# EVA — A Short Description

The programme EVA simulates the process $e^+e^- \to \pi^+\pi^-\gamma$. For a given set of general and experiment specific input parameters it generates sets of four-vectors of the outgoing particles distributed according to the probability to observe such events in the experiment. EVA also provides the value for the total or differential cross section and offers the possibility of presenting various distributions as histogrammes.

EVA was programmed in Fortran 77 and makes use of the standard random number generator *RANMAR [1]*, the routine *Collinear Real Radiation [2]* and the routine *Hbook [3]*. Histogrammes are produced with the help of $PAW^{++}$ [5].

## 1 Frames of reference

All cuts are given in the laboratory (lab) frame. Before the electron or positron emit a collinear (non-observed) photon, they have the same energies and move in opposite directions in the lab frame. Their total energy squared is called $s$.
After the emission of the collinear photon, the total energy squared of the $e^+e^-$ is reduced to $s'$. The new centre of mass frame after the collinear radiation is referred to as the "$e^+e^-$ centre of mass frame".
The outgoing particles are: photon, $\pi^+$ and $\pi^-$. The $\pi^+\pi^-$ centre of mass frame is used for some of the calculations.

## 2 Input

All constants and experiment specific parameters must be given in an input file. The standard name of this file is "input.dat" but can be changed in the programme. The values of the input parameters can be varied by the user.

**Note:** Their order must not be changed unless the input routine in the programme is also accordingly changed.

EVA uses a number of physical constants. These are:

$\pi = 3.14$
$\alpha = 1/137$ – the fine structure constant
$m_e$ – the electron mass
$m_\rho$ – the mass of the $\rho$ meson
$m_\pi$ – the mass of the pion
$\Gamma_\rho$ – the total decay width of the $\rho$ meson
$\Gamma_{ee}$ – the partial decay width for $\rho \to e^+e^-$
$m_{\rho'}$ – the mass of the $\rho'$ meson
$\Gamma_{\rho'}$ – the total decay width of the $\rho'$ meson
$m_\omega$ – the mass of the $\omega$ meson
$\Gamma_\omega$ – the total decay width of the $\omega$ meson

All masses and widths must be given in $GeV$.

For the parametrization of the pion form factor [4], two more constants are needed:

$a$ – this corresponds to $\alpha$ from table ??? in [4]
$b$ – this corresponds to $\beta$ from table ??? in [4]

The remaining input parameters correspond to the specific experimental settings:

$s$ – the total energy squared of the electron and positron
$E_{min}$ – the minimal energy for the outgoing hard photon to be detected
$q2_{min}$ – the minimal invariant mass squared for the $\pi^+, \pi^-, \gamma$. This value is needed for the collinear radiation routine and corresponds to an upper cut on the energy of the (unobserved) collinear photon that is radiated by the electron or positron.
$g_{min}$ – the minimal energy of the hard, non-collinear, observed photon. Apart from detector characteristics, there can be other reasons to restrict the photon energy. This is realised through $g_{min}$. If, however, the value given for $g_{min}$ is smaller than the value of the detector specific cut $E_{min}$ , EVA will automatically set $g_{min} = E_{min}$.

$\Theta_{\gamma min}$ – the lower cut on the photon azimuthal angle in the lab frame
$\Theta_{\gamma max}$ – the upper cut on the photon azimuthal angle in the lab frame
$\Theta_{\pi min}$ – the lower cut on the pion azimuthal angle in the lab frame

$\Theta_{\pi max}$ – the upper cut on the pion azimuthal angle in the lab frame
$\Theta_{\gamma\pi}$ – the minimal difference between the pion and photon angle
$\Theta_{detector}$ – the minimal angle for the particles to be detected

The energies and energies squared must be given in $GeV$ and $GeV^2$.
The angles must be given in degrees, with respect to the electron beam.

**Note:** Detection of the particles requires the cuts on their azimuthal angles (because of the position of the detectors relative to the beam). If, for example, only the photon has to be detected and the pion cut can be dropped, the user is supposed to set the pion cuts to $\Theta_{\pi min} = 0°$ and $\Theta_{\pi max} = 180°$.
**Note:** The lower cut on the photon energy $g_{min}$ shields the infrared divergence for $E_\gamma \to 0$.

# 3 Basic Outlines

## 3.1 A Rough Description of Monte Carlo Integration

For the Monte Carlo Integration of

$$\int_{x_{1,min}}^{x_{1,max}} \int_{x_{2,min}}^{x_{2,max}} \dots \int_{x_{n,min}}^{x_{n,max}} f(x_1, x_2, \dots, x_n) dx_1 dx_2 \dots dx_n$$

first of all the maximum $max$ of the integrand has to be found. This can be done analytically (if the function is simple) or — as in EVA — by scanning the function. Then, a set of random variables $(x_1, x_2, \dots, x_n, z)$ is generated where $x_i \in \{x_{i,min}, x_{i,max}\}$ and $z \in \{0, max\}$. The value $f$ of the function for this set of $x_i$ is calculated. If $z < f$ then the set of $x_i$ (event) is accepted, if $z > f$, the set of $x_i$ is rejected. The number of accepted events divided by the number of generated events and multiplied by the "integration volume" $max \cdot (x_{1,max} - x_{1,min}) \cdot (x_{2,max} - x_{2,min}) \dots (x_{n,max} - x_{n,min})$ gives an estimated value for the integral. The more events are accepted, the more precise is the estimate.

## 3.2 Event Generation in EVA

The event generation in EVA is based on a Monte Carlo integration. The function to be integrated is the differential cross section for the process $e^+e^- \to \pi^+\pi^-\gamma$. The set of randomly generated variables fully describes the kinematics of the outgoing particles and is used to calculate the particles' four-vectors. The accepted sets of four-vectors (events) will be distributed according to the correct probabilities to observe such events.

**Note:** for a detailed description of the single Monte Carlo steps in EVA see section 5.2, "The Main Programme".

# 4 Output

EVA collects the accepted events, i.e. the sets of four-vectors for the outgoing particles, in a separate output file. At the beginning of each run, the user is asked to enter the name of this output file (see section 4.4). All four-vectors are given in the lab frame. If the output is needed in a different way, this can easily be realised.

# 5 How To Use EVA

## 5.1 Number of Generated Events

After starting the programme, EVA will ask how many sets of random variables ("events") shall be generated for the Monte Carlo routine. The number of events *accepted* and returned in the output file will depend on the specific input data, i.e. the values of the cuts, but typically the acceptance rate is around 18 %.

## 5.2 Scanning of the Integrand

Next, EVA asks how many points shall be used to scan the integrand in order to find the maximum. We recommend that not less than $10^5$ points are used for this purpose. For good results and high accuracy the number of scanning points should be not much smaller than the number of generated events. If during the integration EVA finds a value of the integrand to be bigger than the scanned (and therefore not exact) maximum, it will give a warning. Then EVA must be restarted with more points used for the scanning.

**Note:** A higher number of scanning points increases both accuracy and running time.

**Note:** At the end of the integration, EVA displays the value of the scanned maximum as well as the maximal value of the integrand found during integration. This gives an additional check. In the best case, the latter is slightly smaller than the former or equal to it. If, on the other hand, the maximal value of the integrand is distinctly higher than the scanned maximum and there were several warnings, the scanning was much too rough and the number of scanning points will have to be increased considerably.

## 5.3 Option: Initial State Radiation Only

The matrix element of the process $e^+e^- \to \pi^+\pi^-\gamma$ consists of three terms: initial state radiation, final state radiation and the interference. In addition to studying the complete process, i.e. taking into account all three terms, EVA offers the option to study initial or final state radiation only. This can be useful to get an understanding of how each of the graphs contributes to the complete process. EVA will ask the user which matrix element it is supposed to use.

### 5.4 Output File

At the beginning of the programme, a new output file is produced. The user is asked to enter the path and name for this file. If a file with the name entered already exists, EVA will produce an error message.

### 5.5 Production of Histogrammes

For the production of histogrammes, EVA requires some more input data. Each time the programme runs, it will ask how many histogrammes it has to produce. If the answer entered is "0", no more information will be asked for. Otherwise, the user will have to make some specifications. For more details, see section "Histogrammes".

### 5.6 Results and Checks

At the end of the programme, EVA displays the total number of accepted events, the value of the cross section in *nbarn*, the value of the scanned maximum and the biggest value of the integrand found during the Monte Carlo routine.

# 6 Details of the Programme

## 6.1 List of Variable Names Used

(In alphabetical order)

Angles in the lab frame are given with respect to the electron beam direction, angles in the pions' centre of mass frame are given with respect to the $\pi^+$ direction.
The four-momentum of a particle $p$ is written such that $p(0)$ is the energy of the particle, $p(1)..p(3)$ are the components of the three-momentum.

### Double Precision Variables

| | |
|---|---|
| *a* | a value needed for the substitution that smoothes the infrared divergence peak |
| *accecut* | the minimal angle for the outgoing particles to be detected |
| *al* | a value that is needed for the calculation of the pion form factor and parametrises the $\omega$ contribution |
| *alpha* | the fine structure constant $\alpha$ |
| *b* | a value needed for the substitution that smoothes the pion form factor (Breit-Wigner-) peak |
| *be* | a value that is needed for the calculation of the pion form factor and parametrises the $\rho'$ contribution |
| *Beta* | the relative velocity of the $e^+e^-$ centre of mass frame (where the random four-vectors are produced) and the lab frame (where the value of the integrand is calculated) |
| *c* | a value that is needed for the substitution that smoothes the pion distribution in the pions' centre of mass frame |
| *cosmin* | the cosine of the upper photon angle cut in the lab frame |
| *cosmax* | the cosine of the lower photon angle cut in the lab frame |
| *costheta* | the cosine of the photon azimuthal angle, $\cos\theta_\gamma$, in the $e^+e^-$ centre of mass frame where it is randomly generated |
| *costheta2* | the cosine of the $\pi^+$ azimuthal angle, $\cos\theta_{\pi^+}$, in the $\pi^+\pi^-$ centre of mass frame where it is randomly generated |

| | |
|---|---|
| *cos2min* | the cosine of the upper pion angle cut in the lab frame |
| *cos2max* | the cosine of the lower pion angle cut in the lab frame |
| *cos3* | the $\pi^+$ azimuthal angle in the lab frame |
| *cos4* | $180°-$ the $\pi^+$ angle in the lab frame |
| *dme* | the mass of the electron. This value is needed in the subroutine *radiation*. The electron mass is neglected in all other calculations. |
| *dw* | the error estimate for the value of the cross section |
| *dx* | the substitution function that smoothes the photon angular distribution $dx = 1/(L(1-\cos^2\theta))$, $L$ is described below. |
| *e_beam* | the energy of the electron beam, $e_{beam} = \sqrt{s}/2$ |
| *Emin* | the minimal energy for the outgoing particles to be detected |
| *E01* | the energy of the electron after the collinear radiation |
| *E02* | the energy of the positron after the collinear radiation |
| *fact* | the factor by which the integrand must be multiplied due to collinear radiation |
| *Gam* | $\gamma = 1/\sqrt{1-\beta^2}$ describes the Lorentz transformation from the $e^+e^-$ centre of mass frame (where the random four-vectors are produced) to the lab frame (where the value of the integrand is calculated) |
| *gammarho* | the total width of the $\rho$ meson |
| *gmin* | the minimal energy cut on the photon in the lab frame |
| *Gminp* | the photon energy cut *gmin* is transformed to the centre of mass frame. The value of the transformed cut is always bigger than *Gminp*. |
| *gomega* | the width of the $\omega$ meson |
| *grhoee* | the partial width of the $\rho$ for its decay into electron and positron |

*grhol* — the width of the $\rho'$ meson

*gross* — the maximum of the integrand found during integration

*inte* — the value of the integrand

*L* — a value needed for the substitution that smoothes the photon distribution in $\cos\theta_\gamma$. The simple substitution function $1/(1-\cos^2\theta)$ is divided by the value $L$ such that the integral over it is exactly 1. The integration then becomes completely independent of $s'$, the total energy squared of the $e^+$ and $e^-$ after the collinear radiation.

*mat* — the function that calculates the value of the matrix element squared

*matrixe* — the value of the matrix element squared

*Mmax* — the scanned maximum of the integrand

*momega* — the mass of the $\omega$ meson

*mpi* — the mass of the pion

*mrho* — the mass of the $\rho$ meson

*mrhol* — the mass of the $\rho'$ meson

*n* — the array variable $n(1)..n(4)$ counts how many of the generated events have been used in each of the four subsitution branches

*num* — the value by which the matrix element must be multiplied to yield the differential cross section. This value consists of the phase space factors, the vertex factors and a numerical value that turns the units of the result into *nbarn*.

*phi* — the polar angle of the photon in the $e^+e^-$ centre of mass frame where it is randomly generated

*phi2* — the polar angle of the $\pi^+$ in the $\pi^+\pi^-$ centre of mass frame where it is randomly generated

*photonwinkel* — the function that generates the photon azimuthal angle, $\cos\theta$, in the $e^+e^-$ centre of mass frame *phot1cut* — the lower cut on the photon azimuthal angle in the lab frame, in degrees

*phot2cut* — the upper cut on the photon azimuthal angle in the lab frame, in degrees

*Pi* — the number $\pi$

*piphcut* — the minimal difference between the photon and pion azimuthal angles in the lab frame. This separation is another cut, its value must be given in the input file.

*pi1cut* — the lower cut on the pion azimuthal angle in the lab frame, in degrees

*pi2cut* — the upper cut on the pion azimuthal angle in the lab frame, in degrees

*qq* — the energy squared, $q^2$, of the $\pi^+, \pi^-, \gamma$-system. This value is randomly generated.

*qqmin* — the lower boundary on $q^2$ in the centre of mass frame. Normally, $q^2_{min} = 4m^2_\pi$, but its value can be changed in the programme.

*Qqmaxp* — the maximum of the upper cut on $q^2$ in the $e^+e^-$ centre of mass frame. This value depends on $s'$, the centre of mass energy squared, and $g_{min}$, the cut on the photon energy in the lab frame.

*q0* — the energy of the photon in the $e^+e^-$ centre of mass frame

*q2_min* — the lower cut on the $\pi^+, \pi^-, \gamma$ invariant mass. This value is needed for the subroutine *radiation*. It restricts the energy of the collinearly radiated photon that is not observed. The value of *q2_min* must be given in the input file.

*r* — a value needed for the substitution that smoothes the pion distribution in the $\pi^+\pi^-$ centre of mass frame

*ran1* — a random number needed for the subroutine *radiation*

*ran2* — a random number needed for the subroutine *radiation*

*s* — the total energy squared of the $e^+$ and $e^-$ before the collinear radiation. Its value must be given in the input file.

| | |
|---|---|
| *Sp* | the total energy squared of the $e^+$ and $e^-$ after the collinear radiation |
| *subs* | the value of the "substitution function" that smoothes the peaks in the $q^2$- and $\cos\theta$-distributions |
| *subsfunktion* | the function that calculates the value of the "substitution function" |
| *t* | a value needed for the substitution that smoothes the pion distribution in the $\pi^+\pi^-$ centre of mass frame |
| *tr* | the array variable $tr(1)..tr(4)$ counts how many events have been accepted in each of the four substitution branches |
| *vel* | the four-momentum of the electron in the lab frame |
| *vgamma* | the four-momentum of the photon in the lab frame |
| *vol* | the "integration volume" for the Monte Carlo routine, i.e. the product of the integration intervals and the maximum of the integrand, *Mmax* |
| *vpiminus* | the four-momentum of the $\pi^-$ in the lab frame |
| *vpiplus* | the four-momentum of the $\pi^+$ in the lab frame |
| *vpos* | the four-momentum of the positron in the lab frame |
| *w* | the value of the cross section |
| *z* | the randomly generated variable for the Monte Carlo routine |

**Integer Variables**

| | |
|---|---|
| *anzahl* | the number of histogrammes produced |
| *bins* | the number of bins used for each histogramme |
| *cl* | denotes whether the programme is run with collinear radiation included or excluded |
| *h* | $h = 1, 2$ is a loop variable: The main part of the programme is run twice. For $h = 1$, EVA finds the maximum of the integrand. For $h = 2$, the integral is calculated. |

| | |
|---|---|
| *j* | the loop variable for the Monte Carlo loop |
| *k* | the number of events generated in each of the two runs. In the first run, $k = nm$, in the second run, $k = nges$. For the scanning, i.e. the first run, less events will be needed. |
| *nges* | the number of events produced in the second run for the calculation of the integral |
| *nm* | the number of events produced in the first run to scan the integrand |
| *reject* | used as true/false variable for the rejection of unwanted events (such as events with too small photon energies) |
| *sun* | array variable. $sun(0)$ denotes whether the $q^2$ generation uses the Breit-Wigner- or the *ln*- branch of the substitution function<br>$sun(1)$ denotes which branch of the subsitution function is used in the $\cos\theta_2$ generation |
| *welche* | this variable can take values from 1 to 4. It depends on $sun(0)$ and $sun(1)$ and denotes which branch is used in the current calculation |

**Real Variables**

| | |
|---|---|
| *Ar* | is a *10*-dimensional array of random numbers between *0* and *1*. This variable is produced by the subroutine *RANMAR* and used for the event generation. |
| *oben* | the upper limit for the production of the *i*-th histogramme. This is an array variable. |
| *unten* | the lower limit for the production of the *i*-th histogramme. This is an array variable. |

**Character Variables**

| | |
|---|---|
| *fname* | the name of the output file |
| *rad* | denotes whether EVA uses the complete matrix element or initial/final state radiation only |

| | |
|---|---|
| *titel* | the title of the $i$-th histogramme. This is an array variable. |
| *wohin* | the file name of the histogramme file |
| *xachse* | this array variable denotes which distribution is depicted in the $i$-th histogramme.<br>At the start of the programme, the user is asked to specify which distributions should be produced. |

## 6.2 The Main Programme

This section describes how the Monte Carlo Routine is realised in the programme EVA. The numbers in round brackets are the numbers that mark the corresponding lines in the programme.

At the start of the programme, EVA gets all input variables including the cuts from the input file *input.dat* and by asking for data (1). The histogramme production is initialized by calling the routine *hist_init* (2).To initialize event generation, the subroutine *radiation* has to be called with the parameters $\{0, 1, 1, 1, 1, 1\}$ (3). The ouput file is opened (4).

There are two runs of the main programme part (6). In the first, the maximum of the integrand is determined, in the second, the integral is calculated. The first run starts with the number of generated events equal to the number of scanning points, $k = nm$ (5). In this run, the maximum of the integrand is found by generating random events, calculating the value of the integrand *inte* for each of these events and comparing this value with the biggest value found so far, $gross$ (17). At the end of the first run, $k$ is set to $k = nges$ (24) and $Mmax$ is set to $Mmax = 1.1 \times gross$ (25) to be on the safe side. In addition, if in the second run it turns out that there are values of the integrand even bigger than $Mmax$, there will be a warning (18).

In the second run $nges$ events are generated. All counting variables such as $tr$ and $n$ are set to zero before the loop is entered (7). At the beginning of each loop, the variable $reject$ is also set to zero ("don't reject") (8). The random number generator $RANMAR$ is called (9), then, if collinear radiation is required, the subroutine *radiation* provides the new energies of the electron and positron, $E01$ and $E02$ as well as the additional factor $fact$ (10). If collinear radiation is not demanded, EVA sets $E01 = \sqrt{s}/2 = E02, fact = 1$. The new invariant mass $s'$ as well as $Beta, Gam$ for the Lorentz transformation and the transformed maximal cuts $Gminp, Qqmaxp$ are calculated (11).

Now, EVA calls the subroutines and functions that provide the set of variables to determine an event, namely $\{qq, costheta, costheta2, phi, phi2\}$ (12). These variables are in the centre of mass frame.
Where possible, the variables are produced such that there values after the boost to the lab frame will lie inside the cuts. In some cases, such as the $q^2$ generation, we were not able to achieve this. For these cases, additional rejection steps are performed later in the programme.

To make the integrand smoother and increase the acceptance rate, appropriate variable transformations (substitutions) are necessary. These are realised in the respective subroutines. The main part of the programme need not care about the form of the substitutions.

However, to understand each step of the main programme, one must know that there are two “2-component substitutions“, namely in the $q^2$ and $\cos\theta_2$ integrations. “2 component“ means two different variable transformations that are performed alternately. For each of the two 2-component substitutions, EVA produces an additional random number to decide which of the two transformations will be used in the current Monte Carlo run. Thus, there are four possible combinations of transformations. The variables *sun* and *welche* fix which of these four combinations or “branches“ is calculated in the current run (13).
The number of events generated and accepted must be counted separately for each branch.

After the production of the random variables, the subroutine *vektoren* provides all four-momenta in the lab frame (14). The $\pi^+$ angles *cos*3 and *cos*4 are calculated for the histogramme production. Then, the additional rejection steps are performed by the subroutine *sortiere* (15): Events with the photon energy too small ($E_\gamma < gmin$) and with unappropriate pion azimuthal angles (smaller/bigger than the cut angles) are rejected, i.e. the rejection variable *reject* is set to $reject = 1$.

If $reject = 0$ and the event in the current run is not rejected, the value of the matrix element squared is then calculated by the subroutine *mat* (16). The remaining factors needed to calculate the value of the integrand are called with the subroutines *integfaktoren* and *subsfunktion*. The value of the integrand, *inte*, is calculated.
If on the other hand the event was rejected by *sortiere*, EVA sets $inte = 0$.

*inte* is compared with the biggest value so far found for the integrand, *gross* (17). In the first run of the main part of the programme, this is done to find the maximum of the integrand, in the second run, it serves to check that the used *Mmax* was big enough. There is a warning in the second run ($h = 2$) if $inte > Mmax$ (18).

Now, for $h = 2$, the randomly produced event is either accepted or rejected. To decide this, the additional variable $z$ is compared to the value of the integrand, *inte* (19). If $z > inte$, the event is rejected. Nothing else is

calculated, EVA starts to generate a new event.
If $z < inte$, the event is accepted. It is counted in the variable *tr* (20), the corresponding four-vectors are written to the output file (21), and the event is added to the histogrammes (22). Then, EVA returns to the beginning of the loop (23).

At the end of the Monte Carlo routine, the result for the value of the cross section is calculated from *vol*, *tr* and *n* (26). An error estimate is given by the subroutine *devia* (27). The histogrammes are written to a hbook file by calling *hist_end* (28). Finally, EVA displays the number of accepted events, the value of the cross section, the error estimate, and, for a check, the biggest value of the integrand found, *gross* as well as the value of the scanned maximum, *Mmax* (29).

### 6.3 The Subroutine *input*

Input variables: —
Output variables: *nges, nm, rad, cl, fname, s, COMMON blocks const, const2, param, cuts*

The subroutine *input* is called at the very beginning of the programme. It gets all input variables from the input file (see section 1) and asks for additional information: How many events shall be generated? Should the complete process or initial or final state radiation only be calculated? Should the current run consider collinear initial state radiation or not? What is the name of the output file? — To make sure that the demanded values for the angular cuts are sensible regarding the minimal angle for detection, *input* contains an additional check.

### 6.4 The Subroutine *histin*

Input variables: —
Output variables: *wohin, COMMON block hipara*

The subroutine *histin* is called at the beginning of the programme and gets all input data for the production of the histogrammes: How many histogrammes should be produced? — If the answer is *0*, no more information will be asked for. Otherwise, *histin* will ask the user to specifiy which distributions should be presented in the histogrammes, with which limits, in how many bins. There is also the option to give a title to each histogramme. Finally, the user must enter the name of the *Hbook* file he wants the histogrammes to be saved in.

**Note:** For more information on how to produce histogrammes, see section "Histogrammes".

**Note:** For more information on the histogramme routines, see description of the subroutines *addiere, hist_init, hist_add, hist_end* as well as [3] and [5].

## 6.5 The Subroutine *hist_init*

Input variables: *anzahl, unten, oben, bins, titel*
Output variables: —

The subroutine *hist_init* initialises histogramme production and makes use of *PAW*. Several user-specified histogrammes are produced in one run.

## 6.6 The Subroutine *radiation*

The subroutine *radiation* generates the energies of the electron and positron after the emission of collinear photons. It was written by H. Czyz and uses formulae that are described in [4].
This subroutine has to be called once before the generation loop starts in order to set up the parameters.

## 6.7 The Subroutine *qquadrat*

Input variables: *Sp, qqmin, Qqmaxp, Ar*
Output variables: *a, b, qq, sun*

The subroutine *qquadrat* generates the value for $q^2$ in the centre of mass frame. For this generation, a two-component-substitution is needed to smooth the Breit-Wigner, i.e. pion form factor, and the infrared divergence peak. The substitution for the infrared divergence (at $q^2 \to s' = Sp$) is a logarithmic substitution and corresponds to $a, amin, amax, fak1, qq01$. For the Breit-Wigner peak (at $q^2 = m_\rho$) the substitution is an *arctan* substitution and corresponds to $b, bmin, bmax, fak2, qq02$. The random variable $p$ decides, and *sun* denotes, which of the substitutions is used in the current generation. An additional check guarantees that the generated $q^2$ lies inside the generation limits.

**Note:** All cuts are given in the lab frame. When boosted to the centre of mass frame, where they are needed to determine the generation limits, they become $s'$ (or $\beta$) dependent. To make the generation limits $s'$ independent, an additional substitution is performed. This is why $a, b$ are given to the main programme.

**Note:** For more details about the substitutions, see section "Substitutions"

### 6.8 The Function $qq01$

Input variables: $Sp, amin, a, fak1, Ar$
Output variable: $qq$
The function *qq01* calculates the value for $q^2$ in the centre of mass frame, using the logarithmic substitution.

### 6.9 The Function $qq02$

Input variables: $b, bmin, fak2, Ar$
Output variable: $qq$
The function *qq02* calculates the value for $q^2$ in the centre of mass frame, using the *arctan* substitution.

### 6.10 The Subroutine *photonwinkel*

Input variables: *Beta, cosmin, cosmax, Ar*
Output variables: *costheta, L*

The subroutine *photonwinkel* generates the photon angle $\cos\theta_\gamma$ in the centre of mass frame. To smooth the integrand that behaves like $1/(1-\cos^2\theta_\gamma)$, a simple *tanh* substitution is used. An additional check guarantees that the value for $\cos\theta_\gamma$ lies inside the generation limits.

**Note:** All cuts are given in the lab frame. When boosted to the centre of mass frame, where they are needed to determine the generation limits, they become $s'$ (or $\beta$) dependent. To make the generation limits $s'$ independent, an additional substitution is performed. This is why $L$ is given to the main programme.

**Note:** For more details about the substitutions, see section "Substitutions"

### 6.11 The Subroutine *pionwinkel*

Input variables: *cos2min, cos2max, s, q2_min, gmin, Ar*
Output variables: *costheta2, c, r, t, sun*

The subroutine *pionwinkel* generates the pion angle $\cos\theta_{\pi^+}$ in the pions' centre of mass frame. For this generation, $s'$ independent limits could be found. A two-component logarithmic substitution is used where $r, rmin, rmax$ correspond to the first and $t, tmin, tmax$ correspond to the second substitution. $p$ decides, and $sun$ denotes, which of the two substitution is used in the current generation. An additional check guarantees that the value for $\cos\theta_{\pi^+}$ lies within the generation limits.

**Note:** For more details about the substitutions, see section "Substitutions"

### 6.12 The Subroutine *vektoren*

Input variables: *Sp, qq, q0, phi, costheta, phi2, costheta2, Beta, Gam*
Output variables: *vgamma, vpiplus, vpiminus, vel, vpos*

The subroutine *vektoren* calculates the four vectors of all the particles in the lab frame from the randomly generated variables that are in the $e^+, e^-$ centre of mass frame or $\pi^+\pi^-$ centre of mass frame, respectively.
Electron, positron and photon vectors are first calculated in the $e^+, e^-$ centre of mass frame from $s', q^2, \cos\theta_\gamma, \phi_\gamma$, i. e., in terms of the programme variables, from $Sp, qq, costheta, phi$. The pion vectors are calculated in the pions' centre of mass frame. They are then boosted to the $e^+, e^-$ centre of mass frame by the subroutine *bost*1. Then, all four vectors are boosted from the $e^+, e^-$ centre of mass frame to the lab frame by the subroutine *bost*2.

### 6.13 The Subroutine *bost*1

Input variables: *ruhvektor, phi, costheta, betrag1, q0*
Output variable: *labvektor*

The subroutine *bost1* boosts vectors that are given in the $\rho$ rest frame to the $e^+, e^-, \gamma$ centre of mass frame. The boost is determined by the photon angles, the photon energy (*betragl*) and the energy of the intermediate particle, $q_0$ — in fact, the transformation is the product of a simple boost and a rotation. $m(i, j)$ are the elements of the transformation matrix.

### 6.14 The Subroutine *bost2*

Input variables: $Beta, Gam, Valt$
Output variable: *vneu*

The subroutine *bost2* boosts vectors from the $e^+, e^-$ centre of mass frame to the lab frame. $m(i, j)$ are the elements of the boost matrix.

### 6.15 The Subroutine *sortiere*

Input variables: *vgamma, vpiplus, vpiminus, costheta, COMMON block cuts*
Output variables: *reject*

The subroutine *sortiere* rejects all unwanted events.
The cuts for the pion angles, *pi1cut, pi2cut, piphcut* are given in the lab frame and cannot easily be transformed to the pions' centre of mass frame where the pions are generated. Therefore, EVA always generates the pions from 0° to 180° in their centre of mass frame, then boosts the four vectors to the lab frame and there checks whether the pion angles lie within the cuts. This check is done by *sortiere*. If the pion angles do not lie inside the cuts, the variable *reject* is set to $reject = 1$, and for the event in question the matrix element will be equal to zero. (*reject* works like a $\Theta$ function.)
A check is also necessary for the photon energy, $vgamma(0)$, but for a different reason: The cut on the photon energy is given in the lab frame. When boosted to the centre of mass frame, this cut will depend on the photon angle. As event generation in the centre of mass frame (here: $q^2$ generation) requires fixed limits, the maximum of the cut in $\cos\theta_\gamma$ is used and thus some unallowed events with too small photon energies will be produced. These are also rejected by *sortiere*.

### 6.16 The Function *mat*

Input variables: *rad, Sp, qq, vgamma, vpiplus, vpiminus, vel, vpos*
Output variable: *mat*

The function *mat* calculates the value of the matrix element squared for the given event. Depending on *rad*, either initial state radiation only, final state radiation only, or the complete process are considered. The subroutine *skalarprodukte* provides the values for the scalar products of all four vectors, these are used to calculate the matrix element. *mat* also calls the function *formfaktor* that gives the value of the pion form factor.

### 6.17 The Subroutine *skalarprodukte*

Input variables: *q2, pi1, pi2, p1, p2*
Output variables: *p1$p2, p1$q2, p2$q2, pi1$pi2, p1$pi1, p1$pi2, p2$pi1, p2$pi2, q2$pi1, q2$pi2*

The subroutine *skalarprodukte* supplies the scalar products needed to calculate the value of the matrix element squared. *a$b* here is just a name for the scalar product of the two vectors $a$ and $b$; the subroutine simply assigns the correct vectors to the variables *a$b*. The products themselves are calculated by the function *dot*.

### 6.18 The Function *dot*

Input variables: $a, b$
Output variable: *dot*

The function *dot* calculates the scalar product of the two input four-vectors $a, b$ in Minkowski space: $a \cdot b = a^0 b^0 - a^1 b^1 - a^2 b^2 - a^3 b^3$.

### 6.19 The Function *formfaktor2*

Input variables: $a, b$
Output variable: *formfaktor*
The function *formfaktor* calculates the value of the pion form factor. Depending on which part of the matrix element is just calculated in *mat*, the input

values $a, b$ are equal to $q^2, q^2$ (initial state radiation), $s', s'$ (final state radiation) or $q^2, s'$ (interference). *formfaktor* calls the function *BW* to calculate the values of the Breit-Wigner functions.

**Note:** The parametrisation of the form factor is taken from [4].

### 6.20 The Function *BW*

Input variables: *m, breite, x, k*
Output variable: *BW*
The function *BW* calculates the value of the Breit-Wigner function for given mass $m$ and width *breite* and for the value $x$ that is either $s'$ or $q^2$. Where the Breit-Wigner of the $\omega$ is demanded, *BW* takes the width to be $x$ independent. See also [4].

### 6.21 The Subroutine *integfaktoren*

Input variables: *Sp, qq, costheta, L*
Output variables: *num, dx*

The subroutine *integfaktoren* calculates all factors the matrix element squared must be multiplied with to yield the differential cross section in units of nanobarn. These are: the phase space factor, the vertex factors, a numerical factor for the correct units, and *dx* that denotes the substitution function for the simple *costheta* substitution (see section 5.6, "*photonwinkel*").

### 6.22 The Function *subsfunktion*

Input variables: *Sp, qq, c, a, b, costheta2*
Output variable: *subsfunktion*

The function *subsfunktion* supplies the value of the substitution functions for the two 2- component substitutions. For more details, see section "*substitutions*".

### 6.23 The Subroutine *addiere*

Input variables: *nges, vol, qq, costheta, costheta2, cos3, cos4*
Output variables: —

The subroutine *addiere* adds an accepted event to the histogrammes. This is done by checking which distributions should be produced and calling the subroutine *hist_add* with the relevant values.

### 6.24 The Subroutine *hist_add*

Input variables: $sr, x, wert, unten, oben, bins$
Output variables: —

The subroutine *hist_add* adds an accepted event to all histogrammes. It makes use of the *Hbook* routine $hf1$. Several user-specified histogrammes can be produced in one run.

### 6.25 The Subroutine *devia*

Input variables: $nges, n, tr, r, t, vol$
Output variable: $dw$

The subroutine *devia* calculates an estimate for the deviation of the cross section value. The equation for this estimate was taken from the book *Numerical Recipes in Fortran, W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T Vetterling* and was derived from the formula for standard deviation:

$$dw = vol \times \sqrt{\frac{tr}{n} - \left(\frac{tr}{n}\right)^2}$$

### 6.26 The Subroutine *hist_end*

Input variables: $fname$
Output variables: —

The subroutine *hist_end* saves all histogrammes. It makes use of the *Hbook* routine *hrput*. Several user-specified histogrammes can be produced in one run.

## 7 Referencies

[1] CERN program library
[2] M. Caffo, H. Czyz, E. Remiddi, Nuovo Cim. 110A (1997) 515
Phys. Lett. B327 (1994) 369
[3] CERN program library
[4] J. Kühn, A. Santamaria, $\tau$ decays to pions
Z. Phys. C48:445-452, 1990
[5] PAW