EXCALIBUR - a Monte Carlo program to evaluate all four-fermion processes at LEP 200 and beyond

Berends, F.A.; Pittau, R.; Kleiss, R.H.P.

DOI
10.1016/0010-4655(94)00138-R

Publication date
1995

Published in
Computer Physics Communications

ELSEVIER

Computer Physics Communications 85 (1995) 437–452

Computer Physics Communications

EXCALIBUR – a Monte Carlo program to evaluate all four-fermion processes at LEP 200 and beyond ★

F.A. Berends [a], R. Pittau [a], R. Kleiss [b,*]

[a] *Instituut-Lorentz, University of Leiden, P.O. Box 9506, 2300 RA Leiden, The Netherlands*

[b] *NIKHEF-H, P.O. Box 41882, 1009 DB Amsterdam, The Netherlands*

Received 2 September 1994

Abstract

A Monte Carlo program is presented that computes all four-fermion processes in e^+e^- annihilation. QED initial state corrections and QCD contributions are included. Fermions are taken to be massless, allowing a very fast evaluation of the matrix element. A systematic, modular and self-optimizing strategy has been adopted for the Monte Carlo integration, which serves also as an example for further event generators in high energy particle physics.

PROGRAM SUMMARY

Title of program: EXCALIBUR

Catalogue number: ADAJ

Program obtainable from: CPC Program Library, Queen's University of Belfast, N. Ireland (see application form in this issue); R. Kleiss, NIKHEF-H, P.O. BOX 41882, 1009 DB Amsterdam, The Netherlands, t30@nikhefh.nikhef.nl; R. Pittau, Instituut-Lorentz, University of Leiden, P.O.B. 9506, 2300 RA Leiden, The Netherlands, rulgm0@leidenuniv.nl

Licensing provisions: none

Computer for which the program is designed and others on which it has been tested:
Computers: HP and SUN workstations

Operating systems under which the program has been tested: UNIX

Programming language used: FORTRAN 77

Memory required to execute with typical data: about 170 kbytes

No. of bits in a word: 32

No. of lines in distributed program, including test data, etc.: 3784

Keywords: decaying vector-boson production, all four-fermion processes, electroweak and QCD background, initial state QED radiation, multichannel Monte Carlo approach

Nature of physical problem
Heavy vector boson production will be investigated at e^+e^- colliders in a wide range of energies. At LEP II, the relevant process is

$$e^+e^- \rightarrow W^+W^- . \tag{1}$$

At higher energies other processes like

$$e^+e^- \rightarrow Z Z , \tag{2}$$

★ This research has been partly supported by EU under contract number CHRX-CT-92-0004.

* E-mail: t30@nikhefh.nikhef.nl.

SSDI 0010-4655(94)00138-3

$$e^+e^- \rightarrow W e \nu_e \ , \tag{3}$$

$$e^+e^- \rightarrow Z e^+e^- \ , \tag{4}$$

$$e^+e^- \rightarrow Z \nu_e \bar{\nu}_e \ , \tag{5}$$

become important. The detected experimental signal for all above processes is a four-fermion final state. Therefore, a Monte Carlo program being able to take into account both signal and background electroweak diagrams for *all* four-fermion processes is required. QED initial state radiation and QCD background play also an important rôle and have to be included.

Method of solution

An *event generator* is the most suitable choice for a program to be able to deal with the above physical problem, since each generated event is a complete description of the momenta of the produced particles and any experimental cut can be easily implemented. There are two basic difficulties. First of all the number of Feynman diagrams can be very large. Secondly, taking into account also the background diagrams, the peaking structure of the matrix element squared is very rich, so that a straightforward integration over the allowed phase space is impractical. The former problem can be solved by using spinorial techniques to compute the amplitudes and taking massless fermions. The latter requires the use of a *multichannel* approach, where the integration variables are generated according to distributions that approximately reproduce the peaking behaviour of the integrand, so reducing the estimated Monte Carlo error.

Since one wishes to take into account *all* possible final state (that means to have from 3 to 144 different Feynman diagrams, many of them leading to different peaks in the phase space), a systematic and automatic procedure for both the generation of the Feynman diagrams and the phase space integration is unavoidable, together with an algorithm for the self-optimization of the predetermined probabilities used to choose the various *channels*.

All that has been implemented in EXCALIBUR. This paper serves also as an example of the entire procedure to be used to build future *event generators*.

Typical running time

about 100 events per second on HP, depending on the chosen physical process

LONG WRITE-UP

1. Introduction

In the near future LEP II will become operative in the energy region around 200 GeV. The physics relevant at higher energies will be investigated at the next generation of e^+e^- linear colliders. Many interesting physics issues can be studied and one of them is gauge-boson production. Around 200 GeV events with the signature of two produced W's have a large cross section, while single boson production processes become important with increasing energy [1]. One can distinguish five sizeable reactions (Eqs. (1)–(5)) in which gauge bosons are produced. Due to the fact that the massive bosons are unstable particles, all those processes end up with a detectable 4-fermion final state to which many Feynman diagrams can contribute. Some of them are related to the reactions (1)–(5) (signal diagrams); others are not (background diagrams). For this reason a precise knowledge of all possible processes

$$e^+ e^- \rightarrow 4 \text{ fermions} \tag{6}$$

is unavoidable in order to make comparison with experiment [2].

In addition to these background effects, one wants to be able to study *any* experimental distribution, taking into account the dominant radiative corrections effects, and the possibility to implement *any* experimental cut. To solve these problems we wrote an event generator, which can handle all diagrams leading to a specified 4-fermion final state (with, of course, the option of a restriction to the signal diagrams), and that incorporates the LL $\mathcal{O}(\alpha)$ and $\mathcal{O}(\alpha^2)$ initial state radiation (ISR), with exponentiation of the remaining soft-photon effects [3]. Furthermore, with a four-quark final state, QCD diagrams are present as well, giving non-negligible effects that have been also included [4].

It should be noted that even when cross sections do not dramatically change under inclusion of tiny effects, there are quantities that are very sensitive to any small correction. Among them is the average energy ϵ radiated by the beams. A precise knowledge of ϵ is required at LEP II when the reconstruction of the jet invariant mass distributions is performed to measure the W mass [4,5]. In addition, ϵ is also very sensitive to the imposed experimental cuts [3], so that, once more, a Monte Carlo approach is to be preferred.

In order to build a fast program we have taken the limit of vanishing fermion masses. Even if this implies the *absence* of diagrams where a Higgs boson couples to the fermions – and therefore we cannot compute the Higgs signal – we can at least estimate the background. On the other hand, the inclusion of the leading Higgs signal is trivial because only few diagrams account for it and, due to their helicity structure, do not interfere with all the others in the limit of massless fermions. However, this has not been implemented.

In the rest of this paper we shall describe EXCALIBUR, our event generator to compute *all* 4-fermion processes in e^+e^- collisions, including QED initial state corrections and QCD diagrams. The general structure of the code is flexible enough to deal with physics at the energy scales from 100 GeV to 1 TeV.

2. Theory and general features

There are two sources of complications. First of all one has to generate and compute all possible Feynman diagrams contributing to any given final state. Then the Monte Carlo integration has to be performed.

As explained in Ref. [2] the former problem can be efficiently solved by using spinorial helicity techniques. The amplitudes receive contributions from Abelian and non-Abelian graphs, with two distinct topological structures (see Fig. 1). In these so-called *generic* diagrams, all particles are assumed to be outgoing: assigning two fermion legs to be the initial-state fermions (by crossing), the actual Feynman diagrams are generated. The particles and antiparticles can each be assigned in six ways to the external lines (in principle). This gives 36 possible permutations. The Abelian diagrams are built by selecting, for each permutation, only those cases in which the exchanged bosons, which may be W^+, W^-, Z or γ, give rise to existing and charge conserving vertices. In the non-Abelian diagrams, two of the vector bosons are fixed to be W^+ and W^-, and the third one can be Z or γ. This procedure gives, for the Abelian graphs, a maximum of 144 different diagrams, and at most 8 for the non-Abelian diagrams.

The spinorial structure of each diagram can always be written in such a way that a particular combination of axial and vector couplings factorizes for a given helicity assignment. For example, if a_i and v_i are the axial and vector couplings in the vertices of the abelian diagram of Fig. 1, the following equation holds

$$\begin{aligned}
&\bar{u}_\lambda(1)\gamma^\mu(v_1+a_1\gamma_5)u_\lambda(2)\\
&\times\bar{u}_\rho(3)\gamma_\mu(v_2+a_2\gamma_5)(\not{p}_1+\not{p}_2+\not{p}_3)\gamma_\nu(v_3+a_3\gamma_5)u_\rho(4)\\
&\times\bar{u}_\sigma(5)\gamma^\nu(v_4+a_4\gamma_5)u_\sigma(6)=\sum_{\alpha,\beta,\tau=\pm}\mathcal{P}(\lambda\beta,\rho\alpha,\sigma\tau)\,A(\lambda,\rho,\sigma;1,2,3,4,5,6),
\end{aligned}\tag{7}$$

where

$$\begin{aligned}
&\mathcal{P}(\lambda\beta,\rho\alpha,\sigma\tau)=P_{\lambda\beta}P_{\rho\alpha}P_{\sigma\tau}V_1^\beta V_2^\alpha V_3^\tau,\\
&P_{\lambda\beta}=\frac{1}{2}(1+\lambda\beta),\qquad V_1^\pm=v_1\pm a_1,\\
&V_2^\pm=(v_2\pm a_2)(v_3\pm a_3),\qquad V_3^\pm=v_4\pm a_4,\\
&A(\lambda,\rho,\sigma;1,2,3,4,5,6)=\bar{u}_\lambda(1)\gamma^\mu u_\lambda(2)\\
&\quad\times\bar{u}_\rho(3)\gamma_\mu(\not{p}_1+\not{p}_2+\not{p}_3)\gamma_\nu u_\rho(4)\times\bar{u}_\sigma(5)\gamma^\nu u_\sigma(6)\,.
\end{aligned}\tag{8}$$

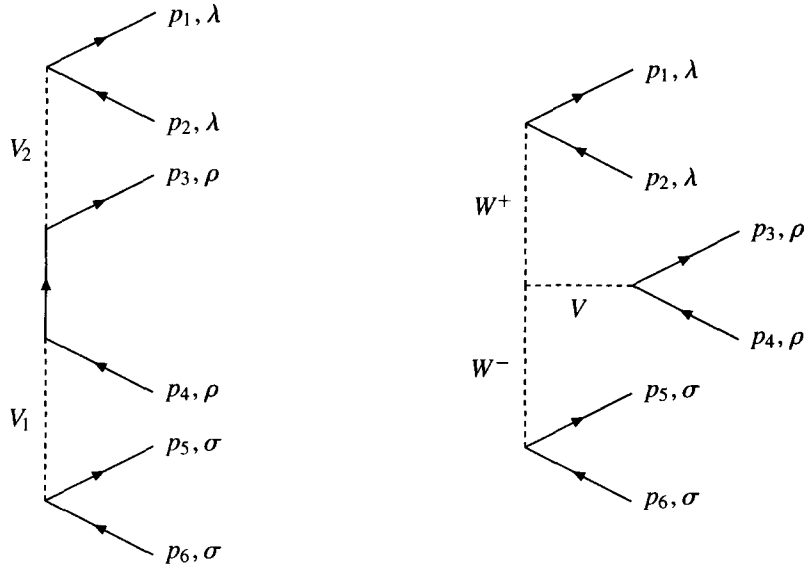

Fig. 1. Generic diagrams for four-fermion production. The fermion momenta and helicities, and the bosons are indicated. The bosons $V_{1,2}$ can be either Z, $W^{\pm}$, or γ; V can be either Z or γ.

Here we have disregarded the particle/antiparticle distinction since it is already implied by the assignment of the external momenta. The helicity labels $\lambda, \rho, \sigma = \pm$ determine the helicity of both external legs on a given fermion line. Using the Weyl–van der Waerden formalism for helicity amplitudes [6] (or, equivalently, the Dirac formalism of [7]), the expression A can easily be calculated [2]. It turns out that, for each permutation of the fermion momenta, all helicity combinations can be computed using only four independent complex functions.

The numerator in the non-Abelian diagrams can also be written in terms of the function A:

$$\begin{aligned}&\bar{u}_\lambda(1)\gamma_\alpha u_\lambda(2)\ \bar{u}_\rho(3)\gamma_\mu u_\rho(4)\ \bar{u}_\sigma(5)\gamma_\nu u_\sigma(6)\\&\quad\times 2\ \{g^{\mu\alpha}(p_1+p_2)^\nu + g^{\alpha\nu}(p_5+p_6)^\mu + g^{\nu\mu}(p_3+p_4)^\alpha\}\\&\quad= A(\lambda,\rho,\sigma;1,2,3,4,5,6) - A(\sigma,\rho,\lambda;5,6,3,4,1,2)\ .\end{aligned} \tag{9}$$

Thus, for massless fermions, every helicity amplitude consists of a sum of very systematic, and relatively compact, expressions.

When four quarks are present in the final state, one has to add the concomitant QCD production channels, and also the production of a quark pair and two gluons, since both types of final states will appear as jets. The former contribution, which we call *interfering* QCD background, is easily implemented once all electroweak diagrams have been computed as shown before. In fact, it is enough to add gluons wherever photons connect quark lines [4] (of course the correct QCD coupling and colour structure should be taken into account). Finally, the latter process can be efficiently computed using the recursion relations of Ref. [8]. Since it does not interfere with the other diagrams, we have written a separate event generator to get this contribution [9]. For the sake of brevity we do not describe it here. However, we point out that, as for the Monte Carlo integration, it has been built following exactly the same strategy used in EXCALIBUR.

The problem of the integration over the final fermion momenta can be solved using a *multichannel* approach [10,2]. If $f(\Phi)$ denotes the matrix element squared and $d\Phi$ the 8-dimensional massless phase space integration element, one has to compute

$$\sigma = \int f(\Phi)\, d\Phi\, \theta(cuts) \ , \tag{10}$$

where $\theta(cuts)$ stands for any kind of experimental cut that, in a Monte Carlo approach, is implemented by simply putting $f(\Phi) = 0$ in the unwanted region of the phase space.

In order to reduce the variance of the integrand, and therefore the Monte Carlo error, it is convenient to introduce an analytically integrable function $g(\Phi)$, called the *local density*, that exhibits approximately the same peaking behaviour of $f(\Phi)$ and is *unitary*, that is, a normalized probability density:

$$\int g(\Phi)\, d\Phi = 1 \ . \tag{11}$$

By multiplying and dividing the integrand by $g(\Phi)$, the cross section can be rewritten as follows

$$\sigma = \int w(\Phi(\rho))\, d\rho\, \theta(cuts) \ , \tag{12}$$

where the new integrand

$$w(\Phi(\rho)) = \frac{f(\Phi)}{g(\Phi)} \tag{13}$$

is a smoother function of the new set of variables $\{\rho_i\}$ defined by

$$d\rho = g(\Phi) d\Phi \ , \quad 0 < \rho_i < 1 \ , \tag{14}$$

so that the variance of $w(\rho)$ is smaller than the variance of $f(\Phi)$.

When the peaking structure of the matrix element squared is very rich one set of new integration variables $\{\rho_i\}$ can only describe well a limited number of peaks. Therefore a *multichannel* approach is required in which

$$g(\Phi) = \sum_{i=1}^{N} \alpha_i\, g_i(\Phi) \ , \qquad \sum_{i=1}^{N} \alpha_i = 1 \ , \qquad \int g_i(\Phi) d\Phi = 1 \ , \tag{15}$$

and where every $g_i(\Phi)$ describes a particular peaking structure of $f(\Phi)$. Note that the conditions on the α_i and $g_i(\Phi)$ ensure unitarity of the algorithm, *i.e.* probability is explicitly conserved at each step of the algorithm, without additional normalization factors at any stage. The numbers α_i are called *a-priori weights* and, although their numerical values are in principle unimportant, they can be used, in practice, to reduce the Monte Carlo error [11].

In EXCALIBUR we have dealt with the problem of the construction of the $g_i(\Phi)$ in a very modular and systematic way. Firstly, we have singled out all possible kinematical diagrams occurring in a four-fermion final state (see Fig. 2). They are pictures, inspired by the Feynman diagrams, which represent the various peaking structures of the matrix element and indicate which variables are most appropriate to a given $g_i(\Phi)$. The explanation of the pictures will be given in Subsection 3.1. Secondly, we have written all building blocks (that is subroutines) necessary for the calculation. Finally, we have put them together to form the $g_i(\Phi)$.

QED corrections are implemented using the structure-function formalism [12,3]. Each of the incoming fermions is assumed to have its energy degraded by the emission of photons parallel to the beam. For the energy distribution of the fermion after radiation we take a structure function Φ that incorporates the leading log $\mathcal{O}(\alpha)$ and $\mathcal{O}(\alpha^2)$ initial state radiation with exponentiation of the remaining soft-photon effects. Its expression can be found in [3]. Our model for the total radiative cross section is then

$$\sigma(s) = \int_0^1 \int_0^1 dx_1\, dx_2\, \Phi(x_1)\, \Phi(x_2)\, \sigma_0(x_1 x_2 s) \ , \tag{16}$$

where σ_0 is the non-radiative cross section and x_1, x_2 represent the energy content of the incoming fermions after radiative emission. This provides an adequate description of the leading QED effects.

3. Program structure

We shall now describe in some detail the salient features and strategies adopted in EXCALIBUR. The Program consists of two parts: the evaluation of the matrix element and the event generation. Both steps require an initialization, according to the chosen final state. Roughly speaking it means that the Feynman diagrams and the kinematical channels have to be built. This initialization is done in SUBROUTINE SETPRO, the matrix element is evaluated in SUBROUTINE DIAGA and SUBROUTINE MATRIX, while nearly all the rest is devoted to the event generation and Monte Carlo integration.

3.1. Subroutine SETPRO

We already described the algorithm used to construct the Feynman diagrams through a big do loop over all 36 permutations of the six fermion momenta. In SUBROUTINE SETPRO the variable KPERM(1:6,1:36) explicitly contains all these permutations, and IPHASE(1:36) the corresponding relative phase. The constructed Abelian (non-Abelian) diagrams are stored in JJ(1:16,NDAB) (JN(1:16,NNAB)), where the first index contains information about the particles involved in the process, the vector boson propagators and the momenta assignment, and the second one enumerates each diagram. In the output each constructed diagram is printed out together with its list number. For particular studies or checks, we give the possibility to switch off diagrams. This can be achieved by putting the variables KAO(I) = 0 (K1O(J) = 0), for the corresponding unwanted Abelian (non-Abelian) diagrams (I=1:NDAB, J=1:NNAB). SUBROUTINE SETPRO also contains the the input parameters of the program. They are α (ALPHA), α_s (ALS, relevant for 4-quarks final states), M_Z (ZM), M_W (WM), $\sin^2\theta_W$ (STH2), Γ_W (WW) and Γ_Z (WZ). The statistical factor STATFAC and the colour factor FCOL are evaluated according to the chosen final state. Furthermore, the coupling combinations $V_i^{\pm}$ of Eq. (8) (and those occurring in the non-Abelian case) are computed. It may happen that, for a particular helicity combination, one or more of the $V_i^{\pm}$ are zero. In the latter cases there is no point in computing the corresponding function A (see Eq. (7)). As a result, less than four independent complex functions are required to evaluate the spinorial part of the diagram. In order to have a fast evaluation of the matrix element those cases have to be excluded. This is achieved by introducing two *occupation matrices* NC(1:36) and NOC(1:36,1:4). For each of the 36 permutation, NC is set zero if the corresponding permutation does not give any Feynman diagram, and, if it does, NOC indicates which complex functions are needed. Through COMMON/AREA3/ these matrices are passed to SUBROUTINE DIAGA, where only those helicity combination for which NOC and NC are different from zero are computed.

Two more operations are performed in SUBROUTINE SETPRO, namely the choice of the kinematical channels for the Monte Carlo integration and the computation of the QCD interfering background.

We singled out a maximum number of 26 kinematical channels. They are given in Fig. 2, together with the name of the corresponding subroutines in EXCALIBUR, and are inspired by all possible occurring Feynman diagrams. Fermionic lines have an arrow, a wavy line represents a photon and a dashed line can be either Z or W (this gives 26 channels). Solid lines connect topological equivalent points. That is a *t-channel* solid line means isotropic angular distributions between the connected fermions while a *s-channel* solid line stands for photon or massive vector boson propagators. Since they only give rise to an s dependent behaviour, the peaking structure relevant for the integration over the final momenta is not affected by them. As an example, with those conventions it is easy to recognize that the last channel RAMBO4 represents an isotropic 4-body decay. If the *inspiring* Feynman diagrams exist, the variables NCHA(1:26,1:48) are set equal to one, where the first index runs over the possible channels and the second one labels the permutation of the final momenta. The number

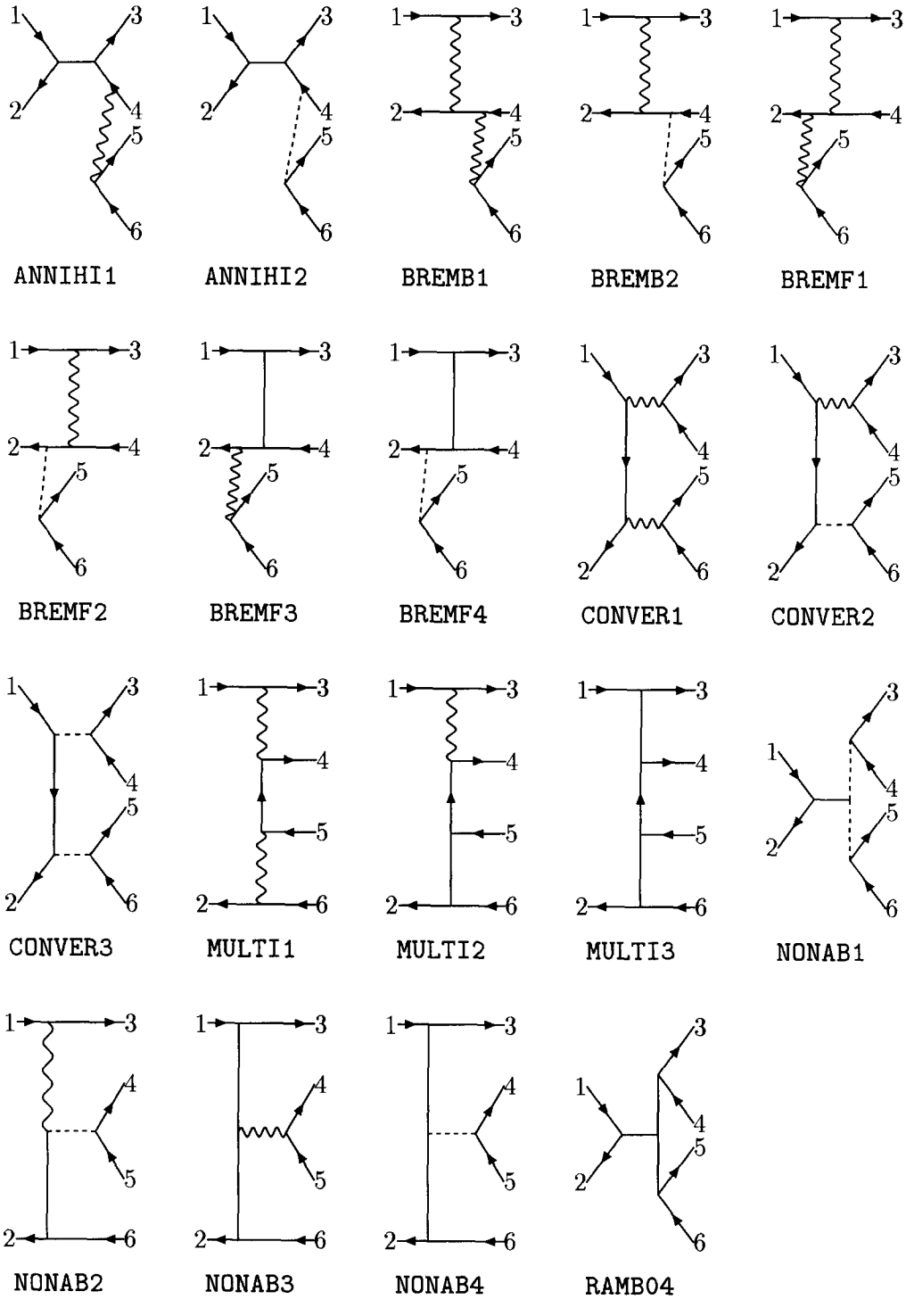

Fig. 2. Kinematical diagrams in EXCALIBUR.

48 is explained as follows. There are 24 permutations of the four final momenta but, for some channel, the case where the initial state labels 1 and 2 refer to e^+ and e^- respectively must be distinguished from the case where they refer to e^- and e^+. This gives 48 possible permutations. Depending on the topology, there are symmetries among the final momenta that have to be taken into account in order to have a minimum number of kinematical channels. For example, in channel NONAB1, permutation 3456 of the final momenta is equivalent to permutation 5634. This *symmetrization* is automatically performed by the program. When the initialization in SUBROUTINE SETPRO is completed, variable NCT indicates the number of found kinematical channels. In the output file they are printed out together with a list number I. An array has been introduced (NCTO), such that putting NCTO(I)= 0 excludes channel with number I (I=1:NCT) when the Monte Carlo integration is performed. This can be used to increase the speed of the program, by switching off those channels for which the procedure of self-optimization (see below) gives very small a-priori weights.

The interfering QCD background is added as an extra contribution proportional to the ratio $\alpha_s/\alpha QQ'$ (the variable GRAP), for those amplitudes where a photon connects two quarks of charge Q and Q'.

This concludes the description of SUBROUTINE SETPRO. Since all possible initializations are performed there, the structure of the rest of the program can be simple and fast.

3.2. Subroutines DIAGA and MATRIX

In SUBROUTINE DIAGA numerators and denominators of all found Feynman diagrams for which NOC and NC are non-vanishing are computed *at once* using the Weyl–van der Waerden formalism. It means that, for each generated event, SUBROUTINE DIAGA is called just once and not n times, where n is the total number of Feynman diagrams. As for the computational speed, this is very important.

In SUBROUTINE MATRIX(SQUAREM) the matrix element squared (SQUAREM) is calculated by putting together the numerators and denominators computed in SUBROUTINE DIAGA and the coupling combinations of Eq. (8) evaluated in SUBROUTINE SETPRO. Since computing the colour factor and QCD interfering background in a four-quark final state with colour labels i, j, l and m requires the part of the amplitude proportional to $\delta_{ij}\delta_{lm}$ to be distinguished from that proportional to $\delta_{il}\delta_{jm}$ [4], the constructed amplitudes in SUBROUTINE MATRIX take care of both contributions separately.

3.3. Phase space generation and integration

In the MAIN of EXCALIBUR the variables XR1 and XR2, representing the energy content x_1 and x_2 of the incoming fermions after radiative emission (Eq. (16)), are generated. Then, the initial configuration of the momenta *in the center of mass frame of the event after ISR* is set calling SUBROUTINE MOMSET and the cuts imposed on the momenta in the Lab frame are rewritten in terms of cuts in the center of mass frame. The kinematical channels are called using SUBROUTINE ADDRESS(LFLAG,NC,NN,DJ). When LFLAG is set 0, the channel number NC, with the momenta permutation labelled by NN, is used for generating the momenta and computing the local density DJ. If LFLAG= 1 the actual momenta configuration is used to compute DJ. The choice of the channel to use is performed, as in Ref. [10], on the basis of the actual values of the a-priori weights α_i by defining the cumulative numbers $\beta_i = \alpha_1 + \cdots + \alpha_i$, taking a random number uniformly distributed between 0 and 1 and choosing channel i if $\beta_{i-1} < z < \beta_i$.

In SUBROUTINE MOMARRAY the generated four momenta are put in a big array PM(0:4,0:900) and stored in

```
COMMON/MOMENTA/ROOTS,XR1,XR2,PM(0:4,0:900)
```

(ROOTS is the center of mass energy of the event). The first index refers to the component of the momenta (0 represents the energy and 4 is the four-momentum squared). As for the second index, the following self-explanatory conventions are used:

```
PM(I,34) ≡ PM(I,43)= PM(I,3)+PM(I,4)  etc.
PM(I,643) ≡ PM(I,346) ≡ ... = PM(I,6)+PM(I,4)+PM(I,3)  etc.
```

Besides, but only if the first digit refers to an incoming momentum (notice the correspondence $7 \rightarrow 1$, $8 \rightarrow 2$)

```
PM(I,734) ≡ PM(I,1)-PM(I,3)-PM(I,4)  etc.
PM(I,851) ≡ PM(I,2)-PM(I,5)-PM(I,1)  etc.
```

Each channel is constructed in a very modular way by putting together basic subroutines that describe different parts of its peaking structure. In Ref. [2] an example of the construction of channels BREMB2 and CONVER2 of Fig. 2 is given. There are 10 of these basic subroutines. They are the *building blocks* of the whole generation procedure. For the sake of brevity we do not list them here. They are well commented in the program. We only notice that, in building the kinematical channels, every t-channel exchanged massive vector boson is always assumed to give a flat angular distribution between the initial and the final fermion. This is done in order to avoid proliferation in the number of channels. In our experience, this gives a very good approximation at center of mass energies up to 500 GeV, a good approximation at higher energies up to 1 TeV and may cause large Monte Carlo errors at 2 TeV. Of course the Monte Carlo program remains correct, but higher statistics runs are required. However, adding channels to map this high energy kinematical behaviour is trivial, because EXCALIBUR already contains all needed ingredients.

As far as the self-optimization of the integration is concerned, a detailed description of the iterative algorithm implemented in EXCALIBUR may be found in Ref. [11]. Here we point out that two variables have to be chosen by the user, namely the maximum number of iterations ISTEPMAX (in the input list) and the number of point NOPT used for the self-optimization (in the MAIN of the program). Then, for each iteration, NOPT/ISTEPMAX points (including zero-weight events) are used to compute the a-priori weights. We found that, with 4–5 hundred thousand points, a good choice is NOPT= 100 000 and ISTEPMAX= 10. However, when very stringent cuts are applied, the majority of the events falls outside the allowed region, so that the ratio NOPT/ISTEPMAX may be a very small number. This causes a bad estimate of the best a-priori weights to be used. In those cases it is convenient to either increase NOPT or decrease ISTEPMAX.

In the input list one has to specify the set of *standard* cuts as specified in the next section. Any other type of cut must be implemented directly in SUBROUTINE CUTS(LNOT), where

```
COMMON/AREA10/PM1(0:4,1:6),PM4(12:65),OMCT1(1:6,3:6)
```

contains the four momenta computed *in the Lab frame* (PM1), the invariant mass squared among all possible particles pairs (PM4) and the quantities $1 - \cos\theta_{ij}$ (OMCT1). If the event is rejected LNOT= 1, and the weight is put equal to zero.

Finally, all weights (computed as in Eq. (13)) are collected using SUBROUTINE INBOOK and the Monte Carlo results called through SUBROUTINE OUTBOK.

4. Input

The meaning of the input parameters is the following:

NPROCESS(INTEGER)
The number of processes to be computed.

N(INTEGER)
The number of points for the Monte Carlo integration.

ISTEPMAX(INTEGER)
The number of iterations for optimizing the a-priori weights.

OUTPUTNAME(CHARACTER*15)
The name of the output file.

KREL(INTEGER)
It selects the signals. If KREL= 0 all Feynman diagrams are taken into account. If KREL= 1-5 only those leading to reactions of Eqs. (1)–(5).

LQED(INTEGER)
It includes (1) or excludes (0) ISR.

ROOTSMUL(REAL*8)
The total energy of the colliding e^+ and e^-. All energies are in GeV.

SHCUT(REAL*8)
Minimum value of the invariant mass squared of the event after QED radiation.

ECUT(3)(REAL*8)
Minimum energy of particle number 3.

ECUT(4)(REAL*8)
Minimum energy of particle number 4.

ECUT(5)(REAL*8)
Minimum energy of particle number 5.

ECUT(6)(REAL*8)
Minimum energy of particle number 6.

SCUT(3,4)(REAL*8)
Minimum value of $(p(3)+p(4))^2$. All invariant masses are in GeV^2.

SCUT(3,5)(REAL*8)
Minimum value of $(p(3)+p(5))^2$.

SCUT(3,6)(REAL*8)
Minimum value of $(p(3)+p(6))^2$.

SCUT(4,5)(REAL*8)
Minimum value of $(p(4)+p(5))^2$.

SCUT(4,6)(REAL*8)
Minimum value of $(p(4)+p(6))^2$.

SCUT(5,6)(REAL*8)
Minimum value of $(p(5)+p(6))^2$.

CMAX(1,3)(REAL*8)
Maximum value of $\cos\theta$ between particle 1 and 3.

CMAX(1,4)(REAL*8)
Maximum value of $\cos\theta$ between particle 1 and 4.

CMAX(1,5)(REAL*8)
Maximum value of $\cos\theta$ between particle 1 and 5.

CMAX(1,6)(REAL*8)
Maximum value of $\cos\theta$ between particle 1 and 6.

CMAX(2,3)(REAL*8)
Maximum value of $\cos\theta$ between particle 2 and 3.

CMAX(2,4)(REAL*8)
Maximum value of $\cos\theta$ between particle 2 and 4.

CMAX(2,5)(REAL*8)
Maximum value of $\cos\theta$ between particle 2 and 5.

CMAX(2,6)(REAL*8)
Maximum value of $\cos\theta$ between particle 2 and 6.

CMAX(3,4)(REAL*8)
Maximum value of $\cos\theta$ between particle 3 and 4.

CMAX(3,5)(REAL*8)
Maximum value of $\cos\theta$ between particle 3 and 5.

CMAX(3,6)(REAL*8)
Maximum value of $\cos\theta$ between particle 3 and 6.

CMAX(4,5)(REAL*8)
Maximum value of $\cos\theta$ between particle 4 and 5.

CMAX(4,6)(REAL*8)
Maximum value of $\cos\theta$ between particle 4 and 6.

CMAX(5,6)(REAL*8)
Maximum value of $\cos\theta$ between particle 5 and 6.

PAR(3)(CHARCTER*8)
Produced fermion with label 3 (to be chosen among 'EL', 'NE', 'MU', 'NM', 'TA', 'NT', 'DQ', 'UQ', 'SQ', 'CQ', 'BQ', 'TQ').

PAR(4)(CHARCTER*8)
Produced antifermion with label 4.

PAR(5)(CHARCTER*8)
Produced fermion with label 5.

PAR(6)(CHARCTER*8)
Produced antifermion with label 6.

5. Test Run Output

To conclude our description, we give an example of a typical calculation that can be performed with EXCALIBUR. One should be able to reproduce this output within the estimated Monte Carlo error (small

differences may occur because the quasi-random number generator used in the program is not strictly portable). Using an input file as follows

```
1              number of energy points
250000         number of Monte Carlo points
10             number of iterations in a.p.weights optimization
output         output program name
0              krel (signal:  0,1,2,3,4,5)
1              lqed (0 or 1)
190.d0         total energy (GeV)
0.d0           cut on reduced inv. mass squared after ISR
0.d0           ecut_3
0.d0           ecut_4
20.d0          ecut_5
20.d0          ecut_6
0.d0           scut_34
0.d0           scut_35
0.d0           scut_36
0.d0           scut_45
0.d0           scut_46
100.d0         scut_56
1.d0           cmax_13
1.d0           cmax_14
0.9d0          cmax_15
0.9d0          cmax_16
1.d0           cmax_23
1.d0           cmax_24
0.9d0          cmax_25
0.9d0          cmax_26
1.d0           cmax_34
1.d0           cmax_35
1.d0           cmax_36
1.d0           cmax_45
1.d0           cmax_46
0.9d0          cmax_56
mu             produced fermion     (3)
nm             produced antifermion (4)
uq             produced fermion     (5)
dq             produced antifermion (6)
```

and the values ALPHA= 1./128., ZM= 91.16, WM= 80.22, STH2= 0.226 , WW= 2.03, WZ= 2.53 we get the following output file

```
                    output

All Feynman diagrams

sqrt(s)  =      .190000D+03
n_points =   250000
```

```
istepmax =       10

energy cuts with ecut_3  = .0
                 ecut_4  = .0
                 ecut_5  = 20.0
                 ecut_6  = 20.0

cut on        s*x1r*x2r  = .0

mass cuts with  scut_34  = .0
                scut_35  = .0
                scut_36  = .0
                scut_45  = .0
                scut_46  = .0
                scut_56  = 100.0

angle cuts with cmax_13  = 1.0
                cmax_14  = 1.0
                cmax_15  = .9
                cmax_16  = .9
                cmax_23  = 1.0
                cmax_24  = 1.0
                cmax_25  = .9
                cmax_26  = .9
                cmax_34  = 1.0
                cmax_35  = 1.0
                cmax_36  = 1.0
                cmax_45  = 1.0
                cmax_46  = 1.0
                cmax_56  = .9

I.S.R. INCLUDED

s^2_thet =     .226000D+00
Z-mass   =     .911600D+02
Z-width  =     .253000D+01
W-mass   =     .802200D+02
W-width  =     .203000D+01
1/alpha  =     .128000D+03
alpha_s  =     .103000D+00

process:  antiel(1) el(2) ---> mu(3) antinm(4) uq(5) antidq(6)

          abelian diagrams                                  phase

1:  [el(1),el(2)] Z [mu(3),mu,nm(4)] W [uq(5),dq(6)]    ph=  1
2:  [el(1),el(2)] G [mu(3),mu,nm(4)] W [uq(5),dq(6)]    ph=  1
3:  [el(1),el(2)] Z [uq(5),uq,dq(6)] W [mu(3),nm(4)]    ph=  1
```

```
4:  [el(1),el(2)] G [uq(5),uq,dq(6)] W [mu(3),nm(4)]    ph=  1
5:  [mu(3),nm(4)] W [uq(5),dq,dq(6)] Z [el(1),el(2)]    ph=  1
6:  [mu(3),nm(4)] W [uq(5),dq,dq(6)] G [el(1),el(2)]    ph=  1
7:  [uq(5),dq(6)] W [el(1),ne,el(2)] W [mu(3),nm(4)]    ph=  1
8:  [uq(5),dq(6)] W [mu(3),nm,nm(4)] Z [el(1),el(2)]    ph=  1

         non-abelian diagrams                           phase

1:  [uq(5),dq(6)] [el(1),el(2)] [mu(3),nm(4)] (WZW)     ph=  1
2:  [uq(5),dq(6)] [el(1),el(2)] [mu(3),nm(4)] (WGW)     ph=  1

         kinematical diagrams

    channel                     permutation

1:  annihi2(wm)                 1 2 3 4 5 6
2:  annihi2(wm)                 1 2 4 3 5 6
3:  annihi2(wm)                 1 2 5 6 3 4
4:  annihi2(wm)                 1 2 6 5 3 4
5:  conver3(wm)                 1 2 5 6 3 4
6:  nonab1(wm)                  1 2 3 4 5 6
7:  rambo4                      1 2 3 4 5 6

 ********** weights analysis **********

 *** variable number  1 ************
sum(w**0)       .250000D+06,  sum(w**1)         .135757D+06
sum(w**2)       .254349D+06,  sum(w**3)         .803758D+06
sum(w**4)       .434411D+07
maximum         .221013D+02,  max.in buffer     .148670D+02
no.weights=0          59062,  no.weights<0               0
estimator x:     .543028D+00
estimator y:     .289008D-05
estimator z:     .730267D-15
average estimate :     .543028D+00
              +\-      .170002D-02
variance estimate:     .289008D-05
              +\-      .270235D-07
efficiency for all weights     :    2.457 %
efficiency for non-zero weights :    3.217 %
overshoot factor of histogram  :    1.487
the distribution of the non-zero weights:
50, log scale;  entries under,inside,over: 0  190928      10

  .1487E+01   .1631E+06     i******************************i
  .2973E+01   .2157E+05     i************************     i
  .4460E+01   .4936E+04     i*********************        i
  .5947E+01   .1089E+04     i*****************            i
```

```
  .7433E+01    .1470E+03     i**************                i
  .8920E+01    .3600E+02     i*********                     i
  .1041E+02    .1600E+02     i********                      i
  .1189E+02    .1100E+02     i*******                       i
  .1338E+02    .7000E+01     i******                        i
  .1487E+02    .5000E+01     i******                        i

differences in the computation
of the a-priori weights:

    diff( 1)=  1.95407687917905
    diff( 2)=  .891414805616945
    diff( 3)=  .7677390572607608
    diff( 4)=  .6967978248575471
    diff( 5)=  .5755243287941117
    diff( 6)=  .7280986181406474
    diff( 7)=  .5755417220460467
    diff( 8)=  .5925286045906328
    diff( 9)=  .5632310939026423
    diff( 10)=  .598279075660049
    diff( 11)=  .5921705560137981
    diff( 12)=  .5181929881140225

  a-priori weights:

  1 :       .899493D-03
  2 :       .165631D-03
  3 :       .114732D-03
  4 :       .460900D-03
  5 :       .881137D+00
  6 :       .117221D+00
  7 :       .817789D-06
```

After information about input parameters and imposed cuts, the program prints out the used Feynman diagrams and kinematical channels. Then, the analysis of the weights giving the Monte Carlo estimate of the cross section (variable number 1) follows. In particular various sums of the weights to powers 0–4 are given as well as the maximum weight and that one in the buffer (that is in the interval of values used in the histogram that shows the weight distribution). The quantity x is the estimator of the average of the distribution defined, for N weights w_i, as

$$\frac{\sum_i w_i}{N}\,, \tag{17}$$

y is the estimator of the variance

$$\frac{1}{N}\left[\frac{\sum_i w_i^2}{N} - \frac{(\sum_i w_i)^2 - \sum_i w_i^2}{N(N-1)}\right]\,, \tag{18}$$

and z is an estimator for the variance of the variance, so that the error on the average and variance estimates are $\sqrt{y}$ and $\sqrt{z}$ respectively. In the example $\sigma = 0.5430 \pm 0.00170$ pb. As usual the efficiency is defined as

$\langle w \rangle / \max(w)$ and the overshoot factor is the ratio between the maximum weight and the maximum weight written in the buffer. In the histogram the 190928 non-zero weights are displayed according to their abundance in bins. Finally, the variables D of Ref. [11] (that measure, at each step in the optimization procedure, how well the actual set of a-priori weights approximates the behaviour of the optimal set) are printed out, together with the found best set of a-priori weights.

References

[1] K. Hagiwara, in: Physics and Experiments with Linear Colliders, R. Orava, P. Eerola and M. Nordberg, eds. (World Scientific, Singapore, 1992) p. 387.
[2] F.A. Berends, R. Kleiss and R. Pittau, All electroweak four-fermion processes in electron-positron collisions, INLO-PUB-1/94 and NIKHEF-H/94-08, Nucl. Phys. B, to be published.
[3] F.A. Berends, R. Kleiss and R. Pittau, Initial-state QED corrections to four-fermion production in e^+e^- collisions at LEP 200 and beyond, INLO-PUB-6/94 and NIKHEF-H/94-20, Nucl. Phys. B, to be published.
[4] R. Pittau, Four quark processes at LEP 200, Phys. Lett. B, to be published.
[5] S. Katsanevas et al., internal report DELPHI 92-166 PHYS (1992) 250.
[6] F.A. Berends and W.T. Giele, Nucl. Phys. B 294 (1987) 700.
[7] R. Kleiss and W.J. Stirling, Nucl. Phys. B 262 (1985) 235.
[8] F.A. Berends, W.T. Giele and H. Kuijf, Nucl. Phys. B 321 (1989) 39.
[9] R. Pittau, a Monte Carlo program to compute $e^+e^- \to 2$ gluons + 2 quarks, unpublished.
[10] J. Hilgart, R. Kleiss and F. Le Diberder, Comput. Phys. Commun. 75 (1993) 191.
[11] R. Kleiss and R. Pittau, Weight optimization in multichannel Monte Carlo, Comput. Phys. Commun. 83 (1994) 141.
[12] E.A. Kuraev and V.S. Fadin, Sov. J. Nucl. Phys. 41 (1985) 466.
G. Altarelli and G. Martinelli, Physics at LEP, CERN report 86-02, eds. J. Ellis and R. Peccei (1986).
F.A. Berends at al., Z Physics at LEP1, eds. G. Altarelli, R. Kleiss and C. Verzegnassi, CERN 89-08, Vol 1 (1989) p. 89.