBAB, the Darmstadt–Siegen Monte Carlo for Bhabha scattering in unitary approximation, IKDA 89/41, SI 89-9 (1989).
T. Ohl, Unitäre Näherungen zur Bhabha-Streuung bei hohen Energien und ihre Implementation in Monte Carlo Eventgeneratoren, Doctoral Thesis, Technische Hochschule Darmstadt (1990).
[14] T. Sjöstrand, Comput. Phys. Commun. 39 (1986) 347.
[15] A. Bodek et al., Phys. Rev. D 20 (1979) 1471.
[16] F.W. Brasse et al., Nucl. Phys. B 110 (1976) 413.
S. Stein et al., Phys. Rev. D 12 (1975) 1884.
[17] J. Blümlein, Z. Phys. C 47 (1990) 89.
[18] N.Yu. Volkonsky and L.V. Prokhorov, JETP Lett. 21 (1975) 389.
[19] A. Akhundov, D. Bardin and L. Kalinovskaja, Electron-state Bremsstrahlung Processes ep $\to \gamma$ep(x) at HERA, DESY 90-130 (1990).
[20] J. Feltesse, in: Proc. HERA Workshop, ed. R. Reccei, DESY, Hamburg (1988).
[21] K. Charchuła, Comput. Phys. Commun. 69 (1992) 360.
[22] G. Marsaglia, A. Zaman and W.-W. Tsang, Stat. Prob. Lett. 9 (1990) 35.
[23] T. Ohl, Comput. Phys. Commun. 70 (1992) 120, this issue.
[24] R. Brun and D. Lienhart, HBOOK Version 4, User Guide, CERN, Geneva (1987).
[25] G. Altarelli, R. Kleiss and C. Verzegnassi, eds., Z Physics at LEP 1, CERN 89-08, Geneva (1989).

TEST RUN

KRONOS version 1.0 is distributed together with a sample command file and hepawk script, which are described below. To run this example, the user will need to link KRONOS with the CERN library, because histogramming is done by HBOOK [24].

The file sample.kronos is read from standard input (unit stdin, which is initialized to 5) and sample.hepawk is read from the file SCRIPT (i.e. under MVS from the file which has been allocated to the DDNAME SCRIPT and under UNIX from the file script or from the value of the environment variable SCRIPT).

Sample KRONOS input

This is a sample JCL for KRONOS version 1.0 at the DESY computing center. The generated output will be shown below.

```
//* 16/09/91 109162204  MEMBER NAME  #SAMPLE  (TEX)      M  JCL
//        JOB CLASS=A,TIME=(000,24),MSGCLASS=T,NOTIFY=T000HL
//*
//**********************************************************************
//*
//*   KRONOS JCL PROCEDURE
//*   ===================
//*
//KRONOS PROC LMOD='MAIN',                  KRONOS EXECUTABLE
//            LLIB='T000HL.KRONOS.LOAD',    LOAD LIBRARY (USE YOUR OWN!)
//            STDOUT='SYSOUT=*',            OUTPUT DATASETS
//            STDERR='SYSOUT=*',
//            PAWDSN='NULLFILE',            HBOOK (ZEBRA) DATASETS
//            PAWDSP='(NEW,CATLG,DELETE)',
//            PAWUNT='DISK',
//            PAWSPC='(TRK,(1,1))'
//*
//G       EXEC PGM=&LMOD
//STEPLIB  DD DISP=SHR,DSN=&LLIB
//*
//FT00F001 DD &STDOUT
//FT06F001 DD &STDERR
//FT05F001 DD DDNAME=KRONOS
//SCRIPT   DD DDNAME=HEPAWK
//PAW      DD DSN=&PAWDSN,
//            DISP=&PAWDSP,
//            UNIT=&PAWUNT,
//            SPACE=&PAWSPC,
//            DCB=(RECFM=F,LRECL=4096,BLKSIZE=4096)
//*
//       PEND
//*
//**********************************************************************
//*
//*
//**********************************************************************
//*
//*   SAMPLE KRONOS JOB
//*   ================
//*
//SAMPLE EXEC KRONOS
//*
//*
```

```
//**********************************************************************
//*
//*   SAMPLE KRONOS INPUT
//*   ===================
//*
//KRONOS   DD *
# sample input for KRONOS v1.0

# set up the cuts
set cutqmi 2.0E+2
set cutymi 2.0E-3
set cutyma 0.97
set cutxmi 3.0E-3

# initialize the generator
init

# generate some events
set nevent 50000
gen

# we're done.
close
quit

//**********************************************************************
//*
//*   SAMPLE HEPAWK ANALYZER
//*   ======================
//*
//HEPAWK   DD *
# sample HEPAWK analyzer for KRONOS v1.0

BEGIN
  {
    printf ("\nWelcome to the KRONOS test:\n");
    printf ("***************************\n\n");

    printf ("Monte Carlo Version: %s\n", REV);
    printf ("                Run: %d\n", RUN);
    printf ("               Date: %s\n\n", DATE);

    theta_min = 50;          # this looks like the Zeus BCAL:
    theta_max = 145;
    E_min = 10;
    E_max = 50;

    h_x = book1 (0, "***  x dsigma/dx (x)  ***", 50, 1.0E-3, 1.0);
    h_y = book1 (0, "***  y dsigma/dy (y)  ***", 50, 1.0E-3, 1.0);

    incut = 0;               # initialize counter
    dumped_an_event = 0;
  }

# print out the first multiphoton event.
```

```
(dumped_an_event == 0)
  {
    n = 0;                   # count the photons
    for (@ph in PHOTONS)
      n++;

    if (n > 1)               # dump if many ...
      {
        printf ("\nDumping the first multiphoton event:\n");
        printf ("************************************\n\n");
        dump ();
        dumped_an_event++;
      }
  }

# Look at the outgoing electron:

  {
    for (@p in ELECTRONS)    # Collect the outgoing electron.
      $el = @p:p;
  }

# apply angle and energy cuts:

(theta_min < angle ($el, @B1:p) * DEG < theta_max)
 && ($el:E > E_min)
  {
    incut++;                 # count this event

    S = (@B1:p + @B2:p)^2;   # Center of mass energy
    $q = @B1:p - $el;        # (Electronic) momentum transfer
    Q2 = - $q^2;
    x = Q2/(2*$q*@B2:p);     # Bjorken variables
    y = Q2/(x*S);

    fill (h_x, x);           # fill the histograms
    fill (h_y, y);
  }

END
  {
    # Dump some numbers
    printf ("\nRESULTS:\n");
    printf ("********\n\n");
    printf ("       raw events: %d, raw cross section:        %g mb\n",
            NEVENT, XSECT);
    printf ("events after cuts: %d, cross section after cuts: %g mb\n",
            incut, (incut/NEVENT) * XSECT);
    printf ("\nHISTOGRAMS:\n");
    printf ("***********\n\n");

    plot ();   # plot the histograms

    printf ("\ndone.\n");
  }

//*********************************************************************
```

Sample output

This is the output of the sample job, which will appear on standard output (unit stdout). There is no output on unit stderr.

```
krdcmd: message: starting KRONOS, Version 00.99/09, (build 910916/2200)
hepawk: message: starting HEPAWK, Version  0.99/02, (build 910902/1045)

Welcome to the KRONOS test:
**************************

Monte Carlo Version: v00.99 (Sep 16 00:00:00  1991)
                Run: 1035996160
               Date: Sep 16 22:05:00  1991


Dumping the first multiphoton event:
************************************

======================================================================
Dumping the event record for event #   2
There are    7 entries in this record:
----------------------------------------------------------------------
Entry #   1 is an incoming (HERWIG convention) electron
p: (0.3000E+02; 0.0000E+00, 0.0000E+00, -.3000E+02),  m: 0.0000E+00
s: (0.1000E+01; 0.0000E+00, 0.0000E+00, 0.0000E+00)
----------------------------------------------------------------------
Entry #   2 is an incoming (HERWIG convention) proton
p: (0.8200E+03; 0.0000E+00, 0.0000E+00, 0.8200E+03),  m: 0.0000E+00
s: (0.1000E+01; 0.0000E+00, 0.0000E+00, 0.0000E+00)
----------------------------------------------------------------------
Entry #   3 is an existing photon
p: (0.2683E-01; -.2599E-05, -.2604E-05, -.2683E-01),  m: 0.0000E+00
v: (0.0000E+00; 0.0000E+00, 0.0000E+00, 0.0000E+00)
s: (0.1000E+01; 0.0000E+00, 0.0000E+00, 0.0000E+00)
The mother is the electron                 #  1.
----------------------------------------------------------------------
Entry #   4 is an existing photon
p: (0.9949E-01; -.1810E-01, 0.2036E-01, -.9569E-01),  m: 0.0000E+00
v: (0.0000E+00; 0.0000E+00, 0.0000E+00, 0.0000E+00)
s: (0.1000E+01; 0.0000E+00, 0.0000E+00, 0.0000E+00)
The mother is the electron                 #  1.
----------------------------------------------------------------------
Entry #   5 is an existing electron
p: (0.3862E+02; -.2856E+02; 0.2534E+02, -.5752E+01),  m: 0.0000E+00
v: (0.0000E+00; 0.0000E+00, 0.0000E+00, 0.0000E+00)
s: (0.1000E+01; 0.0000E+00, 0.0000E+00, 0.0000E+00)
The mother is the electron                 #  1.
----------------------------------------------------------------------
Entry #   6 is a decayed photon
p: (-.8740E+01; 0.2858E+02, -.2536E+02, -.2413E+02),  m: -.4434E+02
v: (0.0000E+00; 0.0000E+00, 0.0000E+00, 0.0000E+00)
s: (0.1000E+01; 0.0000E+00, 0.0000E+00, 0.0000E+00)
This is an initial entry.
This particle has no daughters.
----------------------------------------------------------------------
Entry #   7 is an existing up-quark
p: (0.5508E+02; 0.2858E+02, -.2536E+02, 0.3970E+02),  m: 0.0000E+00
v: (0.0000E+00; 0.0000E+00, 0.0000E+00, 0.0000E+00)
s: (0.1000E+01; 0.0000E+00, 0.0000E+00, 0.0000E+00)
The mother is the proton                   #  2.
======================================================================

RESULTS:
********

      raw events:      50000, raw cross section:        0.2235E-05 mb
events after cuts:      12718, cross section after cuts: 0.5685E-06 mb
```

```
 HISTOGRAMS:
 ***********

1***  x dsigma/dx (x)  ***

 HBOOK     ID =         1                                      DATE  16/09/91               NO =
1

      560                               3
      540                         4     X
      520                        4X823  X
      500                      82XXXXXX X
      480                     2XXXXXXXX5X
      460                     XXXXXXXXXXX67
      440                     XXXXXXXXXXXXX4
      420                    XXXXXXXXXXXXXXX6
      400                    XXXXXXXXXXXXXXXX794
      380                    XXXXXXXXXXXXXXXXXXX
      360                    XXXXXXXXXXXXXXXXXXX
      340                    XXXXXXXXXXXXXXXXXXX9 9
      320                   3XXXXXXXXXXXXXXXXXXXX9X
      300                  5XXXXXXXXXXXXXXXXXXXXXXX91
      280                  XXXXXXXXXXXXXXXXXXXXXXXXXX
      260                  XXXXXXXXXXXXXXXXXXXXXXXXXX6
      240                 5XXXXXXXXXXXXXXXXXXXXXXXXXXX0
      220                 XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
      200                1XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
      180                XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX91
      160                XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
      140                XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
      120               2XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
      100               XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX8
       80               XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX1
       60               XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
       40              8XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
       20              XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

CHANNELS  10   0        1         2         3         4         5
           1   12345678901234567890123456789012345678901234567890

CONTENTS 100            1122344445555545444433333322222111
          10           3083902698021000745521998313985276496
           1.          7521060475886570172492588899932192072

LOW-EDGE   1.                                    1111122233455678
*10**  1   0   00000000000000000111112223344567890246915838307657
           0   11111222333456679023580371617432359545081810615080
           0   01357026049520991408828956368914156859571411514659
           0   05214093278753242720509942196504805946478038183865

 * ENTRIES =      12718      * ALL CHANNELS = 0.1272E+05      * UNDERFLOW = 0.0000E+00      * OVERFLOW
= 0.0000E+00
 * BIN WID = 0.1998E-01      * MEAN VALUE   = 0.6490E-01      * R . M . S = 0.8389E-01
```

```
1***  y dsigma/dy (y)  ***

 HBOOK     ID =         2                                      DATE  16/09/91              NO =
2

     1520                                                     4
     1480                                                     X
     1440                                                     X
     1400                                                     X
     1360                                                     X
     1320                                                     X
     1280                                                     X
     1240                                                    9X
     1200                                                    XX
     1160                                                    XX
     1120                                                   4XX
     1080                                                   XXX
     1040                                                   XXX
     1000                                                   XXX
      960                                                   XXX
      920                                                  2XXX
      880                                                  XXXX
      840                                                 8XXXX
      800                                                 XXXXX
      760                                                 XXXXX
      720                                                 XXXXX
      680                                               14XXXXX
      640                                               XXXXXXX7
      600                                              2XXXXXXXX
      560                                             7XXXXXXXXX
      520                                            4XXXXXXXXXX
      480                                           2XXXXXXXXXXX
      440                                           XXXXXXXXXXXX
      400                                         25XXXXXXXXXXXX
      360                                         XXXXXXXXXXXXXX
      320                                      678XXXXXXXXXXXXXX
      280                                      XXXXXXXXXXXXXXXXX
      240                                    09XXXXXXXXXXXXXXXXX4
      200                                   4XXXXXXXXXXXXXXXXXXXX
      160                                 49XXXXXXXXXXXXXXXXXXXXX
      120                               08XXXXXXXXXXXXXXXXXXXXXXX
       80                             36XXXXXXXXXXXXXXXXXXXXXXXXX
       40                          128XXXXXXXXXXXXXXXXXXXXXXXXXXX

CHANNELS  10   0        1         2         3         4         5
             1 12345678901234567890123456789012345678901234567890

CONTENTS1000                                                111
         100                             111122333334455668802462
          10                         3568135703001785946453993921
          1.                       782261298837582111798573169697

LOW-EDGE  1.                                     1111122233455678
*10**  1     0 00000000000000000111112223344567890246915838307657
             0 11111222333456679023580371617432359545081810615080
             0 01357026049520991408828956368914156859571411514659
             0 05214093278753242720509942196504805946478038183865

* ENTRIES =       12718      * ALL CHANNELS = 0.1272E+05      * UNDERFLOW = 0.0000E+00      * OVERFLOW
= 0.0000E+00
* BIN WID = 0.1998E-01      * MEAN VALUE   = 0.3980E+00      * R . M . S = 0.2434E+00

done.
krdriv: message: bye.
```