DELPHI Collaboration DELPHI 99-4 PHYS 816
October 23, 2000

DELTGC

A Program for Four Fermion Calculations

O.P.Yushchenko and V.V. Kostyukhin
IHEP, Protvino, Russia

Abstract

The description of the program DELTGC suitable for four-fermion calculations is presented. DELTGC is designed with a special attention to the Anomalous Gauge Boson Couplings and can be used for total and differential cross-section calculations as well as for event generation.

1 Introduction

The DELTGC package offers powerful and convenient tools for the calculation of the cross section for the processes $e^+e^- \to 4f$ with special attention to the triple gauge boson couplings. It also supports the generation of single-weighted events for a particular final state or a mixture of them.

A number of packages can perform at present the same task (see, for example, the review in [1]). They give different possibilities and features and are widely used in practical calculations. In addition to the properties and features that are common to all the packages dealing with the 4-fermion final states at LEP II energies, DELTGC gives:

- Neutral TGC ($Z\gamma\gamma$, $ZZ\gamma$, and ZZZ vertices) in addition to the charged couplings;
- TGC decomposition of the amplitudes, so all the information suitable for the TGC studies for any particular process can be obtained in a single DELTGC run;
- Processes with CKM mixing;
- Careful treatment of the gauge cancellations, so the integration behavior remains stable in the full angular interval for the final-state fermions for all processes;
- Direct access to the matrix element, so it is possible to perform unbinned likelihood fitting using all the kinematical information without referring to any particular variable or distribution;
- Arbitrary complex cuts on the final fermion configurations;
- User-driven integration behaviour, so that if the user is not satisfied with the results of the integration (especially with the errors of any particular TGC-expansion coefficient), the integration can be continued without losing the information already obtained;
- Convenient access to the internal physical parameters;
- Z'-related cross-section expansion coefficients.

DELTGC is designed as a set of libraries to perform different actions:

- General Spirality and TGC Library (GSTL): a collection of routines to maintain the spirality calculus technique described in Section 2.1 as well as VVV vertex calculation in the representation of Section 2.2 with a set of initialization and general (such as Lorentz transformation, ISR, QCD and Coulomb corrections) routines;
- Amplitudes of $2 \to 4$ Library (A24L): a collection of routines to perform calculation of the matrix element squared for the process $e^+e^- \to 4f$;
- Adaptive Integration Library (AIL): a set of routines to perform kinematics transformations from the internal VEGAS [2] variables to the physical momenta and also containing the VEGAS routines;

and the DELTGC steering routines maintain an overall package behaviour.

These libraries are not independent and can be used as separate tools only after quite serious modifications, that can be hardly done by the user.

The next version of the DELTGC package will release more advanced, powerful, fast and package-independent libraries to make their usage more convenient in other applications. Additional features in the next DELTGC release will be:

- Calculation of cross-sections not only for single processes but for large classes of processes with definite signature;
- Matrix element access for definite or flavour-independent final states containing quarks, so the natural mixture of all flavours with all possible permutations and symmetrizations can be calculated in a single call;
- DELTGC - JETSET interface.

The DELTGC package was created and tested under VAX-VMS and DIGITAL UNIX. The current version of the library is placed in

```
disk$delphi2:[ioucht.deltgc]deltgc.olb      VAX-VMS version (VXCERN)

/afs/cern.ch/user/i/ioucht/deltgc/deltgc.o   DIGITAL UNIX (dxplus)
```

The library requires CERNLIB to be linked.

2 Theoretical Background

2.1 Spiral Calculus

It is common knowledge that the calculation of diagrams with more than 5 external lines using a conventional spurs calculation technique is a tedious task. Instead of summing the matrix element squared over the external particle polarizations, a number of methods were proposed [3, 4, 5], based on the calculation of amplitudes for different polarization states. For fermions the natural polarization states are those with definite spirality: in particular, this reduces the number of amplitudes if massless fermions are involved.

The projection on the spirality states can be easily done by expressing any vertex in terms of left- and right spirality projectors:

$$P_L = \frac{1-\gamma_5}{2} \qquad \text{and} \qquad P_R = \frac{1+\gamma_5}{2}$$

For example, the $\gamma f f$ vertex can be expressed as:

$$iQ_f\gamma_\mu = iQ_f\gamma_\mu(P_L + P_R).$$

Then, applying the well known relations

$$\gamma_5\gamma_\mu - \gamma_\mu\gamma_5 = 0, \text{ and } P_iP_j = \delta_{ij}P_i,$$

any Dirac string will take the form $\bar{\psi}(1)[a]P_{L/R}\psi(2)$ (or the sum of such forms), where ψ denotes either u or v Dirac spinors and $[a]$ represents any sequence of gamma matrices.

Here the method of handling spirality projected amplitudes that is suitable for the numerical and analytical calculation of Feynman diagrams with a large number of external particles is presented. This technique is an extension of the one proposed by Hagiwara [5].

2.1.1 General case

Let us consider u and v spinors that are the eigenstates of the chiral operators:

$$u^\pm(q) = \frac{1}{2}(1 \pm \gamma_5)u^\pm(q), \quad \overline{u}^\pm(q) = \overline{u}^\pm(q)\frac{1}{2}(1 \mp \gamma_5),$$

$$v^\pm(q) = \frac{1}{2}(1 \mp \gamma_5)v^\pm(q), \quad \overline{v}^\pm(q) = \overline{v}^\pm(q)\frac{1}{2}(1 \mp \gamma_5).$$

The γ-matrices in the chiral representation have the form:

$$\gamma_0 = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}; \;\; \vec{\gamma} = \begin{pmatrix} 0 & \vec{\sigma} \\ -\vec{\sigma} & 0 \end{pmatrix}; \;\; \gamma_5 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix},$$

with the Pauli matrices:

$$\sigma_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \;\; \sigma_2 = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \;\; \sigma_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix},$$

Making use of the Weil spinors

$$\begin{aligned}\chi_p^+ &= \frac{1}{2|p|(|p|+p_z)}\begin{pmatrix} |p|+p_z \\ p_x+ip_y \end{pmatrix},\\ \chi_p^- &= \frac{1}{2|p|(|p|+p_z)}\begin{pmatrix} -p_x+ip_y \\ |p|+p_z \end{pmatrix},\end{aligned}$$

and $\omega_\pm = \sqrt{E \pm |p|}$, the Dirac spinors can be represented as:

$$u^+(p) = \begin{pmatrix} \omega_+\chi_p^+ \\ \omega_-\chi_p^+ \end{pmatrix},\ v^-(p) = \begin{pmatrix} \omega_+\chi_p^+ \\ -\omega_-\chi_p^+ \end{pmatrix}$$

$$u^-(p) = \begin{pmatrix} \omega_-\chi_p^- \\ \omega_+\chi_p^- \end{pmatrix},\ v^+(p) = \begin{pmatrix} -\omega_-\chi_p^- \\ \omega_+\chi_p^- \end{pmatrix}$$

Or in a shorthand notation:

$$\psi = \begin{pmatrix} \psi_U \\ \psi_D \end{pmatrix}, \quad \overline{\psi} = (\psi_D^\dagger, \psi_U^\dagger),$$

and

$$u^\sigma_{U/D}(p) = \omega_{\pm\sigma}(p)\chi_p^\sigma, \quad v^\sigma_{U/D}(p) = (\mp)\omega_{\mp\sigma}(p)\chi_p^{-\sigma},$$

where σ is the spirality of the fermion. These spinors are normalized as follows:

$$\overline{u}(p,\lambda)u(p,\lambda) = 2m, \quad \overline{v}(p,\lambda)v(p,\lambda) = -2m.$$

If $|p| = -p_z$ the above spinors can be simplified to the form:

$$u^+(p) = \begin{pmatrix} 0 \\ \sqrt{E+|p|} \\ 0 \\ \sqrt{E-|p|} \end{pmatrix};\ v^-(p) = \begin{pmatrix} 0 \\ \sqrt{E+|p|} \\ 0 \\ -\sqrt{E-|p|} \end{pmatrix}$$

$$u^-(p) = \begin{pmatrix} -\sqrt{E-|p|} \\ 0 \\ -\sqrt{E+|p|} \\ 0 \end{pmatrix};\ v^+(p) = \begin{pmatrix} \sqrt{E-|p|} \\ 0 \\ -\sqrt{E+|p|} \\ 0 \end{pmatrix};$$

To fulfill the notation the explicit expressions for polarization vectors of the external bosons should be given. Let the momentum of the external boson with mass m in some coordinate system be $p^\mu = (E, p_x, p_y, p_z)$, then in the same system its polarization vectors are:

$$\begin{aligned}\epsilon^\mu(1) &= \frac{1}{|p|p_t}(0, p_xp_z, p_yp_z, -p_t^2),\\ \epsilon^\mu(2) &= \frac{1}{p_t}(0, -p_y, p_x, 0),\\ \epsilon^\mu(3) &= \frac{E}{|p|m}\left(\frac{|p|^2}{E}, p_x, p_y, p_z\right),\end{aligned}$$

where vectors $\epsilon(1)$ and $\epsilon(2)$ represent transverse polarization states, while $\epsilon(3)$ gives the longitudinal one and should vanish for massless bosons.

Generally speaking this explicit form of Dirac spinors and polarization vectors is quite sufficient to perform numerical calculations for any kind of Feynman diagram. The numerical procedure of the matrix element calculation can be as follows:

1. For any particular set of the external particle spirality states and momenta, construct general four-tensors originating from the Dirac strings in all diagrams. Any particular diagram is now a product of four-tensors and external particle momenta and polarizations.

2. Contract all indices performing explicit matrix and vector multiplication. Any particular diagram becomes a scalar after this procedure.

3. Add propagator factors for all diagrams.

4. Sum all spirality and polarization states and calculate the modulus of the sum. The resulting scalar represents the matrix element squared for the particular phase-space point.

The integration of the matrix element squared over the whole phase-space should be performed by a suitable integration procedure.

2.1.2 Massless case

Considering particular processes, say those involving massless fermions, the spirality technique gives the possibility to perform the calculations in a much more compact form, making use of explicit gamma-matrix decomposition, Fiertz identities for Weil spinors, and the special relations for Weil spinor products.

For zero mass fermions, the explicit form of the u and v spinors becomes:

$$u^+(p) = \frac{1}{\sqrt{|p|+p_z}} \begin{pmatrix} |p|+p_z \\ p_x + ip_y \\ 0 \\ 0 \end{pmatrix} = v^-, \qquad u^-(p) = \frac{1}{\sqrt{|p|+p_z}} \begin{pmatrix} 0 \\ 0 \\ -p_x + ip_y \\ |p|+p_z \end{pmatrix} = v^+,$$

and for the case $|p| = -p_z$:

$$u^+(p) = \begin{pmatrix} 0 \\ \sqrt{2|p|} \\ 0 \\ 0 \end{pmatrix} = v^-, \qquad u^-(p) = \begin{pmatrix} 0 \\ 0 \\ -\sqrt{2|p|} \\ 0 \end{pmatrix} = v^+.$$

The next step in the spiral calculus technique is the method of handling the scalar products of the external four-vectors (momenta or polarization vectors) and the gamma matrix.

Making use of the explicit form for the Dirac matrices, the "hat" value $\hat{a}$ can be expressed in the following form:

$$\hat{a} = a_0\gamma_0 - \vec{a}\vec{\gamma} = \begin{pmatrix} 0 & a_0 + a_i\gamma_i \\ a_0 - a_i\gamma_i & 0 \end{pmatrix} = \begin{pmatrix} 0 & a^+ \\ a^- & 0 \end{pmatrix},$$

where

$$a^+ = \begin{pmatrix} a_0 + a_z & a_x - ia_y \\ a_x + ia_y & a_0 - a_z \end{pmatrix}; \;\; a^- = \begin{pmatrix} a_0 - a_z & -a_x + ia_y \\ -a_x - ia_y & a_0 + a_z \end{pmatrix}.$$

In practical calculations a lot of simplifications will occur after "hat" decomposition for an arbitrary 4-vector a:

$$a^{\pm} = a_{\pm}\chi_+(a)\chi_+^{\dagger}(a) + a_{\mp}\chi_-(a)\chi_-^{\dagger}(a),$$

where $a_{\pm} = a_0 \pm |a|$.

Considering how the matrix $\hat{a}$ acts on the different spinors in the massless case, one obtains:

$$\hat{a}u_+ = \begin{pmatrix} 0 & a^+ \\ a^- & 0 \end{pmatrix} \begin{pmatrix} \psi \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ a^-\psi \end{pmatrix}; \; \hat{a}u_- = \begin{pmatrix} 0 & a^+ \\ a^- & 0 \end{pmatrix} \begin{pmatrix} 0 \\ \psi \end{pmatrix} = \begin{pmatrix} a^+\psi \\ 0 \end{pmatrix}.$$

Thus the string of the matrix $\hat{a}$ acts on the spinor like:

$$\hat{a}_1 \ldots \hat{a}_{2n} \begin{pmatrix} \chi \\ 0 \end{pmatrix} = \begin{pmatrix} a_1^+ a_2^- \ldots a_{2n-1}^+ a_{2n}^- \chi \\ 0 \end{pmatrix}; \; \hat{a}_1 \ldots \hat{a}_{2n+1} \begin{pmatrix} \chi \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ a_1^- a_2^+ \ldots a_{2n}^+ a_{2n+1}^- \chi \end{pmatrix}.$$

Analogously:

$$\hat{a}_1 \ldots \hat{a}_{2n} \begin{pmatrix} 0 \\ \chi \end{pmatrix} = \begin{pmatrix} 0 \\ a_1^- a_2^+ \ldots a_{2n-1}^- a_{2n}^+ \chi \end{pmatrix}; \; \hat{a}_1 \ldots \hat{a}_{2n+1} \begin{pmatrix} 0 \\ \chi \end{pmatrix} = \begin{pmatrix} a_1^+ a_2^- \ldots a_{2n}^- a_{2n+1}^+ \chi \\ 0 \end{pmatrix}.$$

These relations lead to a quite simple technique for reducing the string composed of Dirac spinors and gamma matrices to one of Weil spinors:

$$\begin{array}{lcl}
\begin{pmatrix} \bar{u}^-(1) \\ \bar{v}^+(1) \end{pmatrix} \Big| \gamma_1 \ldots \gamma_{2n+1} P_L \Big| \begin{pmatrix} u^-(2) \\ v^+(2) \end{pmatrix} & = & \bar{\chi}_1^- \gamma_1^+ \gamma_2^- \ldots \gamma_{2n+1}^+ \chi_2^- \\
\begin{pmatrix} \bar{u}^+(1) \\ \bar{v}^-(1) \end{pmatrix} \Big| \gamma_1 \ldots \gamma_{2n} P_L \Big| \begin{pmatrix} u^-(2) \\ v^+(2) \end{pmatrix} & = & \bar{\chi}_1^+ \gamma_1^- \gamma_2^+ \ldots \gamma_{2n}^+ \chi_2^- \\
\begin{pmatrix} \bar{u}^+(1) \\ \bar{v}^-(1) \end{pmatrix} \Big| \gamma_1 \ldots \gamma_{2n+1} P_R \Big| \begin{pmatrix} u^+(2) \\ v^-(2) \end{pmatrix} & = & \bar{\chi}_1^+ \gamma_1^- \gamma_2^+ \ldots \gamma_{2n+1}^- \chi_2^+ \\
\begin{pmatrix} \bar{u}^-(1) \\ \bar{v}^+(1) \end{pmatrix} \Big| \gamma_1 \ldots \gamma_{2n} P_R \Big| \begin{pmatrix} u^+(2) \\ v^-(2) \end{pmatrix} & = & \bar{\chi}_1^- \gamma_1^+ \gamma_2^- \ldots \gamma_{2n}^- \chi_2^+,
\end{array}$$

where $\gamma^{\pm}$ denotes the upper-right or lower-left blocks of the gamma-matrix. Similar relations, slightly more complicated, can be obtained for arbitrary mass assignments for the Dirac spinors.

After such reduction, one is left with the products of Weil strings with arbitrary contraction of indices either between strings or between strings and external bosons. The indices contraction between Weil strings can be easily performed by means of Fiertz identities (s_i represents any sequence of gamma matrices and "hat" momenta):

$$\{\bar{\chi}_1[s_1]\gamma_\mu^{\pm}[s_2]\chi_2\}\{\bar{\chi}_3[s_3]\gamma_\nu^{\mp}[s_4]\chi_4\}g^{\mu\nu} = 2\{\bar{\chi}_1[s_1][s_4]\chi_4\}\{\bar{\chi}_3[s_3][s_2]\chi_2\}$$

$$\{\bar{\chi}_1[s_1]\gamma_\mu^{\pm}[s_2]\chi_2\}\{\bar{\chi}_3[s_3]\gamma_\nu^{\pm}[s_4]\chi_4\}g^{\mu\nu} =$$

$$2\{\bar{\chi}_1[s_1][s_2]\chi_2\}\{\bar{\chi}_3[s_3][s_4]\chi_4\} - 2\{\bar{\chi}_1[s_1][s_4]\chi_4\}\{\bar{\chi}_3[s_3][s_2]\chi_2\}$$

Using explicit reduction of Dirac strings to Weil ones, Fiertz identities and "hat" decomposition, one can reduce any Feynman amplitude to a product of bi-spinor "building blocks" of the form $[\bar{\chi}_1^i \chi_2^j]$ that can be easily calculated numerically or analytically.

Additional simplifications can be obtained if one uses some relations among spinor products. A lot of different relations can be obtained that help in complicated cases, but the most useful ones that should be used practically in any calculations are:

$$[\bar{\chi}_i^+ \chi_j^-] = -[\bar{\chi}_j^+ \chi_i^-], \quad [\bar{\chi}_i^+ \chi_j^+] = [\bar{\chi}_j^- \chi_i^-], \quad [\bar{\chi}_k^i \chi_k^j] = 2E(k)\delta_{ij},$$
$$[\bar{\chi}_i^- \chi_j^-] \cdot [\bar{\chi}_k^- \chi_l^-] - [\bar{\chi}_i^- \chi_l^-] \cdot [\bar{\chi}_k^- \chi_j^-] = [\bar{\chi}_l^+ \chi_j^-] \cdot [\bar{\chi}_i^- \chi_k^-]$$

Making use the above technique, all amplitudes for the process $e^+e^- \to 4$f can be reduced to an extremely short form that is quite convenient for numerical calculations.

This technique can be used also for massive particles if the ratio m^2/E^2 is sufficiently small. In this case all "non-natural" spiralities in u/v spinors will be suppressed by this factor. At LEP II energies the relative difference between the exact and quasi-massless techniques is less than 10^{-4}, and it can be at most as large as 10^{-3} for processes that involve b-quarks.

2.2 TGC parameterization

2.2.1 Charged couplings

The most general phenomenological parameterization of the effective Lagrangian can be found in [6, 7, 8] (below the subscript V denotes γ or Z until stated definitely):

$$\begin{aligned} i\mathcal{L}^{VWW} \;=\; & g_{VWW}\Big[g_1^V V^\mu (W^-_{\mu\nu}W^{+\nu} - W^+_{\mu\nu}W^{-\nu}) + \kappa_V W^+_\mu W^-_\nu V^{\mu\nu} + \\ & \frac{\lambda_V}{m_W^2} V^{\mu\nu} W^{+\rho}_\nu W^-_{\rho\mu} + i g_5^V \epsilon_{\mu\nu\rho\sigma}\left((\partial^\rho W^{-\mu})W^{+\nu} - W^{-\mu}(\partial^\rho W^{+\nu})\right) V^\sigma + \\ & i g_4^V W^-_\mu W^+_\nu (\partial^\mu V^\nu + \partial^\nu V^\mu) - \frac{\tilde{\kappa}_V}{2} W^-_\mu W^+_\nu \epsilon^{\mu\nu\rho\sigma} V_{\rho\sigma} - \frac{\tilde{\lambda}_V}{2m_W^2} W^-_{\rho\mu} W^{+\mu}_\nu \epsilon^{\nu\rho\alpha\beta} V_{\alpha\beta}\Big], \end{aligned} \tag{1}$$

where $W_{\mu\nu} = \partial_\mu W_\nu - \partial_\nu W_\mu$ and $V_{\mu\nu} = \partial_\mu V_\nu - \partial_\nu V_\mu$. The convention of Ref.[9] is adopted, where $g_{\gamma WW} = -e$ and $g_{ZWW} = e\cot\Theta_W$.

Within the SM, at tree level, the couplings are given by $g_1^\gamma = g_1^Z = \kappa_\gamma = \kappa_Z = 1$, with all other couplings in (1) vanishing. For on-shell photons, electromagnetic gauge invariance fixes the values $g_1^\gamma = g_5^\gamma = 1$. The terms proportional to g_1^V, κ_V and λ_V conserve C and P separately, while g_5^V violates C and P but conserves CP. Other terms in (1) are connected with a possible CP violation in the bosonic sector.

It is well known (see, for example, [10]) that C and P conserving terms in $\mathcal{L}^{\gamma WW}$ correspond to the lowest order terms in a multipole expansion of the $W-\gamma$ interactions. The charge Q_W, the magnetic dipole moment μ_W and the electric quadrupole moment q_W are defined as:

$$Q_W = e g_1^\gamma, \qquad \mu_W = \frac{e}{2m_W}(g_1^\gamma + \kappa_\gamma + \lambda_\gamma), \qquad q^W = -\frac{e}{m_W^2}(\kappa_\gamma - \lambda_\gamma). \tag{2}$$

The general form of the VWW vertex is considered, taking all the external particles *off* their mass-shell. Expressing anomalous couplings in terms of the deviations from the expected SM values one obtains:

$$\Delta g_1 = (g_1^Z - 1) = \tan\theta_W \delta_Z, \quad g_1^\gamma = 1$$

$$\Delta\kappa_\gamma = \kappa_\gamma - 1 = x_\gamma, \quad \Delta\kappa_Z = \kappa_Z - 1 = \tan\theta_W(x_Z + \delta_Z), \tag{3}$$
$$\lambda_\gamma = y_\gamma, \quad \lambda_Z = \tan\theta_W y_Z,$$

In the above notation the explicit VWW vertex takes the form (neutral/charged bosons are considered to be incoming/outgoing):

$$\begin{aligned}
\Gamma^{VWW}_{\mu\alpha\beta} &= g_{VWW}\left\{\left(1 + \Delta g_1 + \frac{\lambda_V}{2m_W^2}p^2\right) g_{\alpha\beta}(q-\bar{q})_\mu - \frac{\lambda_V}{m_W^2}p_\alpha p_\beta (q-\bar{q})_\mu + \right. \\
&(2 + \Delta\kappa_V + \Delta g_1)(g_{\mu\beta}p_\alpha - g_{\mu\alpha}p_\beta) + \frac{\lambda_V}{m_W^2}(g_{\mu\beta}p_\alpha q^2 - g_{\mu\alpha}p_\beta \bar{q}^2) + \\
&\left(1 + \Delta g_1 + \frac{\lambda_V}{m_W^2}p\cdot\bar{q}\right) g_{\mu\alpha}\bar{q}_\beta - \left(1 + \Delta g_1 + \frac{\lambda_V}{m_W^2}p\cdot q\right) g_{\mu\beta}q_\alpha + \\
&\frac{\lambda_V}{m_W^2}\left[g_{\alpha\beta}p_\mu\frac{\bar{q}^2 - q^2}{2} - p_\alpha\bar{q}_\mu\bar{q}_\beta + p_\beta q_\alpha q_\mu\right] + ig_5\epsilon_{\alpha\beta\mu\gamma}(q-\bar{q})^\gamma - \\
&ig_4(g_{\mu\beta}p_\alpha + g_{\mu\alpha}p_\beta) - \tilde{\kappa}\epsilon_{\alpha\beta\mu\gamma}p^\gamma + \\
&\left.\frac{\tilde{\lambda}}{m_W^2}p^\gamma\left(\epsilon_{\beta\rho\gamma\mu}q^\rho\bar{q}_\alpha + \epsilon_{\rho\alpha\gamma\mu}q_\beta\bar{q}^\rho - \epsilon_{\beta\alpha\gamma\mu}q\cdot\bar{q}\right)\right\}
\end{aligned} \tag{4}$$

where $g_{\gamma WW} = -ie$, $g_{ZWW} = ie\cot\Theta_W$, and $\epsilon_{1234} = -1$.

The general three-boson vertex with arbitrary values of the couplings is not locally gauge invariant and this fact leads to some difficulties, such as violation of tree unitarity in scattering processes and bad divergences in loop corrections. Some symmetry requirements, however, can be imposed on the general Lagrangian to ensure the absence of the most serious divergences [11]. These requirements appear as relations among *a priori* arbitrary anomalous couplings, and reduce the number of them.

Turning from the phenomenological Lagrangian, one can consider $SU(2)\otimes U(1)$ operators of high dimensions to reproduce all couplings in (1). Depending on the new physics dynamics, such operators could be generated at the mass scale Λ, with a strength which is generally suppressed by a factors like $(m_W/\Lambda)^{d-4}$ [12]. As the reduced three-boson vertex contains only $d = 4, 6$ operators, it should be stressed that it represents the low-energy limit of the general Lagrangian expansion over relative operator strengths if only the leading terms are kept. This method reproduces the so-called "Linear Realization" of the gauge invariant three-boson vertex parameterization. The Lagrangian that generates these couplings is as follows:

$$i\mathcal{L} = g'\frac{\alpha_{B\phi}}{m_W^2}\mathcal{O}_{B\phi} + g\frac{\alpha W\phi}{m_W^2}\mathcal{O}_{W\phi} + g\frac{\alpha_W}{m_W^2}\mathcal{O}_W + \frac{gg'}{2}\frac{\tilde{\alpha}_{BW}}{m_W^2}\tilde{\mathcal{O}}_{BW} + g\frac{\tilde{\alpha}_W}{m_W^2}\tilde{\mathcal{O}}_W \tag{5}$$

where g, g' are $SU(2)_L$ and $U(1)_Y$ couplings respectively.

Replacing the Higgs doublet field by its vacuum expectation value, one obtains:

$$\Delta g_1 = \frac{\alpha_{W\phi}}{c_W^2}, \quad \Delta\kappa_\gamma = -\frac{c_W^2}{s_W^2}(\Delta\kappa_Z - \Delta g_1) = \alpha_{W\phi} + \alpha_{B\phi}, \quad \lambda_\gamma = \lambda_Z = \alpha_W \tag{6}$$
$$\tilde{\kappa}_\gamma = -\cot^2\theta_W\tilde{\kappa}_Z = \tilde{\alpha}_{BW}, \quad \tilde{\lambda}_\gamma = \tilde{\lambda}_Z = \tilde{\alpha}_W.$$

2.2.2 Neutral couplings

In addition to charged couplings $WW\gamma$ and WWZ, neutral ones can also be defined. The well known representation by Hagiwara et al.[7] gives these couplings, fixing selected bosons at mass-shell. Generally speaking, this representation cannot be satisfactory for general amplitudes due to the finite width of Z-boson. Adequate treatment of neutral couplings requires careful operator analysis and is not performed yet. `DELTGC` adopts the simplest parametrization presented in [7]. Particular vertices have the form:

$$\Gamma^{VZZ}_{\mu\alpha\beta} = i\frac{s-m_V^2}{m_Z^2}\left\{f_4^V(p_\alpha g_{\mu\beta}+p_\beta g_{\mu\alpha})+f_5^V\epsilon_{\mu\alpha\beta\rho}(q-\bar{q})^\rho\right\}, \tag{7}$$

and

$$\Gamma^{VZ\gamma}_{\mu\alpha\beta} = i\frac{s-m_V^2}{m_Z^2} \quad \left\{h_1^V(\bar{q}_\mu g_{\alpha\beta}-\bar{q}_\alpha g_{\mu\beta})+\frac{h_2^V}{m_Z^2}p_\alpha(p\cdot\bar{q}g_{\mu\beta}-\bar{q}_\mu p_\beta)\right.$$
$$\left. h_3^V\epsilon_{\mu\alpha\beta\rho}\bar{q}^\rho+\frac{h_4^V}{m_Z^2}p_\alpha\epsilon_{\mu\beta\rho\sigma}p^\rho\bar{q}^\sigma\right\} \tag{8}$$

In the above expressions the indices and momentum flows are the same as for charged couplings.

2.3 Gauge Cancellations

Some processes from the $e^+e^- \to$ 4f set contain singular diagrams that show bad t-channel divergences. Nevertheless, the final result after integration over the whole phase-space should be finite. This means that gauge cancellation should occur between singular diagrams, leading to an integrable matrix element.

To handle the gauge cancellations, two aspects should be taken into account.

2.3.1 Input Parameters

Calculating the cross section for the processes $e^+e^- \to$ 4f one deals with parameters such as: α_e, M_Z, Γ_Z, M_W, Γ_W, $\sin\Theta_W$ (the QCD related parameter α_s and external fermion masses are omitted because they do not affect the gauge behaviour of the calculations).

The value of the electroweak coupling constant α_{em} can also be separated from the problem of gauge cancellation because it appears as a common factor for all diagrams. It must be noticed that one needs two different values of this constant: $\alpha_{em}(0)$ and $\alpha_{em}(2M_W)$. The first one is used for the calculation of the initial state radiation, while the other is the electromagnetic coupling constant at the LEP-II energy scale.

The remaining set of parameters is over-defined, as any SM process can be described by means of *four* independent parameters (α_{em} is already removed from this set). It is well known that at tree-level the gauge cancellations are exact and the relations among the parameters can be used to reduce the number of them.

One can choose the well-measured set of independent parameters such as G_F, M_Z, M_W, and Γ_Z, with Γ_W and $\sin\Theta_W$ calculated from:

$$\Gamma_W = \frac{3G_FM_W^3}{\sqrt{8}\pi} \quad \text{and} \quad \sin^2\Theta_W = \frac{\pi\alpha(2M_W)}{\sqrt{2}G_FM_W^2}\,.$$

2.3.2 Gauge Boson Propagator Treatment

One may consider this problem simply by calculating all the diagrams in the "zero-width" approximation, which obviously ensures gauge invariance and good high-energy behaviour of the amplitudes. On the other hand, these amplitudes result in quite hard problems during the phase-space integration, due to the explicit poles for the time-like gauge-boson momenta. Some prescriptions were developed to soften this problem [13, 14]. The recipe consists in the multiplication of the zero-width amplitudes by the factor $(q^2 - M^2)/(q^2 - M^2 + iM\Gamma)$. This factor cancels the contribution of the pole to the amplitude, at the price of mistreating the non-resonant parts.

Another way to solve the problem is to use the "fixed-width" propagators of the form $1/(q^2 - M^2 + iM\Gamma)$ both for time- and space-like momenta. This procedure ensures $U(1)_{em}$ current conservation, but it has no physical motivation because the propagator does not develop an imaginary part for space-like momenta ($q^2 < 0$) in the perturbative theory. Additional problems in the "fixed-width" approximation will occur when one deals with W_L final state re-scattering, for example in $e^+e^- \to 6$ fermions processes. The "fixed-width" propagators will explicitly violate $SU(2) \otimes U(1)$ invariance and develop bad high-energy behaviour.

The most theoretically motivated approach is based on the consideration of the fermion-loop corrections to the three-boson vertex that compensate the imaginary part in the Ward identities. The systematic study of this method can be found, for example, in [15, 16]. The resulting prescriptions are described below:

- the time-like gauge boson propagators are treated with s-dependent widths;
- the space-like propagators are taken in the "zero-width" approximation;
- the three-boson vertex is multiplied by a factor (in the simplified approach [15])

$$1 + i\frac{\Gamma_W \bar{q}^2}{M_W(\bar{q}^2 - q^2)}\,. \tag{9}$$

The above prescriptions ensure electromagnetic current conservation and gauge cancellations.

A much more elaborated technique is developed in [16].

Another approach was developed in [8] where the method of applying correction factors to the gauge boson propagators was proposed. This method assumes the constant width approximation for both time- and space-like propagators with substitutions:

$$\tilde{M}_V = \frac{M_V}{\sqrt{1+\gamma_V^2}}, \quad \tilde{\Gamma}_V = \frac{\Gamma_V}{\sqrt{1+\gamma_V^2}},$$

where $\gamma_V = \Gamma_V/M_V$.

This method is extremely simple and gives quite the same numerical results as more theoretically elaborated ones.

2.4 Z'-related coefficients

Despite its tremendous success in describing the experimental data within the range of energies available today, the SM based on the gauge symmetry $SU(3)_C \otimes SU(2)_L \otimes U(1)_Y$

is widely believed not to be the ultimate truth. Different theoretical arguments can be presented in the favour of SM extension. Well known among them are the reduction of the parameters introduced by hand and possible unification of the electro-weak and strong interactions.

Different models based on the extension of the symmetry group have been proposed (for a review see, for example [17]).

All these models predict the existence of (at least) one additional neutral gauge boson Z'. The Z' search strategy can be based on different approaches. One of them is the direct search for a new physics signal in $e^+e^- \to f\bar{f}$ at current or higher energies: this can manifest itself as a resonance peak or as a strong interference effect on the SM cross-section.

Another approach is connected with W production, where it can be shown (see, for example [18]) that the effect of an additional gauge boson can be absorbed in a modification of the SM triple gauge couplings g_Z and g_γ. The resulting expressions are as follows:

$$\begin{aligned} \Delta g_\gamma &= g_{Z'WW}\frac{q^2}{M_{Z'}^2 - q^2} g_{Vl}(\xi_{Vl} - \xi_{Al}), \\ \Delta g_Z &= -g_{Z'WW}\frac{q^2 - M_Z^2}{M_{Z'}^2 - q^2} g_{Vl}\xi_{Al}, \end{aligned}$$

where the functions ξ_{Al} and ξ_{Vl} absorb $Z' - l$ couplings. DELTGC calculates the convolution of the amplitudes with the q^2 and mass dependent form-factors:

$$\begin{aligned} F_\gamma &= \frac{c_w}{s_w}\frac{q^2}{M_{Z'}^2 - q^2}\frac{M_Z}{M_{Z'}}, \\ F_Z &= \frac{q^2 - M_Z^2}{M_{Z'}^2 - q^2}\frac{M_Z}{M_{Z'}}. \end{aligned}$$

By default the mass of $Z' = 1000$ GeV but it can be changed by the user.

3 The Program Description

3.1 Program Features

DELTGC performs an adaptive integration, with optional subsequent event generation, for processes of the type $e^+e^- \to$ 4f with special emphasis on anomalous triple gauge couplings. The program calculates the cross-section for any particular set of the TGC, as well as the coefficients in the cross-section decomposition in terms of the TGC. All the coefficients can be obtained in one single program run. The program supports two different TGC representation schemes: one is the standard scheme of g, κ, λ coefficients (see, for example, [7]), and the other is the commonly used α-representation.

Special attention was devoted to the gauge cancellation of singular diagrams that ensures a high stability of the calculations in a singular region. For example, the range of the integration with respect to the outgoing electron can be extended to the full angular interval with fast convergence of the integral. The gauge boson propagators have been treated by the modified width scheme by Berends [8].

The program implements complex cuts on the final state fermion momenta. Some cuts can be implemented through the external interface file, and non-standard situations can be treated with an optional user-supplied routine.

DELTGC supports two methods of amplitude calculation. One is based on the quasi-massless treatment of the Dirac spinors. This method assumes finite masses for the fermions while all right/left spiralities for u/v spinors are suppressed due to the factor m^2/E^2 arising in the Dirac spinor expansion in terms of the inverse energy. It gives an excellent agreement with the calculations of other programs and is very fast. This method is very convenient for calculations at high energies.

Another method is based on exact calculation of the amplitudes with possible arbitrary mass assignment to the fermions. It is much slower, but is suitable for calculations in a wide energy interval, including quite low $\sqrt{s}$.

DELTGC v.1.0 is installed in the fast mode and cannot be switched to the extended amplitudes calculations by the user.

3.2 Program Input

During initialization, the program reads the input file from the FORTRAN logical unit 19. If the user does not assign any file on this unit, the standard file `DELTGC.INP` from the directory where DELTGC starts is opened.

An example of the input is given in Appendix A. In this file the user should define the process under study (the processes are listed in Appendix B), ISR, QCD and Coulomb correction switches, cuts, and other parameters. The details are given in Appendices A and B.

3.3 The Program Layout

3.3.1 Initialization

The user-called initialization routine

```
subroutine Init_Model (iret)
```

reads the external user-assigned file or the file with the default name `DELTGC.INP`. The parameter `INTEGER iret` returns a non-zero value if errors have been detected during initialization.

This routine initializes all suitable constants and flags. It also checks the consistency of the problem defined, and prepares special flags and internal parameters for the subsequent integration of the matrix element.

Some internal parameters can be changed **before** the main initialization. This can be done by calling

```
subroutine TGC_Parameters (NAME, Value)

          character*4 NAME
          real*8      Value
```

The meanings and default values of the parameters that can be changed by the user are:

NAME	Meaning	Default	NAME	Meaning	Default
ME	m_e GeV	0.511d-03	MMU	m_μ GeV	0.106
MTAU	m_τ GeV	1.777	MZ	m_Z GeV	91.188
GZ	Γ_Z GeV	2.03	MW	m_W GeV	80.23
MUQ	m_u GeV	0.05	MDQ	m_d GeV	0.05
MCQ	m_c GeV	1.5	MSQ	m_s GeV	0.3
MTQ	m_t GeV	170.	MBQ	m_b GeV	5.0
ALPE	α_{em}^{-1}	128.07	ALPS	α_{QCD}	0.12
MZP	$M_{Z'}$ GeV	1000.	VUD	V_{ud}	0.9751
VUS	V_{us}	0.2225	VUB	V_{ub}	0.0035
VCD	V_{cd}	0.221	VCS	V_{cs}	0.9743
VCB	V_{cb}	0.041	VTD	V_{td}	0.009
VTC	V_{tc}	0.04	VTB	V_{tb}	0.9991

Please note that α_{QCD} does not affect the value of the final state QCD corrections because the running coupling constant is used for correction calculations that depend on the W mass: α_{QCD} is used for the calculation of diagrams with time like hard gluons.

CKM mixing constants accept the specified values only if the CKM flag in the file `DELTGC.INP` is set to a non-zero value (see Appendix A). Zero value for this flag will force `DELTGC` to set all diagonal mixing parameters to 1 and non-diagonal ones to 0.

3.3.2 Integration and Distributions

The routine

```
subroutine Get_CS (sig,err,chi2)
```

performs an adaptive integration of the matrix element using a user-defined number of iterations and function calls per iteration.

The data flow and the routines interdependence during integration is given in Figure.1.

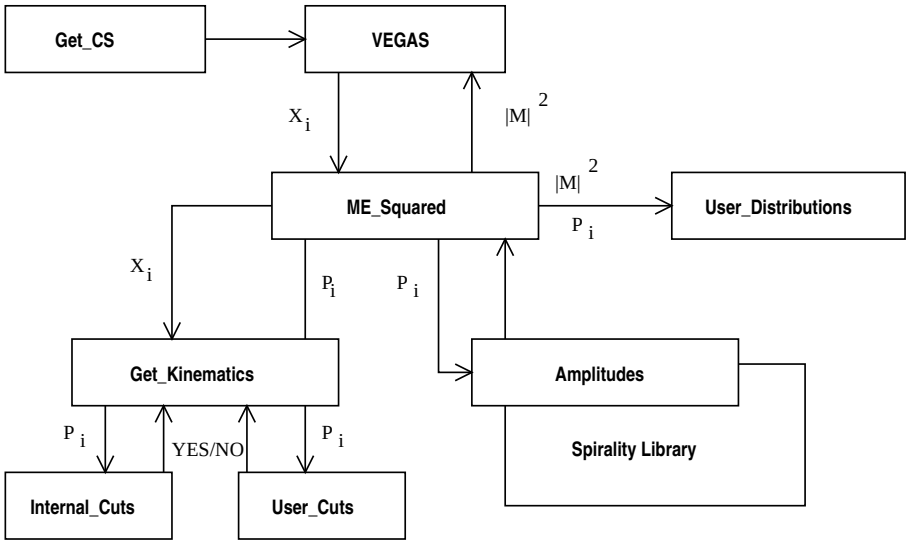

Figure.1. Data flow during integration.

`Get_CS` is in fact a dummy routine that calls `VEGAS` to perform adaptive integration. The algorithm of Lepage [2] for multidimensional numerical integration using importance- and stratified samplings is used. The number of integration variables is either 7 (no ISR corrections) or 9 (ISR corrections on).

`VEGAS` is designed in such a way that the integration is always performed over the unit hypercube, so the relevant transformation from `VEGAS` variables to the physical momenta should be performed. Moreover, the meaning of the variables depends strongly on the process under study. The transformation from `VEGAS` variables X_i to momenta P_i is designed in such a way that non-regular matrix element behaviour is concentrated along separate variables X_i. This is quite an important point because any adaptive algorithm gives an advantage only when non-regularities in the integrand can be separated along different coordinate axes.

This is a non-trivial task and the routine `Get_Kinematics` performs the transformation from `VEGAS` variables to the physical ones taking into account the resonance structure, t- and u-channel singularities of the matrix element. Obviously, in complex cases it is impossible to separate the singularities completely and the convergence of the integration

depends strongly on the process. Nevertheless, the usual time to integrate the amplitude by adaptive methods is 2-3 orders of magnitude less than for non-adaptive methods.

After creation of the external particle momenta, `Get_Kinematics` calls two logical functions: `Internal_Cuts` that checks the consistency of the momenta generated and the cuts defined in `DELTGC.INP`, and `User_Kinematics` that treats non-standard cuts. If both functions returns logical `.TRUE.` the resulting momenta are passed to the routine that calculates amplitudes.

Finally, the user-defined routine `User_Distributions` is called to provide the user with the momenta and the matrix element squared to perform actions on the accumulating distributions. An example of the program can be found in Appendix C. Please, note, that the `WGT` parameter passed to the user is constructed in such a way that it can be used directly for the distributions accumulation, and is measured in `[pb]`.

The user-defined routine `User_Distributions` has the parameters:

```
subroutine User_Distributions (p,wgt,imode)

real*8 p(4,6)       ! particles momenta in the order:
                    ! e-, e+ and other momenta in the order
                    ! of process table

real*8 wgt(327)     ! weights for the current event

integer imode       ! switch parameter
```

The switch parameter `imode` can accept two values: `0` and `1`. The value `0` corresponds to the call of the `User_Distributions` from the main iterations loop.

`DELTGC` provides another possibility to obtain the distributions of the external particles kinematical functions. It consists in a separate call to the routine

```
call Get_Distributions (nevent)
```

that should be done **after** the main integration loop. In this case the parameter `imode=1`.

The parameter `nevent` informs `DELTGC` how many events should be generated during the run. Please note that the efficiency of the generation algorithm varies from one process to another depending on the complexity of the underlying kinematics and is never 100%. The typical value varies from 30 to 60%.

Only events that were effectively generated are passed to the user routines. The final number of events passed to the subroutine `User_Distributions` will **always** be less than `nevent`. So, the recommended value for the parameter `nevent` is $2-3\cdot 10^5$. The resulting number, around 10^5 events, is sufficient to have meaningful distributions in all cases.

Some explanations should be given concerning weights passed while creating the distributions. The array `wgt(327)` contains the full TGC-decomposition of the matrix element for both $\kappa-\lambda$ and α representations, as well as Z'-related coefficients.

The first 300 elements of this array contain the upper right triangle of the coefficient matrix for the $\kappa - \gamma$ representation, together with the neutral TGC parameters, written by rows. The order of the TGC parameters is as follows:

$$SM,\ g_Z,\ k_Z,\ k_\gamma,\ \lambda_Z,\ \lambda_\gamma,\ g_4,\ g_5,\ \tilde{\kappa}_Z,\ \tilde{\kappa}_\gamma,\ \tilde{\lambda}_Z,\ \tilde{\lambda}_\gamma,\ f_4^Z,\ f_4^\gamma,\ f_5^Z,\ f_5^\gamma,\ h_1^Z,\ h_1^\gamma,\ h_2^Z,\ h_2^\gamma,\ h_3^Z,\ h_3^\gamma,$$

$$h_4^Z,\ h_4^\gamma.$$

So, the term `wgt(1)` gives the SM value, `wgt(2)` gives the interference term SM-g_Z for $g_Z = 1$, `wgt(25)` gives the term proportional to g_Z^2, and so on.

The elements from 301 to 306 give the Z'-related coefficients in the order:

$$SM,\ {g'}_Z,\ {g'}_\gamma.$$

The terms from 307 to 327 contain the upper right triangle for the α coefficients matrix in the order:

$$SM,\ \alpha_{W\phi},\ \alpha_{B\phi},\ \alpha_W,\ \tilde{\alpha}_{BW},\ \tilde{\alpha}_W.$$

Passed to `User_Distributions` array **always** contains the full matrix element decomposition regardless of the parameters defined in `DELTGC.INP`. This gives the possibility to have the "distribution decomposition" in terms of the TGC parameters.

The `VEGAS` routine creates an optimized grid using the user defined number of iterations and accumulates internal weights to be used during the optional subsequent event generation.

3.3.3 Access to the Results

If the fixed mode of the calculation has been chosen (see Appendix A), `Get_CS` returns the value of σ in `[pb]`, its error and χ^2 value. If the χ^2 value is much larger than 1, this indicates a bad convergence.

If the full table mode was chosen, the returned parameters have the Standard Model value for the process calculated and all other parameters can be obtained via a call to the routine

```
subroutine Get_Full_Table (i,j,sig,err,chi2)
```

that returns the coefficient σ_{ij} in the cross section decomposition:

$$\sigma(\vec{x}) = \sum_{i,j} \sigma_{ij} x_i x_j$$

where $x_i = 1$.

The order of TGC parameters is as follows:

- SM(0), $g_Z(1)$, $\kappa_Z(2)$, $\kappa_\gamma(3)$, $\lambda_Z(4)$, $\lambda_\gamma(5)$, $g_4(6)$, $g_5(7)$, $\tilde{\kappa}_Z(8)$, $\tilde{\kappa}_\gamma(9)$, $\tilde{\lambda}_Z(10)$, $\tilde{\lambda}_\gamma(11)$, $f_4^Z(12)$, $f_4^\gamma(13)$, $f_5^Z(14)$, $f_5^\gamma(15)$, $h_1^Z(16)$, $h_1^\gamma(17)$, $h_2^Z(18)$, $h_2^\gamma(19)$, $h_3^Z(20)$, $h_3^\gamma(21)$, $h_4^Z(22)$, and $h_4^\gamma(23)$ – for the standard representation,

- SM(0), $\alpha_{W\phi}(1)$, $\alpha_{B\phi}(2)$, $\alpha_{B\phi}(3)$, $\tilde{\alpha}_{BW}(4)$ and $\tilde{\alpha}_W(5)$ – for the α-representation.

So, the coefficient $1,1$ for the standard representation corresponds to the coefficient at g_Z^2 in the cross section decomposition and $0,1$ to that of the SM-g_Z interference term.

Setting one of the parameters `i` or `j` to a negative value will cause the retrieval of the Z'-related coefficients. The meaning of these parameters is as follows:

$$SM(0),\ {g'}_Z(1) \text{ and } {g'}_\gamma(2)$$

The user can check the quality of the integration using the errors and χ^2 returned, and can repeat the call to `Get_CS`. The integration procedure will then continue the grid optimization using all the information from the previous iterations.

To calculate the value of the cross section for a particular value of a TGC parameter the function

```
real*8 function Get_Current_Sig (itgc,vtgc)

integer itgc          ! Number of the TGC parameter (see p.16)
real*8  vtgc          ! TGC parameter value
```

can be used.

The routine

```
real*8 function Get_CS (sig,err,chi2)
```

described above returns a cross-section table that can be used for most calculations, but DELTGC provides the possibility to obtain the TGC dependences with much higher precision.

Routine

```
real*8 function Get_All_Dependencies (sig)

real*8 sig(327)       ! The cross-section decomposition
```

will perform the calculations with high precision. The coefficients are calculated by a parabolic interpolation of the cross-sections calculated at three different values of the particular TGC parameter. One value coincides with the SM and two others are chosen at $+v$ and $-v$, where v is a value supplied by the user in the file DELTGC.INP for the particular TGC parameter.

To save calculation time, not all TGC coefficients will be calculated. Only those with non-zero values in DELTGC.INP will be used. The value (v) supplied by the user for the particular TGC parameter should not be either too small (because it will decrease the final precision) or too large (in this case the integration can become unstable). The optimal value is typically in the interval $2-4$.

3.3.4 Matrix Element Access

To access the matrix element squared for any particular process and configuration of external momenta, the user should (after normal DELTGC initialization) call the routine

```
subroutine Get_Matrix_Element (ipro,p,sig)

integer ipro          ! The process label
real*8 p(3,4)         ! final particles momenta (WITHOUT ENERGY)
real*8 sig(327)       ! Matrix element square decomposition
```

The user can choose any process regardless of the value specified in `DELTGC.INP`. DELTGC will initialize the masses of the final state fermions, perform the transformation to the center-of-mass system, and calculate the cross-section for the given phase-space point. Note, please, that DELTGC **does not perform** any check of the consistency of the momenta supplied with the user-defined cuts, and will not call `User_Kinematics`.

From the supplied values of the final-state particle momenta, DELTGC will define the fractions of the colliding energy carried off by the ISR photons from the e^- and e^+, calculate the relevant weight of the ISR correction function, and correct the resulting values of the cross-section decomposition by the resulting factor.

The returned values of the decomposition (`sig`) can be used to calculate the value of the matrix element squared for the particular value of TGC. The function

```
real*8 function Get_Current_Ratio (sig,itgc,vtgc)

real*8 sig(327)       ! The matrix element decomposition
integer itgc          ! Number of the TGC parameter (see p.16)
real*8  vtgc          ! TGC parameter value
```

will return the value of the ratio of the matrix element squared calculated for the specified value of the TGC to the SM value.

Appendix A.

An example of the input file `DELTGC.INP`

```
25                              ! IPROC
1                               ! Fermion mass flag
172.                            ! SQS
1                               ! ISR switch
1                               ! QCD corrections switch
1                               ! Coulomb corrections switch
0                               ! CKM flag
1                               ! external mass cuts
    5.   5.   5.   5.   5.   5.
  172. 172. 172. 172. 172. 172.
1                               ! External energy cuts
    3.   3.   3.   3.
  170. 170. 170. 170.
1                               ! Angular cuts
    0.   0.   0.   0.
  170. 180. 180. 180.
1                               ! Separation cuts
    0.   0.   0.   0.   0.   0.
  180. 180. 180. 180. 180. 180.
1                               ! PT cuts
    0.   0.   0.   0.
  170. 170. 170. 170.
1                               ! Acoplanarity cuts
    0.   0.   0.   0.   0.   0.
  180. 180. 180. 180. 180. 180.
20                              ! Number of iterations
20000                           ! Number of function eval./iteration
1                               ! Select TGC representation
1                               ! Select fixed/table mode
0. 0. 0. 0. 0.                  ! TGC values for charged couplings
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.   ! TGC values for neutral couplings
```

IPROC The value of the process to be calculated. See Appendix B. for the list of the processes.

Fermion mass flag Normally, should be set to **1**. Turning it to **0** the user will set all the external fermion masses to zero. This can lead to hard integration behaviour and **should be avoided by the user**.

SQS The value of $\sqrt{s}$ in GeV.

ISR switch The corrections for soft photon bremsstrahlung in the initial state switched on/off (2/1/0). The value **1** corresponds to collinear ISR in the structure function approximation, and **2** corresponds to the final P_T ISR. In the last case during **event generation** DELTGC will return into `common /sa_phot/ phot(3,2)` the REAL*8 momenta of the photons generated.

QCD switch The corrections from soft gluon radiation from the final quarks switched on/off (1/0).

Coulomb switch The Coulomb corrections switched on/off (1/0).

CKM switch Normally should be set to 0. In this case no CKM mixing processes will be calculated and diagonal mixing parameters will be set to 1. Turning it to 1 the user restores the pre-defined CKM mixing parameters and obtains access to the processes with CKM mixing.

Mass cuts This switch indicates that the program should perform internal mass cuts. The next two strings give the lower and upper mass cuts respectively in the order: 3+4, 3+5, 3+6, 4+5, 4+6, 5+6. The values are given in GeV.

Energy cuts This switch indicates that the program should perform internal cuts on the energies of the external fermions. The next two strings give the lower and upper values for the energies respectively in the order: 3, 4, 5, 6. The values are given in GeV.

Angular cuts This switch indicates that the program should perform internal cuts on the angle of the external fermion with respect to the z-axis. The value **1** corresponds to degrees, while the value **2** corresponds to the cuts in radians. **Note please!** In DELTGC the positive direction of the z-axis is defined along the incoming **positron** momentum. The next two strings give the lower and upper angular cuts respectively in the same order as the energy cuts.

Separation cuts This switch indicates that the program should perform internal cuts on the opening angle between two external fermions. The value of the switch is the same as for angular cuts, the order of fermion pairs is the same as for mass cuts.

PT cuts This switch indicates that the program should perform internal cuts on the P_t values of the external fermions. The order of values is the same as for Energy cuts.

Acoplanarity cuts This switch indicates that the program should perform internal cuts on the acoplanarity value for two external fermions. The value of the switch is the same as for angular cuts, the order of fermion pairs is the same as for mass cuts.

Number of iterations Gives the number of iterations to be performed by DELTGC to set the optimized grid in the integration variables.

Number of func. eval. Number of amplitude calls during every iteration.

TGC representation Selects between the g-κ-λ (0) and the α (1) representation.

Calculation mode Selects between fixed mode (1), when DELTGC calculates the cross section for the particular set of the TGC values, and table mode (0), when the whole table of the cross-section decomposition will be created. For the last case the values of the TGC parameters supplied by the user are not taken into account.

Charged TGC values Supply here the values of TGC parameters for **fixed** mode.

Neutral TGC values As above.

Important notes

1. The lines with the values of kinematical cuts are read by the initialization routine regardless of the values of the cut switches.

2. The generation of the events can be performed for any particular set of TGC parameters. If the user is going to perform event generation, the values of the TGC parameters should be set.

Appendix B.

The table below gives the process numbers as they should be referred to in DELTGC.INP

ID	Final state	ID	Final state	ID	Final state	ID	Final state
1	$e^-\bar{\nu}_e\nu_e e^+$	2	$e^-\bar{\nu}_e\nu_\mu\mu^+$	3	$e^-\bar{\nu}_e\nu_\tau\tau^+$	4	$\mu^-\bar{\nu}_\mu\nu_\mu\mu^+$
5	$\tau^-\bar{\nu}_\tau\nu_\tau\tau^+$	6	$\mu^-\bar{\nu}_\mu\nu_\tau\tau^+$	7	$e^-e^+e^-e^+$	8	$e^-e^+\mu^-\mu^+$
9	$e^-e^+\tau^-\tau^+$	10	$\mu^-\mu^+\mu^-\mu^+$	11	$\tau^-\tau^+\tau^-\tau^+$	12	$\mu^-\mu^+\tau^-\tau^+$
13	$e^-\bar{\nu}_\mu\nu_\mu e^+$	14	$e^-\bar{\nu}_\tau\nu_\tau e^+$	15	$\mu^-\bar{\nu}_e\nu_e\mu^+$	16	$\tau^-\bar{\nu}_e\nu_e\tau^+$
17	$\mu^-\bar{\nu}_\tau\nu_\tau\mu^+$	18	$\tau^-\bar{\nu}_\mu\nu_\mu\tau^+$	19	$\nu_e\bar{\nu}_e\nu_e\bar{\nu}_e$	20	$\nu_e\bar{\nu}_e\nu_\mu\bar{\nu}_\mu$
21	$\nu_e\bar{\nu}_e\nu_\tau\bar{\nu}_\tau$	22	$\nu_\mu\bar{\nu}_\mu\nu_\mu\bar{\nu}_\mu$	23	$\nu_\tau\bar{\nu}_\tau\nu_\tau\bar{\nu}_\tau$	24	$\nu_\mu\bar{\nu}_\mu\nu_\tau\bar{\nu}_\tau$
25	$e^-\bar{\nu}_e u\bar{d}$	26	$e^-\bar{\nu}_e c\bar{s}$	27	$\mu^-\bar{\nu}_\mu u\bar{d}$	28	$\mu^-\bar{\nu}_\mu c\bar{s}$
29	$\tau^-\bar{\nu}_\tau u\bar{d}$	30	$\tau^-\bar{\nu}_\tau c\bar{s}$	31	$e^-e^+u\bar{u}$	32	$e^-e^+c\bar{c}$
33	$e^-e^+d\bar{d}$	34	$e^-e^+s\bar{s}$	35	$e^-e^+b\bar{b}$	36	$\mu^-\mu^+u\bar{u}$
37	$\mu^-\mu^+c\bar{c}$	38	$\tau^-\tau^+u\bar{u}$	39	$\tau^-\tau^+c\bar{c}$	40	$\mu^-\mu^+d\bar{d}$
41	$\mu^-\mu^+s\bar{s}$	42	$\mu^-\mu^+b\bar{b}$	43	$\tau^-\tau^+d\bar{d}$	44	$\tau^-\tau^+s\bar{s}$
45	$\tau^-\tau^+b\bar{b}$	46	$u\bar{u}\nu_e\bar{\nu}_e$	47	$c\bar{c}\nu_e\bar{\nu}_e$	48	$d\bar{d}\nu_e\bar{\nu}_e$
49	$s\bar{s}\nu_e\bar{\nu}_e$	50	$b\bar{b}\nu_e\bar{\nu}_e$	51	$u\bar{u}\nu_\mu\bar{\nu}_\mu$	52	$c\bar{c}\nu_\mu\bar{\nu}_\mu$
53	$u\bar{u}\nu_\tau\bar{\nu}_\tau$	54	$c\bar{c}\nu_\tau\bar{\nu}_\tau$	55	$d\bar{d}\nu_\mu\bar{\nu}_\mu$	56	$s\bar{s}\nu_\mu\bar{\nu}_\mu$
57	$b\bar{b}\nu_\mu\bar{\nu}_\mu$	58	$d\bar{d}\nu_\tau\bar{\nu}_\tau$	59	$s\bar{s}\nu_\tau\bar{\nu}_\tau$	60	$b\bar{b}\nu_\tau\bar{\nu}_\tau$
61	$u\bar{d}d\bar{u}$	62	$c\bar{s}s\bar{c}$	63	$u\bar{d}s\bar{c}$	64	$u\bar{u}u\bar{u}$
65	$c\bar{c}c\bar{c}$	66	$d\bar{d}d\bar{d}$	67	$s\bar{s}s\bar{s}$	68	$b\bar{b}b\bar{b}$
69	$u\bar{u}c\bar{c}$	70	$u\bar{u}s\bar{s}$	71	$u\bar{u}b\bar{b}$	72	$c\bar{c}d\bar{d}$
73	$c\bar{c}b\bar{b}$	74	$d\bar{d}s\bar{s}$	75	$d\bar{d}b\bar{b}$	76	$s\bar{s}b\bar{b}$
77	$e^-\bar{\nu}_e u\bar{s}$	78	$e^-\bar{\nu}_e u\bar{b}$	79	$e^-\bar{\nu}_e c\bar{d}$	80	$e^-\bar{\nu}_e c\bar{b}$
81	$\mu^-\bar{\nu}_\mu u\bar{s}$	82	$\mu^-\bar{\nu}_\mu u\bar{b}$	83	$\mu^-\bar{\nu}_\mu c\bar{d}$	84	$\mu^-\bar{\nu}_\mu c\bar{b}$
85	$\tau^-\bar{\tau}_\mu u\bar{s}$	86	$\tau^-\bar{\tau}_\mu u\bar{b}$	87	$\tau^-\bar{\tau}_\mu c\bar{d}$	88	$\tau^-\bar{\tau}_\mu c\bar{b}$
89	$u\bar{u}d\bar{s}$	90	$u\bar{u}d\bar{b}$	91	$c\bar{u}d\bar{d}$	92	$c\bar{u}d\bar{b}$
93	$u\bar{c}d\bar{s}$	94	$u\bar{c}d\bar{b}$	95	$c\bar{c}d\bar{b}$	96	$u\bar{u}s\bar{b}$
97	$c\bar{u}s\bar{b}$	98	$c\bar{u}b\bar{b}$				

Appendix C.

Examples of the programs are given below:

```
c
c This program calculates the particular process and saves the full table
c of the cross-section decomposition coefficients into external file.
c
c The distribution over cos(theta) of the third particle also constructed
c
      implicit real*8 (a-h,o-z)
      common /pawc/ hmemor(10000)
      real*4 hmemor
c
c Initialize HBOOK area
c
      call hmemor (10000)
      call hbook1 (1,' Cos(Theta_3) ',100,-1.,1.,0.)
c
c Re-define some parameters
c
      call TGC_Parameters ('MW  ',80.40d0)         ! M_W
      call TGC_Parameters ('ALPE',130.d0)          ! 1./\alpha_e
c
c Initialize DELTGC
c
      call Init_Model (iret)
      if (iret.ne.0)             STOP
c
c All is OK. Calculate cross-section.
c
      call Get_CS (sig,err,chi2)
c
c Create some distributions
c
      call Get_Distributions (300000)
c
c Save the results in the external file
c
      open (unit=2,file='res.dat',status='new')
      do i1 = 0,5
         do i2 = 0,5
            call Get_Full_Table (i1,i2,sig,err,chi2)
            write (2,*) sig
         enddo
      enddo
```

```
c
c Save histogram
c
      call hrput (0,'res.hbk','N')
c
      stop
      end
c------------------------------
      logical function User_Kinematics (p)
c----------------------------------------
      implicit real*8 (a-h,o-z)
      real*8 p(4,6)
c
      User_Kinematics = .FALSE.
c                                    Define your cuts here
      if ( NOT true for your cuts ) return
c
      User_Kinematics = .TRUE.
c
      return
      end
c------------------------------
      subroutine User_Distributions (p,wgt,imode)
c-----------------------------------------
      implicit real*8 (a-h,o-z)
      real*8 p(4,6),wgt(327)
      real*4 cost
c
      if (imode.eq.0) return  ! We will use events in separate call
c
      cost = p(3,3)/sqrt(p(1,3)**2+p(2,3)**2+p(3,3)**2)
      call hf1 (1,cost,sngl(wgt(1)))
      return
      end
```

```
c
c This program generates 1000 events for the particular process
c
      implicit real*8 (a-h,o-z)
c
c Initialize DELTGC
c
      call Init_Model (iret)
      if (iret.ne.0)             STOP
c
c All is OK. Calculate cross-section.
c
      call Get_CS (sig,err,chi2)
c
c Now, generate 1000 events
c
      call Get_Events (1000000)
c
      stop
      end
c-------------------------------
      logical function User_Kinematics (p)
c-----------------------------------------
      implicit real*8 (a-h,o-z)
      real*8 p(4,6)
c
      User_Kinematics = .FALSE.
c                                    Define your cuts here
      if ( NOT true for your cuts ) return
c
      User_Kinematics = .TRUE.
c
      return
      end
c-------------------------------
      subroutine User_Event (p,wgt)
c-----------------------------------------
      implicit real*8 (a-h,o-z)
      real*8 p(4,6),wgt(327)
      integer nevents
      data nevents/0/
      save nevents
c
      nevents = nevents + 1
      if (nevents.gt.1000) stop
c
```

```
c
c Do something with this event
c
c
      return
      end
```

References

[1] Physics at LEP2, Eds. G.Altarelli, T.Sjöstrand and F.Zwinger, *CERN* 91-01.

[2] G.P. Lepage, *Jour. of Comp. Phys.*, **27**, 192 (1978).

[3] D. Danekaert, P. De Gausmaecker, R. Gastmans, W. Troost, T.T. Wu, *Phys. Lett.* **114B** (1982) 203

[4] F.A. Berends et al., *Nucl. Phys.* **B264** (1986) 243

[5] K. Hagiwara, D. Zeppenfeld, *Nucl. Phys.* **B274** (1986) 1

[6] K. Gaemers and G. Gounaris, *Zeit. Phys.* **C1** (1979) 259.

[7] K. Hagiwara, K. Hikasa, R.D. Peccei, D. Zeppenfeld, *Nucl. Phys.* **B282** (1987) 253.

[8] F.A. Berends and A.I. van Sighem **HEP-PH-9506391** and *Preprint* INLO-PUB-7/95.

[9] M. Böhm, A. Denner, *Preprint* CERN-PPE-92-002

[10] H. Aronson, *Phys. Rev.* **186** (1969) 1434.

[11] M. Bilenky, J.L. Kneur, F.M. Renard, D. Schildknecht, *Nucl. Phys.* **B409** (1993) 22.

[12] G.J. Gounaris, F.M. Renard and G. Tsirigoti, *Phys. Lett.* **B350** (1995) 212.

[13] U. Baur, J.A.M. Vermaseren, D. Zeppenfeld, *Nucl. Phys.* **B375** (1992), 3

[14] Y.Kurihara, D. Perret-Gallix, Y. Shimizu, *Phys. Lett.* **B349** (1995), 367.

[15] E.N. Argyres, A. Denner, S.Dittmaer, J. Hooland, R. Kleiss, C.G. Papadopoulos, G. Passarino, *Preprint* INLO-PUB-8/95 (1995)

[16] Wim Beenakker et al., *Preprint* NIKHEF-96-031 (1996).

[17] G.Altarelli, R.Kleiss, C.Verzegnassi (eds.); F.Boujema and F.M.Renard in Polarization at LEP, CERN-88-06.

[18] P.Chiappetta, F.M.Renard and C.Verzegnassi, *Preprint* PM/95-39, CPT-95/P.3247 (1995).