GRACE User's manual
version 2.0

MINAMI-TATEYA collaboration
T.Ishikawa, S.Kawabata, Y.Kurihara, Y.Shimizu
National Laboratory for High Energy Physics(KEK),
Tsukuba, Ibaraki 305, Japan
T.Kaneko
Faculty of general education, Meiji-gakuin University,
Totsuka, Yokohama 244, Japan
K.Kato
Department of physics, Kogakuin University,
Tokyo 160, Japan
H.Tanaka
Faculty of general education, Rikkyo University,
Tokyo 171, Japan

August 1, 1994

Acknowledgements

On the way to develop the program system GRACE we have received much encouragement and support from many people. We would like to thank our colleagues in TRISTAN theory group, particularly T. Munehisa, N. Nakazawa and J. Fujimoto for discussions which were helpful for getting the system better. We are also grateful to express our sincere gratitude to Professors H. Sugawara, S. Iwata and J. Arafune for the encouragement. Discussions with colleagues in Nuclear Physics Institute of Moscow State University and LAPP, Laboratoire d'Annecy-le-Vieux de Physiques des Particules, with whom we have been collaborating on the automatic calculation of Feynman amplitudes during these years, were very fruitful and useful for this work.

We are indebted to companies Fujitsu limited, Intel Japan K.K., KASUMI Co. Ltd and SECOM Co. Ltd. for their kind supports and understanding our work. A part of the calculations was performed on FACOM M1800, AP1000, VP series, S series, HITAC S820, M880, 3050, HP 9000 and Intel iPSC/860.

This work was supported in part by Ministry of Education, Science and Culture, Japan under Grant-in-Aid for International Scientific Research Program No.03041087 and No.04044158.

Notice

This document is the revision of the Ref.[8] which also includes the theoretical background of `GRACE` system. For the method to calculate the cross section, there is no difference between this version and the version 1.0 in Ref.[8]. However, the user interface is much upgraded here, so that the generation of code for kinematics is possible.

We expect the user who will make publication by use of the system refers to the `GRACE` system.

E-mail address : `grace@minami.kek.jp`

ftp location : `/kek/minami/grace` at `ftp.kek.jp`

Contents

Chapter 1

Introduction

1.1 What is the problem ?

During the last two decades, it has been established that the gauge principle governs the interactions between elementary particles. In electroweak theory, leptons and quarks are interacting through exchange of three kinds of gauge bosons, photon, Z^0 and $W^{\pm}$. The assumed gauge group is $SU(2)_L \times U(1)$ and the original gauge symmetry is broken by the non-zero vacuum expectation value of Higgs field.[1] On the other hand strong interaction between quarks is described by color $SU(3)$ gauge group. [2] All the experimental facts seem to support these theories at present. Though it is still an open question how these different kinds of forces are unified into more fundamental theory, it is now of no doubt that these theories contain some truths and will remain as effectively correct ones.

This success of gauge theories or standard models of elementary particles, implies that we have definite Lagrangians and thus we can, in principle, predict any process based on these Lagrangians in perturbation theory. When one wants to perform calculation in this way, however, one meets a technical difficulty due to the complexity of the interaction Lagrangian. This is particular to non-abelian gauge theory in which we have three- and four-point self-couplings of gauge bosons as well as interactions of unphysical particles such as Goldstone bosons or ghost particles in general covariant gauge fixing. Hence even in the lowest order of perturbation, that is, in tree level, one finds a number of diagrams for a given process when the number of final particles increases. For example, we have only 3 diagrams for $e^+e^- \to W^+W^-$, but when one photon is added, $e^+e^- \to W^+W^-\gamma$, then 18 diagrams appear even after omitting the tiny interaction between $e^{\pm}$ and scalar bosons(Higgs and Goldstone bosons). Addition of one another photon, $e^+e^- \to W^+W^-\gamma\gamma$, yields 138 diagrams. Further if one wants to make more realistic calculation around the threshold of $W^{\pm}$ pair production, taking into account the decay of $W^{\pm}$, say, $W^- \to e^-\bar{\nu}_e$ and $W^+ \to u\bar{d}$, then one has to consider 24 diagrams for $e^+e^- \to e^-\bar{\nu}_e u\bar{d}$ and 202 for $e^+e^- \to e^-\bar{\nu}_e u\bar{d}\gamma$. In unitary gauge, as only physical particles appear in the Lagrangian, the numbers of diagrams are less than those mentioned above.

One may think that it is enough to select several diagrams which dominate the cross section. Even if one can find such dominant diagrams, one has to respect the gauge invariance among this subset of diagrams. Usually number of diagrams in the gauge invariant subset is not so small. For example, for the process $e^+e^- \to \nu_e\bar{\nu}_e W^+W^-$, we have 60 diagrams in all. Among them 30 diagrams form one gauge invariant set and the rest does another one. Hence still we meet the same difficulty to handle with many diagrams. In addition, there remains a possibility that the experimental cuts imposed on the final particles renders the dominant diagrams to be less prominent and all diagrams give somehow the same order of magnitude to the cross section. If this is the case, one has to keep whole the diagrams in the calculation after all.

Through the numerous experiments done at e^+e^- colliders, we have learned that higher order corrections should be included when we want to compare theories with experimental data in detail. This implies that we have to calculate at least one-loop corrections to a given process. As an example, consider the process $e^+e^- \to W^+W^-\gamma$. To regularize the infrared divergence due to soft photon emission, we have to include loop diagrams for $e^+e^- \to W^+W^-$ beyond the tree level, which contain virtual photon exchange and remove the divergence when combined with real photon emission process. The requirement of gauge invariance among one-loop diagrams demands, in turn, inclusion of other one-loop diagrams with exchanges of Z^0, $W^\pm$ or other possible particles. Then it is clear that the total number of diagrams becomes very huge and it is almost impossible even to enumerate all diagrams. In many cases it seems out of ability of mankind. For simple W-pair production, in general covariant gauge, the number is around 200 diagrams in the same approximation stated above, but for $e^-\bar{\nu}_e u\bar{d}$ it amounts more than 3,700.

Facing to the difficulty described above, we cannot help to find some ways to get rid of. As a solution we can choose the following one: As diagrams are constructed based on a set of *definite rules*, Feynman rules, it is natural to develop a computer code which can generate all the diagrams to any process, once initial and final particles are given. It should be able not only to enumerate diagrams but also generate automatically relevant amplitudes to be evaluated on computers, in other words, create a FORTRAN source code ready for amplitude calculation. `GRACE` (Ref.[3]) is such a system that realizes this idea and help us to reduce the most tedious part of works.

1.2 What we can do with `GRACE` ?

Before introducing what `GRACE` system can provide, let us remind the standard way to calculate cross sections at the tree level. Usually it consists of the following several different steps:

1) Specify the process.

2) Choose appropriate models.

3) Fix the order of perturbation(at the tree level, this is unique).

4) Enumerate all possible diagrams.

5) Write down amplitudes.

6) Prepare the kinematics for final particles.

7) Integrate the amplitude squared in the phase space of final particles, including experimental cuts, if necessary.

8) Generate events so that the simulation of the process in a detector is available.

9) Check the results.

Among these steps the first three, 1), 2) and 3), are trivial matter. For the step 7) one can rely on well established programs which are designed to make integration of multi-dimensional variables. This is of no problem, except for CPU-time, once the kinematics, step 6), is written so that the estimate of the integral is reliable within required accuracy. The step 8) is related with the preceding step. The last step 9) could be done to compare the results with other calculations or with approximated one. Hence the most tedious steps are 4) and 5). `GRACE` is a system of program packages for this purpose, namely, it carries out these most tedious steps on computers to save our elaboration.

1.2.1 What `GRACE` provides us?

The present version of `GRACE` generates:

- All the tree diagrams for a given process when the order of perturbation is fixed.
- Diagrams on X-window and its print-out.
- FORTRAN source code which contains helicity amplitude of the process.
- Default values of all physical constants, *except for the strong coupling constant.*
- Interface routines to the program package `CHANEL` (Ref.[4],[5]), which contains subroutines designed to evaluate the amplitude.
- Default code for kinematics.
- Interface routines to the multi-dimensional integration package `BASES` (Ref.[6]).
- Interface routines to the event generation package `SPRING` (Ref.[6]).
- Test program for gauge invariance check of the generated amplitude.
- Any diagram and its amplitude can be omitted in the calculation by setting the appropriate flags off. In the integration step the unitary gauge is the default(see section 3.2.1).

What the user should do first is to tell GRACE a set of parameters which specifies the process considered. It should include

1) names of initial particles,

2) names of final particles,

3) order of perturbation in QED, electroweak and/or QCD.

4) code number of kinematics.

in the given format explained later. For 4), the list of available built-in kinematics is provided with the system.

When GARCE is initiated with the data file containing these inputs, it constructs all possible diagrams and creates an output file to draw all Feynman diagrams for the convenience of the user to look them by eyes. At the same time a set of FORTRAN subprograms is generated. These include those which are needed to calculate the amplitude with the help of CHANEL, to integrate over phase space by BASES and to generate events by SPRING.

After all the programs are successfully generated, the remaining tasks for user are

1) to examine the kinematics,

2) to check and to edit some parameters in a few subroutines,

3) to check the gauge invariance of the amplitude,

4) to supply the value of strong coupling constant.

For the item 1) and 2), the system generates default code. However, it may not give the best solution. If the convergence of integral is not good, the user must switch to the other kinematics sample, or the user must write it by oneself. Also, some important parameters, e.g., the center-of-mass energy, are written in the generated source code, so that the user would change them by editor. The variety and the location of these parameters are given in the document in the Appendix. The item 3) can be done by a sequence of command as will be described later. If colored quanta exists in the process, one must edit the source to add an multiplicative factor since $g_s = 1$ in the generated code (item 4)) while the color factors are properly included in the code.

1.2.2 What structure GRACE has ?

In this subsection we show how the whole system of GRACE is constructed and how each step proceeds. The system consists of the following four subsystems, whose interrelation is depicted in Fig.1.1.

(1) **Graph generation subsystem**
When initial and final states of the elementary process are given as the input as well as the orders of couplings, a complete set of Feynman graphs is generated according to the theoretical model defined in a model definition file. ([1],[2],[7]) For the time being QED, Electroweak and QCD models in the tree level are supported. The information of generated graphs is stored in a file as an output.

Reading the graph information from the file, the graph drawer displays the Feynman graphs on the screen under the X-Window system or prints them on a paper.

(2) **Source generation subsystem**
From the graph information produced by the first subsystem, a FORTRAN source code is generated in a form of program components suited for the numerical integration package `BASES` and the event generation package `SPRING`.

The source code is constructed based on our helicity amplitude formalism, which consists of many calling sequences of subprograms given in `CHANEL` and its interface routines.

In addition to these program components, the subsystem generates a main program, by which the gauge invariance of the generated amplitudes can be tested.

(3) **Numerical integration subsystem**
Combining the generated source code together with the kinematics routines and the `GRACE` library, the numerical integration is performed by `BASES` to obtain the total cross section. For this, however, in general, one may have to prepare the kinematics routines when the default one is not appropriate. As the output of integration, the numerical value of total cross section, the convergency behavior of integration, one and two dimensional distributions of the cross section are given besides the probability information in a file, which is used in the event generation. Looking the convergency behavior carefully one can judge if the resultant value is reliable or not.

(4) **Event generation subsystem**
Using almost all the same subprograms in the integration, events with *weight one* are generated by the event generation program `SPRING`. To achieve a high generation efficiency, it uses the probability information produced by `BASES`. Conceptually, `SPRING` samples a point in the integration volume according to the probability. If the probability information is a complete one, the sampled point is exactly corresponding to a generating event. Since, however, it is impossible to get a complete information numerically, the sampled point is tested whether it is accepted or not. The user can record and analyze the generated events according to the information in the following chapters.

1.2.3 How to do with kinematics ?

In order to get the numerical value of cross section, we integrate the differential cross section over the phase space of final particles. As the integral is multi-dimensional, 4 for 3-body, 7 for 4-body and 10 for 5-body process(if the cylindrical symmetry is assumed around the initial beam axis), we usually use adaptive Monte Carlo integration packages. (In our system BASES is assumed.) We have to express all momenta (or equivalently invariants composed of them) of final particles by independent integration variables. Generally speaking, the integration routine feeds a set of random numbers in the space of given dimension. Let us denote these random numbers as

$$\texttt{X(I)}, \texttt{I} = \texttt{1}, \cdots, \texttt{NDIM},$$

and assume their values are normalized in, say, $[0, 1]$. (In BASES, the upper and lower bounds for X(I) can be arbitrary numbers.) Then we have to translate these variables into four-momentum of final particle, say J-th particle, P(1,J), P(2,J), P(3,J), P(4,J) of total N particles (in GRACE, P(4,J) is the energy),

$$\texttt{X(I)} \Longrightarrow \texttt{P(K,J)}. \quad \texttt{K} = 1, \cdots, 4, \quad \texttt{J} = \texttt{1}, \cdots, \texttt{N}$$

This is known as kinematics for the given process. This mapping is not always unique and in some cases a single value of X(I) may correspond to multi-value of particle momenta.

GRACE, unfortunately, only can give a candidate of the kinematics. The reason is that the present popular integration packages, such as BASES or VEGAS, utilize a special algorithm to search for the singularities of the integrand. The matrix element squared, the integrand, becomes singular when the denominators of propagators of internal particles become very small compared with the typical energy of the process considered. This happens when a mass of an internal line is very small. As is well known, if a singularity is running along the diagonal in a plane of two integration variables, these programs cannot give reliable estimate of the integral, because they fail to catch the singularity at all. In order to get good convergence of the integration over many iterations, all the singularities must be parallel to the integration axes. This means that these peaks located in the space of kinematical variables, are mapped onto the line of constant value of some X(I). In order to do this, we have to choose very carefully the transformation between random numbers and kinematical variables. The typical kinds of singularities we meet in real calculation are as follows;

- mass singularity
- infrared singularity
- t-channel photon exchange
- resonance formation(decay of heavy particles)

(Precise description of how to deal with these singularities will be found in section 2.6 in Ref.[8]).

In some processes the number of independent variables is greater than that of singularities, and one can easily find a kinematics which is suitable to make them smooth. If this is not the case, however, one may not be able to find such good kinematics to avoid diagonal singularity even after much efforts. Hence it is quite difficult to give the general kinematics which is capable of dealing with all kinds of singularities at once, or a single set of transformations.

Considering the situation described above, `GRACE` only provides some sample of kinematics to users. So the following points are left for the users:

- Select the ordering of particles to use kinematics.
- Select the kinematics among the candidates.
- Set physical parameters, e.g., $\sqrt{s}$.
- Change the parameters for the integration, e.g., the number of iteration, number of required accuracy and so on.
- Revise or write the source code by the user when all candidates provided by the system are no good.

As was stressed before, the nature of singularity is related to the physical problem at hand, so that the user knows best about the tuning of kinematics.

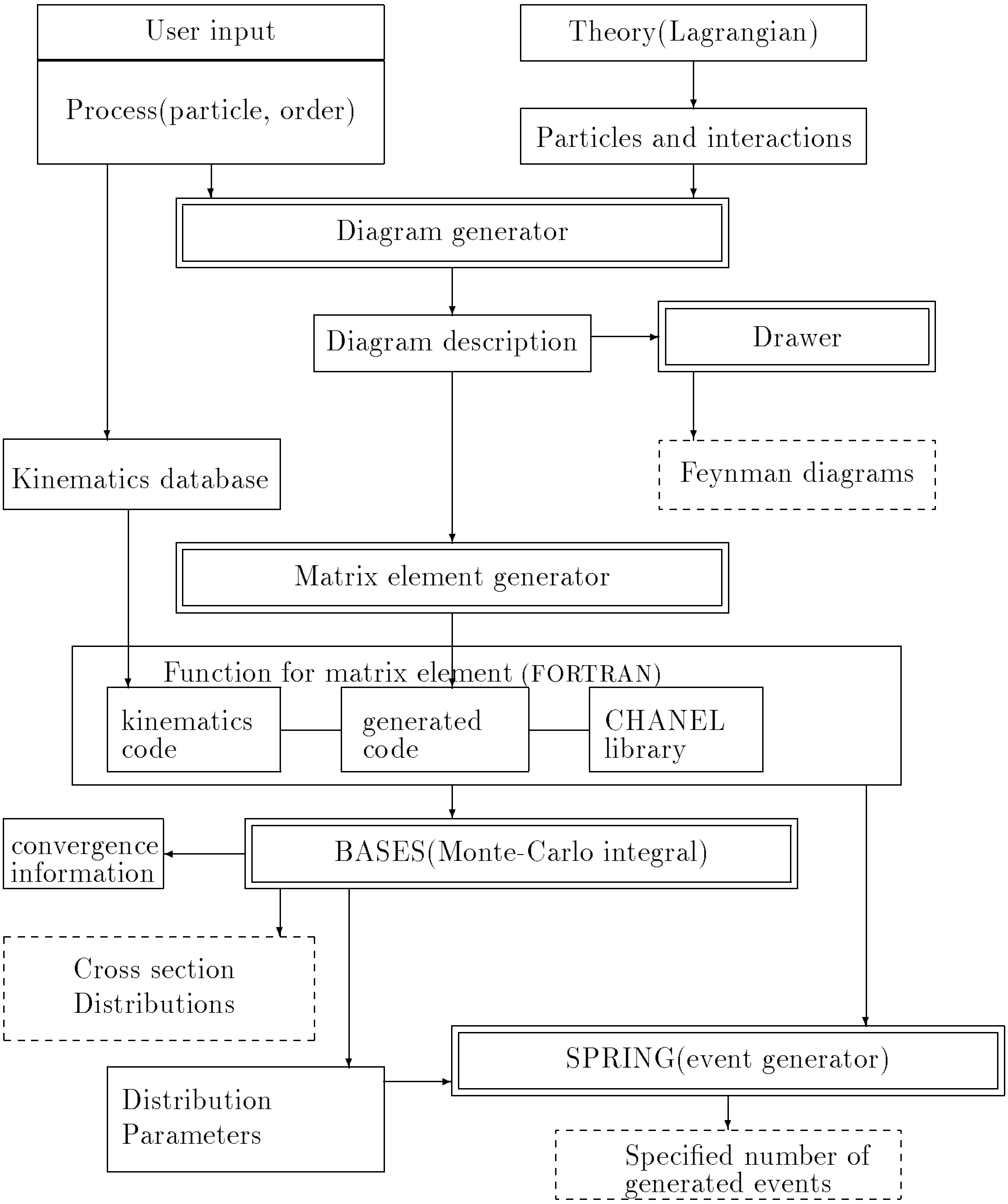

Figure 1.1: GRACE system flow

Chapter 2

How to use GRACE system

The GRACE system was developed on a main frame computer FACOM, but now it is available on UNIX system. The usage of GRACE on FACOM is basically to use JCL in the Batch Job environment. Although an interactive mode is available on an UNIX system, the interpreter of GRACE on UNIX system is still very primitive. It will be improved in near future. We suppose the X-Window system is available on UNIX system, which is used for drawing Feynman graphs on the screen. The UNIX systems, where we have installed and tested GRACE, are SUN SPARC and HP9000/750. Here we only describe the usage on UNIX machine. The description for FACOM and that to run on a parallel processor is given in Ref.[8].

2.1 Environment

At first user should add the following statement in the file ".cshrc". [1]

```
set path = ($path /usr/local/grace )
setenv GRACEDIR   /usr/local/grace
setenv GRACETABLE /usr/local/grace/data/particle.table
```

Here /usr/local/grace is the directory where GRACE system is installed. It depends on the way how the system is installed. The directory is connected the the name GRACEDIR.

Also, the name GRACETABLE is defined and it specifies the physical model. Two model definition files, "particle.table" and "particle.table0", are prepared in GRACE system as the standard. Both files represent the standard model, i.e., electroweak theory and the QCD. In "particle.table0", the couplings between Higgs and light fermions are suppressed in order to reduce the number of unimportant graphs. [2] As the default the file $GRACEDIR/data/particle.table is used for the model definition file. If user wants to use his own model definition file instead of this default file, then

[1] We assume user uses C shell.

[2] Detailed description is given in chapter 6 in Ref.[8].

user should connect the input file name (for example "`myparticle.table`") with the environment parameter "`GRACETABLE`" by `setenv`.

2.2 Generate Feynman graph

It is recommended to create a new directory for the calculation of one physical process. For the process $e^+e^- \rightarrow W^+W^-\gamma$, as an example, we create a directory `eewwa` and move to the directory as below: [3]

```
o..............................................................o

grace% mkdir eewwa
grace% cd eewwa

o..............................................................o
```

2.2.1 Definition of physical process

For the graph generation the following input file to define the physical process is necessary. As an example, we consider the process $e^+e^- \rightarrow W^+W^-\gamma$. We can use the file `/data/d5120`, where the following parameter is saved.

```
o..............................................................o

* 5120      E+E-  => W+ W- A     TREE
WORDER       3
INITIAL EL   1
INITIAL ELB  1
FINAL   AB   1
FINAL   WB   1
FINAL   WBB  1
KINEM  3002
END
```

Fig. 2.1 Input file `eewwa.input` for the process $e^+e^- \rightarrow W^+W^-\gamma$

```
o..............................................................o
```

[3] In this section `grace%` stands for the `UNIX` prompt.

Specification of defining physical process is described in subsection 3.1.1. When there is the target process in the file `$GRACEDIR/data/Index`, the input parameters for the process can be found in a file `dnnnn`, where the number `nnnn` is the process number defined in the file `Index`. When there is no candidate for input-parameter file, the user has to write it by oneself according to the specification described in subsection 3.1.1.

One must specify the kinematics code number after the keyword `KINEM` in the input file. Short list of these built-in kinematics is found in subsection 3.1.1. When `KINEM` line is omitted, the system only generates the template code for kinematics, so that the user should complete it according to the details in the next chapter. There is an index file `$GRACEDIR/kinem/Index` which shows the list of available kinematics provided by the system.

The ordering of the particles in the input file is important since the kinematics assigns particles according to the order that they appear in this file. Each kinematics is defined like

$$p_1 + p_2 \to p_3 + p_4 + p_5$$

if it is a 2-body to 3-body reaction. In the example in Fig.2.1, γ is assigned to p_3 and so on. For instance, let's suppose that the kinematics specified by `KINEM` line has feature to catch the singularity for the case where particle p_3 is emitted along the beam axis. [4] Then if there is such particle in the process to be studied, the user must write the input file so that the particle corresponds to p_3. The description of each kinematics is found in the corresponding section in the Appendix.

2.2.2 Execute graph generation

Suppose a file `eewwa.input` (like Fig.2.1) is created in the current directory `eewwa` as the input file defining the physical process, then the graph generation procedure starts by typing the command `gengraph` and the name of input file:

```
grace% gengraph eewwa.input
```

It reports some information on the generation including the total number of generated graphs. Also the graph generator creates files "`INTBL`" , "`OUTDS`" , and "`out.grf`" in the current directory.

[4]This is the case for `KINEM 3002`.

2.3 Draw Feynman graph

This step can be skipped if the user does not want to see graphs. However, it is recommended to do this step for the confirmation of the input. Also, one can print Feynman diagrams at a postscript printer. A Feynman graph drawer is initiated by the command:

o..o

```
grace% treegrf
```

o..o

Then it shows first 16 diagrams in 4 by 4 format, a few control buttons, number of total graphs in covariant gauge and that in unitary gauge, and status flags. (If the total number of diagrams is less than 10, the format is adjusted to the number.) The set of diagrams shown in the window is called as 'page'. The number of diagrams shown in a page can be changed. This drawer uses OSF/Motif interface and a few buttons are shown at the right-hand side of window. The user can use the following functions:

Button / Sub-menu	Function
`Quit`	Exit from the drawer.
`Next Page`	Show next page.
`Previous Page`	Show last page.
`+5Page`	Show 5-th page next.
`-5Page`	Show 5-th page previous.
`Scale up`	Make the size of graph larger.
`Scale down`	Make the size of graph smaller.
`Option menu`	
/ *Graph number*	On/Off display of graph number.
/ *Vertex number*	On/Off display of vertex number.
/ *Particle name*	On/Off display of particle name.
`Graph menu`	
/ *Covariant gauge*	Show diagrams in covariant gauge.
/ *Unitary gauge*	(Default)
/ *Selected graph*	Only show selected graphs.
`Mode Selection`	
/ *Drawing mode*	(Default)
/ *Select mode*	To select, click on a graph you choose.
`EPS output`	Create a file named `treegrc*.eps` which saves
	the graphic image of the page shown
	in eps format. (`*=0,1,2,..`)

The drawer refers file `out.grf` created by `gengraph`.

2.4 Generate source code

After the graph generation, FORTRAN source code is generated by typing the command:

```
o..............................................................o

grace% genfort

o..............................................................o
```

The procedure `genfort` uses the files `INTBL` and `OUTDS` as input, which should not be changed. The files `xxxxxx.f` and `Makefile` are created as the output in the current directory, where `xxxxxx` corresponds to a name of program components described in section 3.2. For instance, subprogram `SETMAS` appeared in section 3.2 is created in a file `setmas.f`.

2.5 Editing FORTRAN source codes

When the result of the program generated automatically is not satisfactory, the user might edit the following components. Also, when the user wants to make histograms, one must edit source code to call `XHINIT`, `DHINIT`, `XHFILL`, `DHFILL`. Here, `*HINIT`'s are to be called for initialization of histogram, and `*HFILL` fills a value on the histogram. The first character `X` and `D` specify the 1–dimension histogram and the 2-dimensional scatter plot, respectively.

1) Initialization routine `KINIT`
   One can choose options and parameters by editing `KINIT` as follows: (See next chapter for technical details.)

   – Modification of physical parameters
     Some physical parameters are to be modified according to the user's will. For instance, the value of center-of-mass energy, the value of the angle cut-off etc. are given in `KINIT`. Further, the kinematics may have some options. For instance, if there is an integration variable corresponds to an invariant mass, M^2, then one may choose to use either M^2 or $\log M^2$ in the kinematics by setting a flag in `KINIT`. Another possible example is the flag to suppress the integration over rotation around beam axis. The variety of these parameters depends on the nature of kinematics at hand. The list of such parameters is also described in the Appendix.

   – Integration parameters which control the action of `BASES`.

NDIM	The number of dimensions of integral
NWILD	The number of wild variables
XL(i),XU(i)	The lower and upper bounds of integration variable X(i)
IG(i)	The grid optimization flag for i-th variable
NCALL	The number of sampling points per iteration
ACC1	The expected accuracy for the grid optimization step
ITMX1	The maximum iteration number for the grid optimization
ACC2	The expected accuracy for the integration step
ITMX2	The maximum iteration number for the integration step

– Initialization of histograms and scatter plots
If the user needs histograms for some distributions, the initialization routines XHINIT and DHINIT should be called for each histogram and scatter plot, respectively. The system automatically generates source code to call these so as to make histograms with respect to each integration variables, X(I), I=1..NDIM and the energies and angles of final particles. The meaning of each X(I) is given in the corresponding section in the Appendix where section name is the code number of kinematics.
If the user does not need these default histograms, one deletes the corresponding lines. Or if the user wants to make histograms for some distributions, one appends lines to call XHINIT or DHINIT according to the specification in subsection 3.5.4.

2) Filling histograms and scatter plots in KFILL
In the generated KFILL by GRACE, histograms for all integration variables and scatter plots for all combinations of them are to be filled by calling XHFILL and DHFILL, respectively. If one has changed the initialization of histograms and scatter plots in KINIT, their filling parts should be also changed (see subsection 3.5.4).

3) If colored particles are included in the process, one must include the strong coupling constant as a multiplicative factor in the function. The generated function properly includes the color factors while it assumes $g_s = 1$. This is because the argument of $\alpha_s(\mu^2)$ cannot be determined uniquely in general. There is a line in the subroutine KINEM to define a variable YACOB. One can multiply g_s^n as

$$\texttt{YACOB} = \texttt{YACOB}* \cdots$$

below that line.

4) Masses and widths.
They are defined in SETMAS subroutine. If one wants to change the value of m_{top}, m_{Higgs} and so on, the user should edit this file.

5) Selection of graphs.
This can be done by setting flags JSELG defined in SETMAS.

2.6 Makefile

The command "genfort" also generates the makefile Makefile. The user creates three executable modules, gauge, integ, and spring by this makefile:

○..○

```
grace% make
```

○..○

An example of Makefile is shown below. The libraries BASES/SPRING, interface to CHANEL and CHANEL are stored in the directory GRACELDIR. The objects commonly used both in BASES and SPRING are defined by macro name OBJS. The macro names INTEG and INTOBJ define the executable and the object of the main program for the integration, respectively. Similarly the macro names SPRING, SPOBJS, GAUGE and GAUGEOBJ are defined.

○..○

```
# FILE "Makefile.f" is generated by GRACE System (Minami-Tateya Group)
# Grace Version   1. 1             10-Aug-94
#
# Makefile for HP
#
SHELL          = /bin/csh
FC             = fort77
#
GRACELDIR      = ........

BASESLIB       = bases
CHANELLIB      = chanel
KINEMLIB       = kinem
BDUMMLIB       = bdummy
#
OBJS           = userin.o amparm.o \
                 func.o   amptbl.o ampsum.o ampord.o \
                 usrout.o kinit.o kinem.o kfill.o setmas.o\
                 am0001.o am0002.o am0003.o am0004.o \
                 am0005.o am0006.o am0007.o am0008.o \
                 am0009.o am0010.o am0011.o am0012.o \
                 am0013.o am0014.o am0015.o am0016.o \
                 am0017.o am0018.o am0019.o am0020.o \
                 am0021.o am0022.o am0023.o am0024.o \
                 am0025.o am0026.o am0027.o am0028.o
INTEG          = integ
INTOBJ         = mainbs.o
SPRING         = spring
```

```
SPOBJS        = mainsp.o spevnt.o spinit.o spterm.o
GAUGE         = gauge
GAUGEOBJ      = gauge.o


all:            $(INTEG) $(GAUGE) $(SPRING)

$(INTEG): $(INTOBJ) $(OBJS) $(GRACELDIR)/lib$(BASESLIB).a \
                      $(GRACELDIR)/lib$(CHANELLIB).a \
                      $(GRACELDIR)/lib$(KINEMLIB).a
$(FC) $(INTOBJ) $(OBJS) -o $(INTEG) -L$(GRACELDIR) \
       -l$(BASESLIB) -l$(CHANELLIB) -l$(KINEMLIB) $(FFLAGS)


$(SPRING): $(SPOBJS) $(OBJS) $(GRACELDIR)/lib$(BASESLIB).a \
                      $(GRACELDIR)/lib$(CHANELLIB).a \
                      $(GRACELDIR)/lib$(KINEMLIB).a
$(FC) $(SPOBJS) $(OBJS) -o $(SPRING) -L$(GRACELDIR) \
       -l$(BASESLIB) -l$(CHANELLIB) -l$(KINEMLIB) $(FFLAGS)


$(GAUGE): $(OBJS) $(GAUGEOBJ) $(GRACELDIR)/lib$(BDUMMLIB).a \
                    $(GRACELDIR)/lib$(CHANELLIB).a \
                      $(GRACELDIR)/lib$(KINEMLIB).a
$(FC) $(GAUGEOBJ) $(OBJS) -o $(GAUGE) -L$(GRACELDIR) \
       -l$(BDUMMLIB) -l$(CHANELLIB) -l$(KINEMLIB) $(FFLAGS)


clean:
\rm -f *.o $(INTEG) $(SPRING) $(GAUGE)
```

Source list 2.1 Makefile for HP9000/750

○...○

2.7 Test of the gauge invariance

The main program `gauge.f` is used to check the generated amplitudes at a point in the integration volume as described in section 3.4. In the main program subroutines `USERIN` and `FUNC` are called, which call the histogram packages. Since the histogram has no meaning in this test, we use dummy library for them stored in the directory `GRACELDIR`. Thus it is not necessary to comment out the statements in the subprograms `FUNC` and `USERIN` for this test, which call relevant histogram routines.

The executable `gauge` is already created and is executed by the following commands:

```
o...............................................................o

grace% gauge

o...............................................................o
```

This reports the value of matrix element at a point in the phase space in covariant gauge and that in unitary gauge. If the two values are same, it passes the check here. The comparison goes as follows:

```
o...............................................................o

    . . . . .
  ANS1   =  .4577959455154742
    . . . . .
         ............
  ANS2   =  .4577959455154741
    . . . . .
         ............
 ANS1/ANS2 - 1 =  2.220446049250313E-16
    . . . . .

o...............................................................o
```

It also reports the relative magnitude of contribution by each graph.

An example of output from the test for the process $e^+e^- \to W^+W^-\gamma$ is shown in section 3.4, where the consistency with 14 digits is found between the covariant and unitary gauges. [5]

It should be noted that this test does not guarantee a complete gauge invariance even though it could give consistency between the two gauges, since it tests only at a specific point in the phase space. It is recommended to test the gauge invariance at several points in the phase space.

[5] However, it might not give long-digit accuracy when unstable gauge particles(W, Z) have finite width. One can check this effect by changing the values of width in `SETMAS`.

2.8 Integration

After examination of the subprograms KINIT and KFILL and a successful test of gauge invariance, we can proceed to the numerical integration by BASES. By make, we have already made the executables integ for the integration and spring for event generation.

For integration the command integ is used.

```
grace% integ
```

The integration package BASES prints the result on the screen at every end of the iteration, and also it writes the same lines together with the histogram output on the file named bases.result at the end.

Normally, if one uses built-in kinematics, the cross section is given in unit of pb.

Before termination of the integration procedure, BASES writes the probability information on a binary file bases.data, which is used for the event generation.

After the integration, the system may issue WARNING messages, if the convergence is not well established. However, this diagnostics message is not absolute, so that the user must be careful for the check of the integration.

It is recommended to look at the integration result carefully, especially over the convergency behaviors both for the grid optimization and integration steps. When the accuracy of each iteration fluctuates, iteration by iteration, and, in some case, it jumps up suddenly to a large value compared to the other iterations, the resultant estimate of integral may not be reliable. There are two possible origins of this behavior; one is due to too small sampling points and the other due to an unsuitable choice of the integration variables for the integrand (see subsection 3.5.5 and also subsection 2.7.4 in Ref.[8]). An example of output for the process $e^+e^- \to W^+W^-\gamma$ is given in subsection 3.5.5.

2.9 Event generation

Since the executable `spring` is created by the make command already, the event generation starts by typing

```
o..........................................................o

grace% spring

o..........................................................o
```

Then `SPRING` reads the probability information from the binary file `bases.data` and asks the number of events with the following prompt:

```
o..........................................................o

Number of events ?

o..........................................................o
```

Here, the user must type the number of events to be generated. The event generation will run until a given number of events are generated or the number of failure for the generation exceeds its given maximum. The reason why we have the maximum number for the failed generation in the event generation is that the generation loop may have a possibility to get into an infinite loop when some mistakes were made (see subsections 3.5.5 item 8, subsections 3.6.1 and 3.6.2).

When the four vectors of generated events are to be written on a file, then this file should be opened in `MAINSP` and the four vectors should be written on the file in `MAINSP`.

In order to estimate the computing time for the event generation, it is recommended to use the expected generation time given in the computing time information of `BASES` output.(see section 3.5.5 item 5)

When the kinematics is made of a single-valued function, the subprogram `FUNC` should be identical both in the integration and event generation. But if it is not the case, `FUNC` in the event generation should be modified from that in the integration as described in subsection 3.6.1.

The output from `SPRING` is written on the file named `spring.result`, which consists of the general information, original and additional histograms, scatter plots, and number of trials distribution. From the original histograms we can see how the generated events reproduce those distributions produced by the integration. In the number of trials distribution we can see the generation efficiency.

Chapter 3

Details of GRACE system

In this chapter, we present technical details of the system. Before coming into the details, it may be useful to summarize briefly here.

Graph generation subsystem

Input :

1) **Definition of physical process**
   Specification method of the physical process is described in subsection 3.1.1.
2) **The model definition file**
   Specification of model is rather complicated, and it is found in chapter 6 in Ref.[8]. We provide a default standard model following Ref.[1], [2] and we recommend to use this model for the first use of this system.

Output :

1) **System files** `OUTDS`, `INTBL`
2) **Graph information file** `out.grf`
3) **Drawn figures**
   Generated graphs are drawn on a graphic device by using the file `out.grf`. They are described in section 3.1.2.

Source generation subsystem

Input :

1) **The model definition file**
2) **System file** `OUTDS`
   which is generated by the graph generation subsystem.

Output :

1) **Generated FORTRAN source code**
   Number of files for FORTRAN source codes are generated by `GRACE`.

Subprograms for amplitude calculation are described in section 3.2. These subprograms use the CHANEL routines through the interface subprograms. Details of the interface subroutines and CHANEL routines will be described in the chapter 7 of Ref.[8].
Also a main program for BASES (section 3.5) and that for SPRING (section 3.6) are generated.

2) **Output of the testing program**
The format of output of the generated test program is given in section 3.4.

Numerical integration subsystem

Input :

1) **Generated FORTRAN source code**
A part of the generated code might need edit. Also if the problem at hand cannot be handled by the built-in kinematics, the user must write it by oneself. The description of related subprograms is given in section 3.3.

Output :

1) **Print out**
The format of output of BASES is given in section 3.5. There may be statistical error in the Monte Carlo integration and systematic error in user's kinematic subroutines. So it is very important to see whether the result is reliable or not.

2) **Probability information file**
As the result of integration, the probability information, contents of histograms *etc.* are saved in this file bases.data, which is used for event generation.

3) **Output file**
The results (1)**Print out**) is also written on th file bases.result for later use.

Event generation subsystem

Input :

1) **Generated FORTRAN source code**

2) **Probability information file**
bases.data which is generated by BASES.

Output :

1) **Print out**
The print out format is given in subsection 3.6.4. This is very useful to see whether the generated events reproduce really the distribution of differential cross section.

2) **Output file for the generated events**
   Generated events are passed to detector simulator or simulator of particle decay. Section 3.6 describes how to deal generated events for this purpose.
3) **Output file**
   The results (1)**Print out**) is also written on th file `spring.result` for later use.

The generated FORTRAN code uses default values of mass parameters, coupling constants and other parameters, whose values are set in the subprograms `SETMAS` and `AMPARM`. If one wants, one can change these values by modifying this subroutine.

Although many physical processes have been calculated for testing the `GRACE` system, it is still possible that a new error may occur in a new reaction. It is important to check the result in a systematic way. Possible origin of error will be

(1) Unsuited kinematical variables to the integrand,

(2) Bugs in the kinematics,

(3) Large numerical cancellation,

(4) Bugs in the `GRACE`.

Numerical cancellation is the most difficult problem to control. Even if the program is logically correct, it is possible to produce completely wrong result. Some of numerical cancellation can be avoided by improving kinematics, but others require modification of generated code.

Anyway one has to check the result intensively. Usual checking method is as follows:

(1) Check gauge invariance of the result,

(2) Check Lorentz frame invariance of the result,

(3) Check numerical stability of the result,

(4) Changing the number of sampling points in the numerical integration,

(5) Comparison with other results.

Before the numerical integration, one should confirm that the generated FORTRAN source code is correct one. `GRACE` system generates a test program, which provides a gauge invariance test by comparing the resultant values on a phase space point for different values of gauge parameters. One can check some kind of numerical cancellation or inconsistency in the generated code. This is the easiest way of checking. However, since this program checks only at one point, one may miss errors in the different region of the phase space.

Since the amplitude is calculated by a numerical way in a special Lorentz frame, one can test the program by changing reference frame. This method also checks numerical cancellation partially, as the four components of momenta are changed.

Direct checking method of numerical cancellation is to change precision of the calculation. If your compiler has an option to change precision of floating point number, it will be easy and powerful method.

The correctness of the kinematics subroutines and statistical reliability will be checked by careful reading of output of `BASES` and changing parameters for `BASES`. If kinematics subroutines fails to catch steep peeks of the differential cross section, the final value may be completely wrong.

3.1 Graph generation

3.1.1 Definition of the physical process

In order to define a physical process we give the order of coupling constants and names of external particles as the input.

Below we show an example, which specifies the process $e^+e^- \to W^+W^-\gamma$.

○..○

```
* 5120     E+ E-  => W+  W-  A       TREE
WORDER      3
INITIAL EL  1
INITIAL ELB 1
FINAL   AB  1
FINAL   WB  1
FINAL   WBB 1
KINEM   3001
END
```

Input file for defining physical process

○..○

The format of input is as follows:

1) **Comment line**
The first line is a comment line, but it should *never* be omitted. It is copied to output files as a header to indicate the process.

2) **The order of coupling constants**
The second line in the example indicates the order of coupling constants.

```
WORDER      3
```

implies that the order of electroweak interaction (order of perturbation) is 3. When one want to restrict the process to pure QED,

```
EORDER
```

should be assigned. It is noted that `WORDER` and `EORDER` are not allowed to set at the same time. For QCD one should give the order of QCD coupling by

```
CORDER.
```

Combination of `WORDER` and `CORDER` or that of `EORDER` and `CORDER` are allowed. In that case the order of each interaction should be defined in different line.

3) **External particles**
To define the external particles, in the first column one has to give whether the current particle is in the `INITIAL` or `FINAL` state. Then name of this particle follows. The list of names is shown in the table below. If it is an anti-particle, `B` should be added to the end of the name. For the W-boson, `WB` defines W^+, so that W^- is written as `WBB`. In the last column the number of identical particles is given by an integer.
Since the ordering of particles in the kinematics is important, the user must carefully place the external particle lines so that the kinematics works efficiently.

name of particle	
WB	W^+
ZB	Z^0
AB	γ
XB	χ^+
X3	χ_3
PH	ϕ (Higgs boson)
NE	ν_e
NM	ν_μ
NT	ν_τ
EL	e^-
MU	μ^-
TA	τ^-
UQ	u-quark
CQ	c-quark
TQ	top-quark
DQ	d-quark
SQ	s-quark
BQ	b-quark
CP	c^+ (ghost for W)
CM	c^- (ghost for W)
CZ	c^Z (ghost for Z)
CA	c^A (ghost for photon)
GL	gluon
CG	c^G (ghost for gluon)

Table 3.1 Names of particles in the default model definition file

4) **Specification of kinematics**
After the keyword `KINEM`, the user specifies the code number of the kinematics to be used. The list of available kinematics provided by the system is found in the Appendix. For each kinematics `nnnn`, there exists a section in the Appendix. The user must consult this document for the usage of the kinematics. The information in Appendix includes the feature of kinematics, the meaning of variables `X(I)`, the list of optional parameters, the hint for the assignment of external particles, and so forth.

When the user does not give this line, a part of source code related to the kinematics are not generated and it is left for the user to complete the program. The detailed information how to construct the code for kinematics is one of the important task of this chapter.

Table 3.1 shows a list of particle names defined in the model definition file, whose format is described in chapter 6 in Ref.[8].

In `UNIX` system, many files named like "`dnnnn`" are given under the directory `$GRACEDIR/data/` as examples of the input file, whose list is in the file `$GRACEDIR/data/Index`. The contents of file "`Index`" is given in Table 3.2, where the last three numbers of each line are the orders of perturbation, `WORDER`, `EORDER` and `CORDER`. If there is the target process in this list, the first number `dnnnn` indicate the file name which contains the input parameters for that process. For example, if one wants to calculate $e^+e^- \to W^+W^-\gamma$, one can use the file `d5120`. When one cannot find the process to be studied, it would be easy to make input file by copying a similar process's file.

The file `particle.table` under the same directory contains all the information on the model used in the graph generation and source generation subsystems.

```
* 5010     E+ E-  => NU_e NU_e Z           TREE
* 5020     E+ E-  => NU_e NU_e H           TREE
* 5030     E+ E-  => E+ E- H               TREE
* 5040     E+ E-  => Z Z H                 TREE
* 5050     E+ E-  => W+ W- H               TREE
* 5060     E+ E-  => Z Z Z                 TREE
* 5070     E+ E-  => W+ W- Z               TREE
* 5080     E+ E-  => t t-bar Z             TREE
* 5090     E+ E-  => t t-bar PH            TREE
* 5100     E+ E-  => H H Z                 TREE
* 5120     E+ E-  => W+  W-  A             TREE
* 6010     E+ E-  => NU_e NU_e W+ W-       TREE
* 6020     E+ E-  => MU+ MU- Gamma Gamma   TREE
* 6030     E+ E-  => NU_mu NU_mu B B-bar   TREE
* 6040     E+ E-  => E+ NU_e t-bar b       TREE
* 6050     E+ E-  => E- NU_e-bar u d-bar   TREE
```

Table 3.2 The list of processes in the file `$GRACEDIR/data/Index`

3.1.2 Drawn Feynman graph

In the graph generation, a file `out.grf` is created under the current directory, where the graph information is stored. By typing command "`treegrf`", Feynman graphs are drawn on the screen when the OSF/Motif on X-Window system is supported.

In Fig.3.1, an example of the Feynman graphs drawn by `treegrf` is shown. The process is $e^+e^- \rightarrow \gamma W^+W^-$ whose input file is given in this section. Here, since unitary gauge is selected, a part of graphs are not shown. If covariant gauge is selected by the button `Graph menu`, the number of diagrams is 28. Also, by the use of the button `Scale down`, the diagrams are shown in 5 by 5 format.

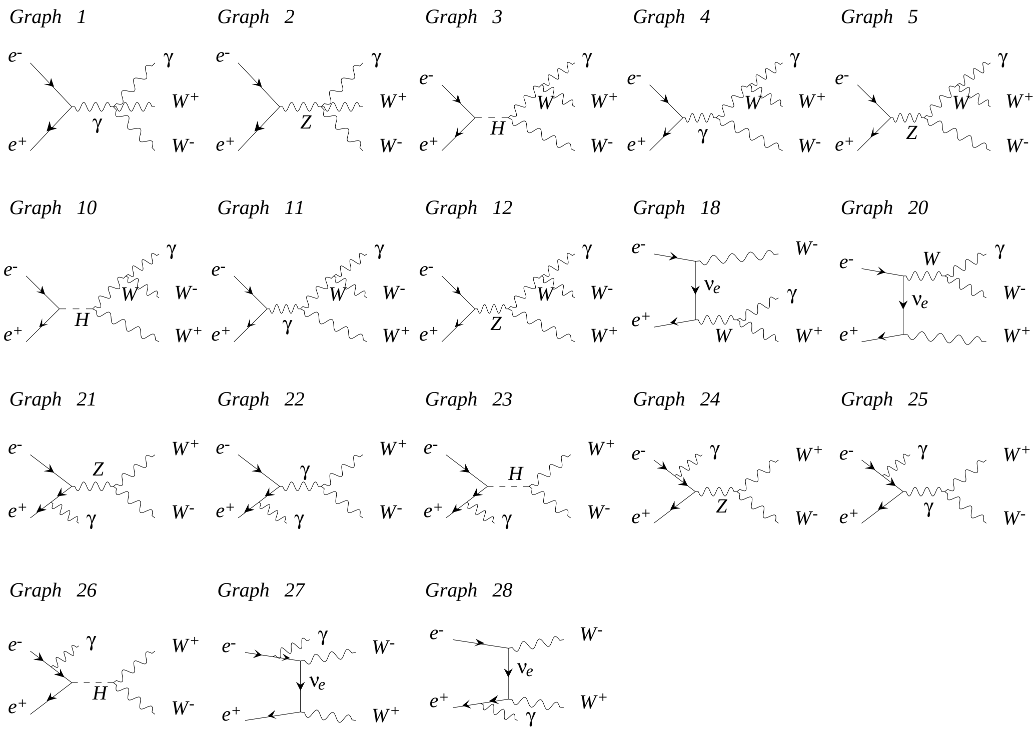

Figure 3.1: An example of drawn graphs for $e^+e^- \rightarrow \gamma W^+W^-$.

3.2 Generated source code

There are three kinds of program components. The first is for the amplitude calculation, the second is necessary for the integration by `BASES` and the third is for the event generation by `SPRING`.

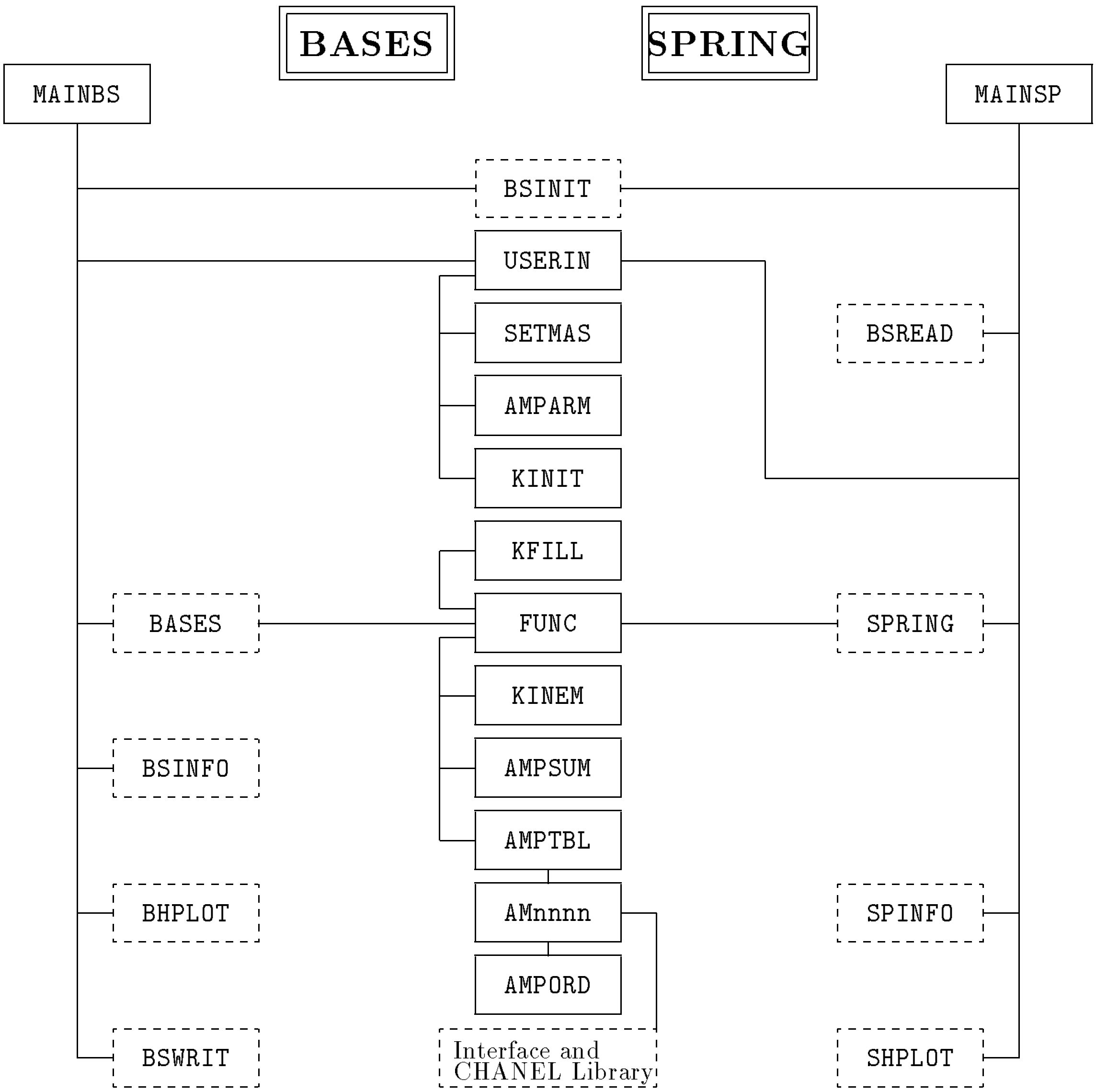

Figure 3.2: Relation among the generated subprograms

The interrelation among the subprograms generated by `GRACE` is depicted in Fig.3.2, where those subprograms in the `solid box` are automatically generated by `GRACE`, while those in the `dashed box` are already contained in other program packages `BASES/SPRING`, interface program library to `CHANEL`, and program package `CHANEL`. This figure omits a few minor modules. The modules located in the middle of the figure are common to BASES and SPRING. [1] The program specifications of the libraries BASES/SPRING, the interface to CHANEL and program package CHANEL are described in sections 3.5.

In the following these program components are summarized.

1) Main programs

MAINBS	*(main)*	is the main program for the integration.
MAINSP	*(main)*	is the main program for the event generation.
GAUGE	*(main)*	is the test main program for the check of gauge invariance.

2) a set of program components for the integration by BASES

USERIN	*(subroutine)*	controls initialization.
SETMAS	*(subroutine)*	defines masses and decay widths of particles.
AMPARM	*(subroutine)*	defines coupling constants and others.
KINIT	*(subroutine)*	initializes BASES, kinematics and user's parameters.
FUNC	*(function)*	calculates the numerical values of differential cross section.
KINEM	*(subroutine)*	derives particle four momenta from the integration variables.
KFILL	*(subroutine)*	fills values in histograms and scatter plots.
AMPTBL	*(subroutine)*	calls AMnnnn to calculate amplitudes.
AMPSUM	*(subroutine)*	sums matrix elements over the helicity states. A matrix element is the square of the sum of amplitudes.
AMnnnn	*(subroutine)*	calculates amplitude of the nnnn-th graph, where the number nnnn of the routine name is equal to the graph number.
AMPORD	*(subroutine)*	arranges amplitudes.
incl1.f	*(include file)*	defines the common variables for masses, amplitude tables *etc.*
incl2.f	*(include file)*	defines the work space for AMPTBL.
inclk.f	*(include file)*	transfers values of masses and a few constants.
USROUT	*(subroutine)*	prints the amplitude summary table. Only used by gauge.

Although the program components KINIT, KFILL, FUNC and KINEM are created automatically by GRACE, they might need modifications. Especially the kinematics routines KINIT and KINEM might need filling up by the user, when the built-in kinematics is not good for the problem at hand. Example of these routines for the process $e^+e^- \to W^+W^-\gamma$ are also given there.

[1]If a multi-valued function appears in the kinematics, FUNC needs trivial modification for the use in SPRING after the integration by BASES.

The subprograms USERIN and FUNC are used both in the numerical integration by BASES and the event generation by SPRING. In the routine USERIN, subroutines KINIT is called to initialize kinematics and other parameters. The routines SETMAS and AMPARM are called for the initialization of physical parameters and amplitude calculation.

The function subprogram FUNC is used for calculating the numerical value of differential cross section, where subroutine KINEM is called for calculating four vectors of external momenta and subroutines AMPTBL and AMPSUM are called for the amplitude calculation. Since specifications for USERIN and FUNC are described in sections 3.5.2 and 3.5.3 respectively, in this section we mention the amplitude calculation part briefly.

3.2.1 Initialization of amplitude calculation

Parameters for the amplitude calculation are set in subprograms SETMAS and AMPARM, and are passed to the relevant subroutines through the several commons, which are given in the include file incl1.f.

Subroutine SETMAS

The structure of subprogram SETMAS is shown in the source list 3.1. In SETMAS the following fundamental parameters are defined.

1) **Constants**
Numerical constants π, π^2, $\pi/180$, GeV/pb and $\alpha = e^2/4\pi$ for the amplitude calculation are set and some of them are passed for later use through the common /AMCNST/ defined in inclk.f.

2) **Selection of diagrams(1)**
If one sets the i-th element of the array JSELG( ) to "zero", then one can omit the corresponding i-th graph and skip the calculation of this amplitude. Each element of the array correspond to the graph number which can be read off from the drawn picture of graphs.

3) **Masses** and **widths** are defined. The user can modify the default values by editing here.

4) **Gauge parameter**
The information about the gauge parameters is summarized in the include file incl1.f.

Calculation either in covariant gauge(R_ξ-gauge) with an arbitrary gauge parameter or in unitary gauge is possible in GRACE system. The distinction between them is given by integer variables in the common /SMGAUS/.

```
COMMON /SMGAUS/IGAUOO, IGAUAB, IGAUWB, IGAUZB, IGAUGL
```

where IGAUAB, IGAUWB, IGAUZB and IGAUGL are the gauge selection flags for photon, $W^{\pm}$, Z^0 and gluon, respectively. Unitary gauge is selected by setting flag IGAUxx to 0 for xx boson. This is effectively equivalent to the case where the gauge parameter of xx boson is set equal to infinity.

For the covariant gauge, four different values of gauge parameters can be set by using an array AGAUGE(i) (i runs from 1 to 4).

In the generated FORTRAN code, unitary gauge is taken as the default gauge.

```
COMMON /SMGAUG/AGAUGE(0:4)
REAL*8 AGAUGE
```

AGAUGE(IGAUAB), AGAUGE(IGAUWB), AGAUGE(IGAUZB) and AGAUGE(IGAUGL) represent the values of gauge parameters α_A, α_W, α_Z and α_G, respectively (see Ref.[8]). To give different values of gauge parameters for each boson, the flags IGAUAB, IGAUWB, IGAUZB and IGAUGL are to be set equal to 1, 2, 3, and 4, respectively, for example. Of course, the values should be set for the variables AGAUGE(IGAUxx)s here.

5) **Spin summation**

The components of spin and polarization vector are controlled by

Fermion	: 0 (helicity = -1), 1 (helicity = $+1$)
Vector boson	: 0, 1 (transverse), 2 (longitudinal).

For each external particle I of non-zero spin, the spin summation is taken from JHS(I) to JHE(I) as follows;

```
      ANS = 0.0
      DO 100 J = JHS(I), JHE(I)
        ANS = ANS + table_of_amplitude(J)
  100 CONTINUE
```

where

```
      JHS(I) = 0
      JHE(I) = LEPEXA - 1
```

and

```
      LEPEXA = 2
```

for the external photon as an example. In the generated code, the spin summation is originally arranged to give unpolarized cross section. The spin freedoms of external particles are given in the include file incl1.f (see Source list 3.3.1) as follows;

LEPEXA	= 2	spin freedom of external photon
LEPEXW	= 3	spin freedom of external $W^{\pm}$ boson
LEPEXZ	= 3	spin freedom of external Z^0 boson
LEPEXG	= 2	spin freedom of external gluon
LEXTRN	= 2	spin freedom of external fermion

The variable ASPIN is the normalization factor of spin average for initial bosons and fermions.

6) **Selection of diagrams(2)**
Also one can choose a class of diagrams by setting flags. If JWEAKB=0, only QED diagrams are selected and if JHIGGS=0 all diagrams including Higgs particle are excluded.

○...○

```
* FILE "setmas.f" is generated by GRACE System (Minami-Tateya Group)
* Grace Version   1. 1          1994-Aug-19
************************************************************************
      SUBROUTINE SETMAS
      IMPLICIT REAL*8(A-H,O-Z)

      INCLUDE 'incl1.f'
      INCLUDE 'inclk.f'
      COMMON /AMSPIN/JHS(NEXTRN), JHE(NEXTRN), ASPIN
*-----------------------------------------------------------------------
* Constants
      PI    = ACOS(- 1.0D0 )
      PI2   = PI * PI
      RAD   = PI / 180.0D0
*      ref .Review of Particle Properties, Phy.Rev.D50(1994)1173
      GEVPB = 0.38937966D9
*     ALPHA = 7.2973503D-3
      ALPHA = 1.0D0/137.0359895D0

*
* Graph selection (1)
      DO 10 NG = 1, NGRAPH
        JSELG(NG) = 1
   10 CONTINUE
*-----------------------------------------------------------------------
* ref.  Review of Particle Properties, Phy.Rev. D50(1994)1173
*
* Mass
      AMWB = 80.22D0
      AMZB = 91.187D0
      AMAB =  0.0D0

      AMXB =   AMWB
      AMX3 =   AMZB
```

```
      AMPH = 2000.0D0

      AMNE = 0.0D0
      AMNM = 0.0D0
      AMNT = 0.0D0
      AMLU = AMNE

      AMEL = 0.51099906D-3
      AMMU = 105.658389D-3
      AMTA = 1.7771D0
      AMLD = AMEL

      AMUQ = 2.0D-3
      AMD0 = 1864.6D-3
      AMJ  = 3.09688D0
      AMCQ = 1.0D0
      AMTQ = 174.0D0
      AMQU = AMUQ

      AMDQ = 5.0D-3
      AMSQ = 100.0D-3
      AMB0 = 5.375D0
      AMU  = 9.46037D0
      AMBQ = 4.1D0
      AMQD = AMDQ

      AMCP = AMWB
      AMCM = AMWB
      AMCZ = AMZB
      AMCA = AMAB

      AMGL = 0.0D0
      AMCG = AMGL
*
* Width
      AGWB = 2.08D0
      AGZB = 2.490D0
      AGAB = 0.0D0

      AGXB = AGWB
      AGX3 = AGZB

      AGPH = 4000.0D0
      . . . . .

* Masses of external particles
      AMASS1(   1) = AMEL
      AMASS1(   2) = AMEL
      AMASS1(   3) = AMAB
      AMASS1(   4) = AMWB
      AMASS1(   5) = AMWB
```

```
      AMASS2(   1) = AMASS1(   1)**2
      AMASS2(   2) = AMASS1(   2)**2
      AMASS2(   3) = AMASS1(   3)**2
      AMASS2(   4) = AMASS1(   4)**2
      AMASS2(   5) = AMASS1(   5)**2
*
* Gauge parameters (default is unitary gauge)
      IGAUAB = 0
      IGAUWB = 0
      IGAUZB = 0
      IGAUGL = 0
      AGAUGE(IGAU00) = 1.0D0
      AGAUGE(IGAUAB) = 1.0D0
      AGAUGE(IGAUWB) = 1.0D0
      AGAUGE(IGAUZB) = 1.0D0
      AGAUGE(IGAUGL) = 1.0D0
*
* Spin average
      ASPIN = 1.0D0
*           1:  EL-  INITIAL   LPRTCL  MASS=AMEL
      JHS(   1) = 0
      JHE(   1) = LEXTRN - 1
      ASPIN = ASPIN/DBLE(JHE(   1)-JHS(   1)+1)
*           2:  EL+  INITIAL   LANTIP  MASS=AMEL
      JHS(   2) = 0
      JHE(   2) = LEXTRN - 1
      ASPIN = ASPIN/DBLE(JHE(   2)-JHS(   2)+1)
*           3:  AB   FINAL     LPRTCL  MASS=AMAB
      JHS(   3) = 0
      JHE(   3) = LEPEXA - 1
*           4:  WB+  FINAL     LPRTCL  MASS=AMWB
      JHS(   4) = 0
      JHE(   4) = LEPEXW - 1
*           5:  WB-  FINAL     LANTIP  MASS=AMWB
      JHS(   5) = 0
      JHE(   5) = LEPEXW - 1
*
* Graph selection (2)
      JWEAKB = 1
      JHIGGS = 1

      RETURN
      END
```

Source list 3.1 subprogram SETMAS

○ . . . . ○

Subroutine AMPARM

In the source list 3.2 the structure of subprogram AMPARM is given, which prepares the following items:

1) **Version number**
The version number of GRACE system is compared with that of the interface package to CHANEL in SMINIT. If they are not consistent, job is terminated for the sake of safety.

2) **Coupling constants**
Coupling constants for various vertices are calculated.

3) **Color factors**
Color factors (the array CF(i,j)) for each combination of two graphs are calculated.

○ . . . . ○

```
* FILE "amparm.f" is generated by GRACE System (Minami-Tateya Group)
* Grace Version   1. 1          1994-Aug-19
************************************************************************
      SUBROUTINE AMPARM
      IMPLICIT REAL*8(A-H,O-Z)

*-----------------------------------------------------------------------
      INCLUDE 'incl1.f'
      INCLUDE 'inclk.f'
*-----------------------------------------------------------------------
      CALL SMINIT(   1,   1)

* Coupling constants
*-----------------------------------------------------------------------
      AMWB2 = AMWB*AMWB
      AMZB2 = AMZB*AMZB
      AMPH2 = AMPH*AMPH
      AMZW2 = AMZB2 - AMWB2
      AMZW  = SQRT(AMZW2)
      RMZW  = AMZB/AMWB
      R2    = SQRT(2.0D0)
      R2I   = 1.0D0/R2
CCCC  ALPHA = 1.0D0/137.0359895D0
      CE2   = 4.0D0*PI*ALPHA
      CE    = SQRT(CE2)
      GW    = AMWB/AMZW
      GZ    = AMZB/AMZW
      GZW   = AMZB/AMWB
      GWZ   = AMWB/AMZB

      QL    = - 1.0D0
      QU    =   2.0D0/3.0D0
      QD    = - 1.0D0/3.0D0
*-----------------------------------------------------------------------
* VVV
      CZWW    =  CE*GW
      CAWW    =  CE
```

```
* VVVV
      CWWAA    =  CE2
      CWWZA    =  CE2*GW
      CWWZZ    =  CE2*GW*GW
      CWWWW    = -CE2*GZ*GZ

* FFW
      GWFL       = CE*GZ*R2I
      GWFR       = 0.0D0
       . . . . .
      CWEL(1,1) = GWFL
      CWEL(2,1) = GWFR
      CWEL(1,2) = CONJG(CWEL(1,1))
      CWEL(2,2) = CONJG(CWEL(2,1))
       . . . . .

* FFA
      GAL      =  QL*CE
       . . . . .
      CAEL(1) =  GAL
      CAEL(2) =  GAL
       . . . . .

* FFZ
      GZA      =  0.5D0*CE*GZW*GZ
      GZC      =  CE/GW
      GZLL     =  - QL*GZC - GZA
      GZLR     =  - QL*GZC
       . . . . .
      CZEL(1) =  GZLL
      CZEL(2) =  GZLR
       . . . . .

* SSV
      CWXP(1) = DCMPLX(0.0D0,  0.5D0*CE*GZ)
      CWXP(2) = - CONJG(CWXP(1))
      CWX3(1) =   0.5D0*CE*GZ
      CWX3(2) = - CONJG(CWX3(1))
      CZXX    =   CE*(0.5D0*GZW*GZ - GW)
      CAXX    = - CE
      CZ3P    = DCMPLX(0.0D0, 0.5D0*CE*GZ*GZW)

* SVV
      CPWW    = CE*AMWB*GZ
      CPZZ    = CE*AMZB*GZ*GZW
      CXWZ(1) = DCMPLX(0.0D0,  CE*AMZW)
      CXWZ(2) = CONJG(CXWZ(1))
      CXWA(1) = DCMPLX(0.0D0, -CE*AMWB)
      CXWA(2) = CONJG(CXWA(1))

* SSVV
```

```
      CPPWW    = 0.5D0*CE2*GZ**2
      CPPZZ    = 0.5D0*CE2*(GZW*GZ)**2
      CPXWZ(1) = DCMPLX(0.0D0, 0.5D0*CE2*GZW)
      CPXWZ(2) = CONJG(CPXWZ(1))
      CPXWA(1) = DCMPLX(0.0D0, -0.5D0*CE2*GZ)
      CPXWA(2) = CONJG(CPXWA(1))
      C33WW    = 0.5D0*CE2*GZ**2
      C33ZZ    = 0.5D0*CE2*(GZW*GZ)**2
      C3XWZ(1) = 0.5D0*CE2*GZW
      C3XWZ(2) = CONJG(CPXWZ(1))
      C3XWA(1) = - 0.5D0*CE2*GZ
      C3XWA(2) = CONJG(CPXWA(1))
      CXXWW    = 0.5D0*CE2*GZ**2
      CXXZZ    = 0.5D0*CE2*(2*GW - GZ*GZW)**2
      CXXAA    = 2.0D0*CE2
      CXXAZ    = CE2*(2*GW - GZ*GZW)

* SSS
      GS     = 0.5D0*CE*AMPH2*GZ/AMWB
      CPXX   = -       GS
      CP33   = -       GS
      CPPP   = - 3.0D0*GS

* SSSS
      GS2    = GS*GS/AMPH2
      C3333  = - 3.0D0*GS2
      CPPPP  = - 3.0D0*GS2
      CXX33  = -       GS2
      CXXPP  = -       GS2
      CPP33  = -       GS2
      CXXXX  = - 2.0D0*GS2

* FFX
      GX     =   CE*GZ/(R2*AMWB)
       . . . . .
      CXEL(1,1)  = DCMPLX(0.0D0,  AMNE*GX)
      CXEL(2,1)  = DCMPLX(0.0D0, -AMEL*GX)
      CXEL(1,2)  = CONJG(CXEL(2,1))
      CXEL(2,2)  = CONJG(CXEL(1,1))
       . . . . .

* FFP
      G3     =   CE*GZ/(2*AMWB)
       . . . . .
      CPEL(1)    = - AMEL*G3
      CPEL(2)    = - AMEL*G3
       . . . . .

* FF3
       . . . . .
      C3EL(1)  = DCMPLX(0.0D0, - AMEL*G3)
      C3EL(2)  = - C3EL(1)
```

```
      . . . . .
*
* QCD coupling constant should be calculated in 'KINIT'.
      CQCD   = 1.0D0
      CQCDSQ = 1.0D0
      CQQG(1) = -1.0D0
      CQQG(2) = -1.0D0
*
      DO 100 I = 1, NGRAPH
        IGRAPH(I) = 0
        DO 100 J = 1, NGRAPH
          CF(J, I) = 1.0D0
  100 CONTINUE
      RETURN
      END
```

Source list 3.2 subprogram `AMPARM`

○ . . . . ○

Include file `incl1.f`

This file is prepared for passing the parameters for the amplitude calculation set in the subroutines `SETMAS` and `AMPARM` to the relevant subroutines through the several commons. In the source list 3.3.1 the structure of `incl1.f` for the process $e^+e^- \rightarrow W^+W^-\gamma$ is shown.

1) **Parameter statements**
   The parameters which define the sizes of arrays are given by the parameter statement. `LEPEXA`, `LEPEXW`, `LEPEXZ` and `LEPEXG` are the spin freedoms of external photon, W-boson, Z-boson and gluon, respectively. `LEPINA`, `LEPINW`, `LEPINZ` and `LEPING` are those for internal lines. `LEXTRN` and `LINTRN` are the spin freedoms for fermions of external and internal lines, respectively.

   The parameters `LOUTGO`, `LINCOM`, `LANTIP` and `LPRTCL` are just the input constants for the program package `CHANEL`.

2) **Table of amplitude**
   The calculated amplitudes for all graphs are stored in an array `AG( )`. An array `APROP( )` is used to keep the numerical value of the denominators of propagators.

   The arrays `AV( )`, `LT( )` and `INDEXG( )` in the common `/SMATBL/` are for temporary use.

4) **Coupling constants**
   The coupling constant for each type of vertex is in the common `/AMCPLC/`, which is defined in `AMPARM`.

5) **Four momenta of external particles**
The four momenta of external particles are given in the arrays `PEnnnn( )`, where the fourth components correspond to the energies. An array `PPROD(`i`,`j`)` gives the inner products of particle momenta i and j. They are derived in `KINEM` and copied to these arrays in `FUNC`.

6) `CHANEL` **inputs for the external particles**
The arrays `PSnnnn`, `EWnnnn`, `CEnnnn` and `EPnnnn` are the lists of light-like vectors, weight factors, phase factors and list of polarization vectors, respectively, which are defined in section 2.4.

o...o

```
      PARAMETER (LOUTGO =  2, LINCOM =  1)
      PARAMETER (LANTIP = -1, LPRTCL =  1)
      PARAMETER (LSCALR =  1)
      PARAMETER (LEPEXA =  2, LEPEXW =  3, LEPEXZ =  3, LEPEXG =  2)
      PARAMETER (LEPINA =  4, LEPINW =  4, LEPINZ =  4, LEPING =  3)
      PARAMETER (LEXTRN =  2, LINTRN =  4)
* Table of amplitudes
      PARAMETER (NGRAPH =   28, NEXTRN =   5, LAG =  72)
      PARAMETER (NGRPSQ = NGRAPH*NGRAPH)
      COMMON /AMSLCT/JSELG(NGRAPH), JGRAPH, JHIGGS, JWEAKB
      COMPLEX*16 AG, APROP
      COMMON /AMGRPH/AG(0:LAG-1,NGRAPH), APROP(NGRAPH),
     &               ANCP(NGRAPH), ANSP(0:NGRAPH),
     &               CF(NGRAPH,NGRAPH), IGRAPH(NGRAPH)

* Coupling constants
      COMMON /AMCPLC/CZWW       ,CAWW       ,CWWAA      ,CWWZA      ,
     &               CWWZZ      ,CWWWW      ,CWL   (2,2),CWEL  (2,2),
      ...............

* Momenta of external particles
      COMMON /AMEXTR/PE0001(4),PE0002(4),PE0003(4),PE0004(4),
     &               PE0005(4),
     &               PPROD(NEXTRN, NEXTRN)

* Switch of gauge parameters
      COMMON /SMGAUS/IGAU00,IGAUAB,IGAUWB,IGAUZB,IGAUGL
      COMMON /SMGAUG/AGAUGE(0:4)

* Normalization
      COMMON /SMDBGG/FKNORM,FKCALL,NKCALL

* Calculated table of amplitudes
      COMMON /SMATBL/AV, LT, INDEXG
      COMPLEX*16 AV(0:LAG-1)
      INTEGER    LT(0:NEXTRN), INDEXG(NEXTRN)
* For external particles
      COMMON /SMEXTP/
```

```
     &    PS0001, EW0001, CE0001,
     &    PS0002, EW0002, CE0002,
     &    EP0003, EW0003,
     &    EP0004, EW0004,
     &    EP0005, EW0005
      REAL*8     PS0001(4,2), EW0001(1)
      COMPLEX*16 CE0001(2,2)
      REAL*8     PS0002(4,2), EW0002(1)
      COMPLEX*16 CE0002(2,2)
      REAL*8     EP0003(4,LEPEXA), EW0003(LEPEXA)
      REAL*8     EP0004(4,LEPEXW), EW0004(LEPEXW)
      REAL*8     EP0005(4,LEPEXW), EW0005(LEPEXW)
```

Source list 3.3.1 Include file `incl1.f`

○ . . . . ○

Include file `incl2.f`

Here, working areas to be used in the computation of amplitudes are defined..

○ . . . . ○

```
      COMMON /AMWORK/ IDMM(       780)
      COMMON /AMWORI/ IDMI(        17)
```

Source list 3.3.2 Include file `incl2.f`

○ . . . . ○

Include file `inclk.f`

1) **Masses and widths**
   The variables in the commons `/AMMASS/` and `/AMGMMA/` are masses and widths of particles, respectively, which are defined in `SETMAS`.

2) **Physical constants**
   The variables in the common `/AMCNST/` , π and others, are defined in `SETMAS`.

3) **Double-valued case**
   The variable `MXREG` in the common `/AMCNST/` gives maximum multiplicity of the indegrand.

4) **External momenta**
   The variables in the common `/KMMASS/` stores external masses defined in `SETMAS` and they are expected to be referred in `KINEM` and `KINIT`.

○ . . . . ○

```
* Masses and width of particles
      COMMON /AMMASS/AMWB,AMZB,AMAB,AMXB,AMX3,AMPH,AMLU,AMNE,AMNM,AMNT,
     &               AMLD,AMEL,AMMU,AMTA,AMQU,AMUQ,AMCQ,AMTQ,AMQD,AMDQ,
     &               AMSQ,AMBQ,AMCP,AMCM,AMCZ,AMCA,AMGL,AMCG
      COMMON /AMGMMA/AGWB,AGZB,AGAB,AGXB,AGX3,AGPH,AGLU,AGNE,AGNM,AGNT,
     &               AGLD,AGEL,AGMU,AGTA,AGQU,AGUQ,AGCQ,AGTQ,AGQD,AGDQ,
     &               AGSQ,AGBQ,AGCP,AGCM,AGCZ,AGCA,AGGL,AGCG
      COMMON /AMCNST/ PI, PI2, RAD, GEVPB, ALPHA
      COMMON /AMREG / MXREG

*** Masses of external particles
      COMMON /KMMASS/AMASS1(   5), AMASS2(   5)
```

Source list 3.3.3 Include file `inclk.f`

○...○

3.2.2 Amplitude calculation

To calculate the numerical values of amplitudes, first the values of integration variables are translated into the four momenta of external particles, which is done by the subroutine KINEM. Then the subroutine AMPTBL is called to calculate the amplitudes.

Subroutine AMPTBL

The subroutine AMPTBL for the process $e^+e^- \to W^+W^-\gamma$ is shown in the source list 3.4, whose functions are as follows;

1) **External particles**
   At the beginning of AMPTBL all the information about the external fermions and vector bosons are prepared in suitable form for the calculation of vertices as shown in the source list 3.4. For the external fermion (vector boson) the subroutine SMEXTF (SMEXTV) is called for this purpose, whose specifications are given in section 7.2 in Ref.[8].

○...○

```
* FILE "amptbl.f" is generated by GRACE System (Minami-Tateya Group)
* Grace Version   1. 1          1994-Aug-19
**********************************************************************
      SUBROUTINE AMPTBL
** 5120     E+ E-  => W+  W-  A       TREE
      IMPLICIT REAL*8(A-H,O-Z)

      INCLUDE 'incl1.f'
      INCLUDE 'inclk.f'
      INCLUDE 'incl2.f'
*---------------------------------------------------------------------
      JGRAPH = 0
```

```
*   External lines
      CALL SMEXTF(LINCOM,AMEL,PE0001,PS0001,CE0001)
      EW0001(1) = LPRTCL
      CALL SMEXTF(LOUTGO,AMEL,PE0002,PS0002,CE0002)
      EW0002(1) = LANTIP
      CALL SMEXTV(LEPEXA,AMAB,PE0003,EP0003,EW0003,IGAUAB)
      CALL SMEXTV(LEPEXW,AMWB,PE0004,EP0004,EW0004,IGAUWB)
      CALL SMEXTV(LEPEXW,AMWB,PE0005,EP0005,EW0005,IGAUWB)


* Graph NO.    1 -    1 (    1)
      IF (JWEAKB.NE.0) THEN
      IF (JSELG(    1).NE.0) THEN

      JGRAPH = JGRAPH + 1
      IGRAPH(JGRAPH) =    1
      CALL AM0001
      ENDIF
      ENDIF

      .............

* Graph NO.   28 -    1 (   28)
      IF (JWEAKB.NE.0) THEN
      IF (JSELG(   28).NE.0) THEN

      JGRAPH = JGRAPH + 1
      IGRAPH(JGRAPH) =   28
      CALL AM0028
      ENDIF
      ENDIF
      RETURN
      END
```

Source list 3.4 Example of subroutine AMPTBL

○..○

The variables LEPEXW and LEPEXA represent the spin freedoms of external W-bosons and photon, respectively, and are set in the include file incl1.f by the parameter statement as shown in the source list 3.3.1. For the fermion the variable EWnnnn(1) is set equal to "1" for particle or "-1" for anti-particle. In this example, EW0001(1) is set equal to "1" (electron) and EW0002(1) to "-1" (positron).

2) **Calculation of each amplitude**
The subroutine AMnnnn is called to calculate the nnnn-th graph. Since there are 28 graphs in the process $e^+e^- \to W^+W^-\gamma$, there are 28 subroutines from AM0001

to AM0028. The flag JSELG(i) is used for selecting the graph. If it is set equal to "zero" in the subroutine SETMAS the corresponding i-th graph is not included in the calculation. This flag is to be set by the user for the time being, but it will be implemented in near future.

Subroutine AMnnnn

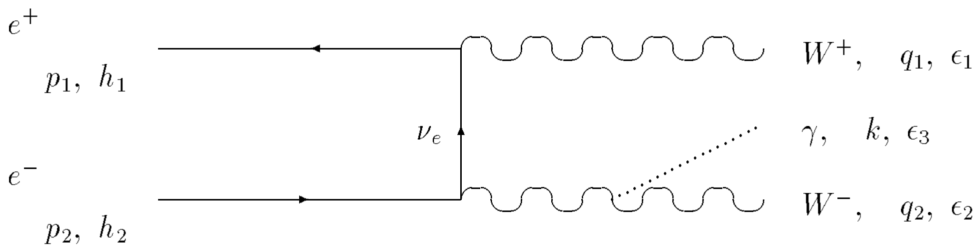

Figure 3.3: A Feynman graph for the process $e^+e^- \to W^+W^-\gamma$.

A main part of amplitude calculation appears in subroutines AMnnnns. To describe the amplitude generation (see section 2.4 in Ref.[8]), we take a Feynman graph as an example in the process $e^+e^- \to W^+W^-\gamma$ shown in figure 3.3. The corresponding subroutine to the graph is AM0020, whose compositions are as follows;

1) **Internal momenta**
The internal momenta PE0141 and PE0143 are calculated from the external momenta, which correspond to those of internal neutrino and W-boson, respectively.

2) **Propagators**
The product of denominators of propagators is calculated by the subroutine SMPRPD, where the inputs are the momentum transfer, mass square and mass times width.

   The numerator of each propagator is handled by the subroutines SMINTF and SMINTV for the internal neutrino and W-boson, respectively.

3) **Vertices**
Numerical values of vertex amplitudes are calculated by subroutines SMFFV and SMVVV. By SMFFV the vertices $\bar{\nu}_e e^- W^+$ and $e^+\nu_e W^-$ are calculated and for the vertex $W^-W^+\gamma$ subroutine SMVVV is used. The calculated amplitudes of vertices $\bar{\nu}_e e^- W^+$ and $e^+\nu_e W^-$ and $W^-W^+\gamma$ are saved in the arrays AV0092, AV0093 and AV0094, respectively.

4) **Connection of vertices**
First the vertices $\bar{\nu}_e e^- W^+$ and $e^+\nu_e W^-$ are connected by the routine SMCONF, where the amplitudes AV0092 and AV0093 are combined by summing over all the

possible helicity states of the internal neutrino with weight EW0141. The resultant amplitude is stored in an array AV0095.

Second the resultant amplitude AV0095 and $W^-W^+\gamma$ amplitude AV0094 are connected by taking summation over all the possible polarization states of internal W-boson with weight EW0143 using the routine SMCONV. The total amplitude is saved in an array AV.

5) **Rearrange the internal structure of amplitude**
In order to sum up all amplitudes, they have to have the same internal structure. However, the internal structure of amplitude AV does strongly depend upon the order of constructing the amplitude, which may be different graph by graph. A subroutine AMPORD is used to change the amplitude AV in an individual structure into the amplitude AG in an unified structure.

○...○

```
* FILE "am0020.f" is generated by GRACE System (Minami-Tateya Group)
* Grace Version   1. 1          1994-Aug-19
* Graph No.   20 -   1 (  20)
************************************************************************
      SUBROUTINE am0020
      IMPLICIT REAL*8(A-H,O-Z)

      INCLUDE 'incl1.f'
      INCLUDE 'inclk.f'
*-----------------------------------------------------------------------
      COMMON /AMWORK/
     &  PE0141, EW0141, PS0141, VM0141, CE0141,
     &  PE0143, EP0143, EW0143, VM0143,
     &  AV0092,AV0093,AV0094,AV0095
      COMMON /AMWORI/
     &  LT0092,LT0093,LT0094,LT0095
*        2528+        68 bytes used.

      REAL*8     PE0141(4), EW0141(2), PS0141(4,3), VM0141
      COMPLEX*16 CE0141(2,4)
      REAL*8     PE0143(4), EP0143(4,LEPINW), EW0143(LEPINW), VM0143
      INTEGER    LT0092(0:3)
      COMPLEX*16 AV0092(0:LINTRN*LEXTRN*LEPEXW-1)
      INTEGER    LT0093(0:3)
      COMPLEX*16 AV0093(0:LEXTRN*LINTRN*LEPINW-1)
      INTEGER    LT0094(0:3)
      COMPLEX*16 AV0094(0:LEPINW*LEPEXW*LEPEXA-1)
      INTEGER    LT0095(0:   4)
      COMPLEX*16 AV0095(0:LEXTRN*LEPINW*LEXTRN*LEPEXW-1)

*-----------------------------------------------------------------------
*  Internal momenta
```

```
      DO   100  I = 1, 4
        PE0141(I) =  -PE0002(I) +PE0004(I)
        PE0143(I) =  -PE0003(I) -PE0005(I)
  100 CONTINUE


      APROP(JGRAPH) = 1.0D0
      VM0141 = - 2.0D0*PPROD(  2,  4) +  1.0D0*AMWB**2 +  1.0D0*AMEL**2
      CALL SMPRPD(APROP(JGRAPH),VM0141,AMNE**2,AMNE*AGNE)
      VM0143 = + 2.0D0*PPROD(  3,  5) +  1.0D0*AMWB**2 +  1.0D0*AMAB**2
      CALL SMPRPD(APROP(JGRAPH),VM0143,AMWB**2,AMWB*AGWB)

*   Internal lines
      CALL SMINTF(AMNE,PE0141,VM0141,EW0141,PS0141,CE0141)
      CALL SMINTV(LEPINW,AMWB,PE0143,EP0143,EW0143,VM0143,IGAUWB)

*   Vertices
      CALL SMFFV(LINTRN,LEXTRN,LEPEXW,EW0141,EW0002,AMNE,AMEL,
     &           CWEL  (1,2),CE0141,CE0002,PS0141,PS0002,EP0004
     &           ,LT0092,AV0092)
      CALL SMFFV(LEXTRN,LINTRN,LEPINW,EW0001,EW0141,AMEL,AMNE,
     &           CWEL  (1,1),CE0001,CE0141,PS0001,PS0141,EP0143
     &           ,LT0093,AV0093)
      CALL SMVVV(LEPINW,LEPEXW,LEPEXA,-1,-1,-1,CAWW  ,PE0143,PE0005,
     &           PE0003,EP0143,EP0005,EP0003,LT0094,AV0094)

*   Connect vertices.
      CALL SMCONF(LT0093,LT0092,   2,   1,EW0141,AV0093,AV0092,
     &            LT0095,AV0095)
      CALL SMCONV(LT0094,LT0095,   1,   2,EW0143,AV0094,AV0095,
     &            LT,AV)
      APROP(JGRAPH) = +1.0D0/APROP(JGRAPH)
      INDEXG(   1) =    5
      INDEXG(   2) =    3
      INDEXG(   3) =    1
      INDEXG(   4) =    2
      INDEXG(   5) =    4
      CALL AMPORD(LT, AV, INDEXG, AG(0,JGRAPH))

      RETURN
      END
```

Source list 3.5 Subroutine AM0020 for $e^+e^- \to W^+W^-\gamma$

○ .... ○

3.3 Specification of the kinematics routines

In general the choice of integration variables is highly dependent on the structure of singularities in the amplitude squared, such as infrared divergence, mass-singularity and t-channel photon exchange. It is quite difficult to prepare a kinematics enough general for any process. The user can write the kinematics most appropriate for the process to be calculated based on the following description. The subroutines related to this work are as follows:

`KINIT`	Initialization of kinematics.
`KINEM`	Calculate four-momenta of final particles from integration variables.

Convention used in `GRACE` for the 4-momentum is that it is stored in an array in the order of p_x, p_y, p_z, E.

3.3.1 Subroutine `USERIN`

After the initialization of `BASES/SPRING`, i.e., call of `BSINIT` for `BASES` or call of `BSINIT` and `BSREAD` for `SPRING`, the subroutine `USERIN` is called for the initialization of amplitude calculation. The subroutine `USERIN` calls `SETMAS` and `AMPARM`.

3.3.2 Subroutine `KINIT`

`KINIT` makes the initialization of the kinematics and is called by `USERIN`.

In the source list 3.6 the subroutine `KINIT` for the process $e^+e^- \to W^+W^-\gamma$ is shown. It is generated by `KINEM 3002`. Its program structure is shown below. Among these items, items 5) and 6) are explained in detail related to the integration stage(sectio 3.5.2).

1) Give center-of-mass energy. Here, `S` and `W` are stored in the common area `/KINAM1/`.

2) Define various cutoff parameters.

3) Print values for 1) and 2), since they might be changed by the user by editing.

4) Prepare some variables which are referred in `KINEM`. Here, `FACT` in the common area `/KINAM1/` is defined.

5) Initialization of BASES parameters. Set the parameters for `BASES`. These parameters are transmitted to `BASES` through the commons `/BPARM1/` and `/BPARM2/`.

6) Initialize histogram by calling `XHINIT`.

7) Set `MXREG` which represents the maximum multiplicity in case of malti-valued kinematics. For simple kinematics, it is 1.

○...○

```
      SUBROUTINE KINIT
C-----------------------------------------------------------------------
C   GRACE System Library File
C   KINEM No. : 3002
C   Date      : 1994.04.30
C   Author    : Y.Kurihara
C-----------------------------------------------------------------------
C
      IMPLICIT REAL*8(A-H,O-Z)

      PARAMETER ( MXDIM = 50 )
      COMMON / LOOP0 / LOOP
      COMMON / BPARM1 / XL(MXDIM),XU(MXDIM),NDIM,NWILD,
     &                 IG(MXDIM),NCALL
      COMMON / BPARM2 / ACC1,ACC2,ITMX1,ITMX2
      COMMON / BASE3 / SI,SI2,SWGT,SCHI,SCALLS,ATACC,NSU,IT,WGT

      CHARACTER XSTR*14

      INCLUDE "inclk.f"
*-----------------------------------------------------------------------
      COMMON/KMCNTL/IRESNS,ICOS3
      COMMON/KINEM1/S,W,FACT
      COMMON/CUT001/COSCUT(2,3),ENGYCT(2,3),AMASCT(2),ARESNS(2)
*-----------------------------------------------------------------------
* 3-body kinematics
*
*  Integration variables
*       COS(the3),Phi3(=0)
*       Q2=(p4+p5)**2
*       COS(the4),Phi4    in particle 4,5 CM frame
*
*                         /P4
*                        /
*           p1\         /
*              \      q/\
*               ***** /  \
*              *******    \P5
*               ***** \
*               /      \
*              /        \
*           p2/          \p3
*
*
*--- 1. Initialize constants for kinematics.
* S = (P1+P2)**2
* W = SQRT(S)
      W =1000.D0
      S = W*W
      totmas=amass1(3)+amass1(4)+amass1(5)
```

```
      if(totmas.gt.w) then
write(6,*)
     .' CM energy is less than the sum of final particle masses.'
stop
      end if
*******************************
* Angular cuts in Lab-frame.  *
*******************************
* particle 3
* minimum cos  cut
      COSCUT(1,1)= -1
* maximum cos  cut
      COSCUT(2,1)=  1
* particle 4
* minimum cos  cut
      COSCUT(1,2)= -1
* maximum cos  cut
      COSCUT(2,2)=  1
* particle 5
* minimum cos  cut
      COSCUT(1,3)= -1
* maximum cos  cut
      COSCUT(2,3)=  1

*******************************
* Energy  cuts in Lab-frame.  *
*******************************
* particle 3
* minimum energy cut
      ENGYCT(1,1) = AMASS1(3)
      ENGYCT(1,1) = 0.1d-2
* maximum energy cut
      ENGYCT(2,1) = W
* particle 4
* minimum energy cut
      ENGYCT(1,2) = AMASS1(4)
* maximum energy cut
      ENGYCT(2,2) = W
* particle 5
* minimum energy cut
      ENGYCT(1,3) = AMASS1(5)
* maximum energy cut
      ENGYCT(2,3) = W

*******************************
* Cut on invariant mass of 4-5*
*******************************
* minimum
      AMASCT(1)= AMASS1(4)+AMASS1(5)
* maximum
      AMASCT(2)= W-AMASS1(3)
```

```
******************************
* Q2 singularity treatment    *
******************************
* no-singularity           : IRESNS= 0
* narrow resonance         : IRESNS= 1
* 1/Q2 singularity         : IRESNS=-1
      IRESNS   = 0
      IF(IRESNS.EQ.1) THEN
C If you want treat narrow resonance, you should set resonance mass
C and width.
C       ARESNS(1)=AM**
C       ARESNS(2)=AG**
        ARESNS(1)=0
        ARESNS(2)=0
      END IF

********************************
* mass singularity treatment    *
********************************
* no-singularity           : ICOS3=0
* mass-singularity         : ICOS3=1
*
*  If ICOS3=1, a particle 3 is assumed to be an initial state
* radiated photon (amass1(3)=0) and particle1 and particle2 have
* the same mass (AMASS1(1)=AMASS1(2)>0), and IRESNS is ignored.
*
      ICOS3=1
      if(icos3.eq.1 .and. amass1(3).gt.1.d-20) then
       write(6,*)'If ICOS3=1, particle 3 should be photon.'
       stop
      endif

      WRITE(6,*)'*'
      WRITE(6,*)'* Kinematics initialization '
      WRITE(6,*)'* CM Energy ',W
      WRITE(6,*)'* COSCUT    '
      WRITE(6,'(2F10.3)')COSCUT
      WRITE(6,*)'* ENGYCT    '
      WRITE(6,'(2F10.3)')ENGYCT
      WRITE(6,*)'* AMASCT    '
      WRITE(6,'(2F10.3)')AMASCT
      WRITE(6,*)'* Singularity treatment '
      if(icos3.eq.1) goto 10
      WRITE(6,*)'*   Q2'
      IF     (IRESNS.EQ.-1) THEN
         WRITE(6,*)'* 1/Q2 singularity '
      ELSE IF(IRESNS.EQ. 0) THEN
         WRITE(6,*)'* NO   singularity '
      ELSE IF(IRESNS.EQ. 1) THEN
         WRITE(6,*)'* Narrow resonance '
         WRITE(6,*)'*  resonance mass and width'
         WRITE(6,*) ARESNS
```

```
      END IF
10    IF     (ICOS3.EQ.0) THEN
         WRITE(6,*)'* NO mass-singularity '
      ELSE
         WRITE(6,*)'*    mass-singularity '
      END IF
      WRITE(6,*)'*'

* Following flux factor is for a particle-antiparticle collision
* at s/m**2 >> 1. For an e-gamma collision or low energy interactions
* you should use appropriate formulae.
      VREL = 2
      FLUX = VREL*S
      FACT = GEVPB/FLUX

*----- BASES RELATED INITIALIZATIONS ----------------------------------
*--- 2. Dimension of integration variables.

          NDIM =   5
          NWILD=   4

C-----------------------------------------------------------------------
*--- 3. Region of integration.

        DO 1 I=1,NDIM
         XL(I)  =   0.D0
         XU(I)  =   1.D0
         IG(I)  =   1
1       CONTINUE
C-----------------------------------------------------------------------
*--- 4. Number of iterations
      ITMX1  =  5
      ITMX2  =  5

      NCALL  = 50000

*--- 5. Set MXREG : the maximum number of values which are returned
*       by FUNC for one phase space point

      MXREG =   1

*--- 6. Set histograms
      NX = 50
      ND = 50

      DO 100 I = 1, NDIM
         WRITE(XSTR, 110) I
  110    FORMAT('X(',I2,') SPECTRUM')
         CALL XHINIT(I, XL(I), XU(I), NX, XSTR)
  100 CONTINUE
         CALL XHINIT(ndim+1, 0.d0, w, NX, 'Energy of Particle 3')
         CALL XHINIT(ndim+2, 0.d0, w, NX, 'Energy of Particle 4')
```

```
      CALL XHINIT(ndim+3, 0.d0, w, NX, 'Energy of Particle 5')
      CALL XHINIT(ndim+4,-1.d0,1.d0, NX, 'cos_the of Particle 3')
      CALL XHINIT(ndim+5,-1.d0,1.d0, NX, 'cos_the of Particle 4')
      CALL XHINIT(ndim+6,-1.d0,1.d0, NX, 'cos_the of Particle 5')

   RETURN
   END
```

Source list 3.6 `KINIT` for the process $e^+e^- \rightarrow W^+W^-\gamma$

o..o

3.3.3 Subroutine KINEM

In order to integrate the differential cross section, BASES samples a point in the integration volume and calls the function subprogram FUNC, which calculates the numerical value of integrand at the sampling point and returns it as the value of function. For calculating the differential cross section integration variables are to be translated into four-momenta of external particles, which is done by the subprogram KINEM. Its program structure is as follows:

○...○

```
SUBROUTINE KINEM( NEXTRN, X, PE, PP, YACOB, NREG, IREG, JUMP )

IMPLICIT REAL*8
PARAMETER (MXDIM = 50)
INTEGER    NEXTRN,    NREG,         IREG,                JUMP
REAL*8     X(MXDIM), PE(4,NEXTRN), PP(NEXTRN,NEXTRN), YACOB
```

○...○

The meanings of the arguments are as follows;

NEXTRN	input	Number of external particles
X	input	Values of integration variables at the sampling point
PE	output	Table of four momenta of external particles
PP	output	Table of inner products of four momenta
YACOB	output	Normalization for converting the square of amplitude to the cross section
NREG	in/out	Control flag in case of multi-valued kinematics. See 5) below.
IREG	input	Control flag in case of multi-valued kinematics. See 5) below.
JUMP	output	Flag for the acceptance of the sampling point. For accepted sampling, it returns 0. If the sampling point is out of kinematical boundary, JUMP is set to be a non zero integer value.

An example of KINEM for the process $e^+e^- \to W^+W^-\gamma$ is shown in the source list 3.7. It is generated by KINEM 3002. The specification for writing KINEM is as follows:

1) **Initialization**
   At the beginning of routine, variable JUMP is 0.

2) **Calculation of four vectors of external particles**
   From the integration variables X(i), four vectors of external particles are derived and are stored in the arrays PE(1~4,k), where PE(1,k), PE(2,k) and PE(3,k) correspond to p_x, p_y and p_z, respectively, and PE(4,k) is energy E of the k-th particle.

3) **Jump flag** JUMP
During calculation of four vector, when the sampling point X(i) in the integration volume is out of the kinematical boundary, then the jump flag JUMP should be set equal to non-zero integer value. Otherwise, it must be kept zero.

4) **Inner products of four momenta**
The inner products, taking all combinations of the external four momenta, are calculated and stored in the array PP(l,m), where the numbers l and m are corresponding to the labels of momenta PE(1∼4,l) and PE(1∼4,m), respectively. Namely,

$$\begin{aligned} \texttt{PP}(l,m) &= \texttt{PE}(4,l) * \texttt{PE}(4,m) - \texttt{PE}(1,l) * \texttt{PE}(1,m) \\ &\quad - \texttt{PE}(2,l) * \texttt{PE}(2,m) - \texttt{PE}(3,l) * \texttt{PE}(3,m). \end{aligned}$$

5) **Control of multi-valued kinematics**
When the kinematics is represented by a many valued function, namely, a sampling point in the integration volume corresponds to several points in the momentum phase space, the control is done by variables NREG and IREG. This is also related to the function FUNC in section 3.5.3.

NREG is the multiplicity in the momentum space which corresponds to a point in the integration volume. This value is set equal to one by FUNC for the first call at each sampling point.

IREG is a counter of calling KINEM at the same sampling point. Function FUNC increments IREG for each call, and calls KINEM while IREG $\leq$ NREG.

In the subprogram FUNC, the variables NREG and IREG are set to "1" at the beginning and subroutine KINEM is called.

The structure of KINEM for the multi-valued case is as follows:

```
SUBROUTINE KINEM(NEXTERN, XX, P, PP, YACOB, NREG, IREG, JUMP)
...
IF(IREG.EQ.1) THEN
    NREG = (the number of multiplicity at the sampling point XX)
    (Calculate four momenta P for the first calculation)
    (Calculate inner products of four momenta PP)
    (Calculate Jacobian YACOB for the first calculation)
    ...
ELSE IF(IREG.EQ.2) THEN
    (Calculate four momenta P for the second calculation)
    (Calculate inner products of four momenta PP)
    (Calculate Jacobian YACOB for the first calculation)
    ...
```

```
      ELSE IF ...
          ...
      ENDIF
      RETURN
      END
```

KINEM calculates the total number of multiplicity at the sampling point and store it in NREG. If it is greater than "1", then the first calculation of four momenta is performed. From the second calculation, IREG is incremented with keeping NREG unchanged and momenta are returned by calling KINEM. The same step is repeated until IREG reaches NREG. The value of MXREG is defined in the subroutine KINIT and is used to protect unexpected repeat. It is clear that NREG is the total number of multiplicity at a sampling point given by KINEM and IREG plays the role of counter which shows the number of KINEM calls.

In the example of 3.7 there is no such a structure, because the kinematics is constructed by a single valued function.

6) **Notice**
One should be careful not to loose the numerical accuracy by the cancellation over many digits which may take place when the inner-product PP are calculated from the four components of momenta. Use of invariants is recommended.

◦ . . . . ◦

```
      SUBROUTINE KINEM(NEXTRN,X,PE,PP,YACOB,NREG,IREG,JUMP)
C----------------------------------------------------------------------
C   GRACE System Library File
C   KINEM No. : 3002
C   Date      : 1994.04.30
C   Author    : Y.Kurihara
C----------------------------------------------------------------------
      IMPLICIT REAL* 8(A-H,O-Z)
************************************************************************
      INTEGER NEXTRN
      PARAMETER ( MXDIM = 50 )

      COMMON / BASE1 / XL(MXDIM),XU(MXDIM),NDIM
      REAL*8  X(MXDIM)
      REAL*8  PE(4,NEXTRN), PP(NEXTRN,NEXTRN)
      REAL*8  YACOB
      INTEGER NREG, IREG
      INTEGER JUMP

      INCLUDE "inclk.f"
*---------------------------------------------------------------------
      COMMON/KMCNTL/IRESNS,ICOS3
      COMMON/KINEM1/S,W,FACT
      COMMON/CUT001/COSCUT(2,3),ENGYCT(2,3),AMASCT(2),ARESNS(2)
```

```
      DIMENSION PBSTCL(4)
* BETA FUNCTION
       BETA(Z1,Z2)=SQRT(1-2*(Z1+Z2)+(Z1-Z2)**2)
*-----------------------------------------------------------------------
      NREG   =   1
      JUMP   =   0
      YACOB  =   0.D0
      AJACOB=    1
* Set integration variables
************
* PHI3     *
************
      IF(NDIM.EQ.5) THEN
       PHI3=X(5)*2*PI
      ELSE
       PHI3=0
      END IF
                 AJACOB = AJACOB*2*PI
      if(icos3.eq.0)then
*************
* Q2        *
*************
      Q2MIN=MAX((AMASS1(4)+AMASS1(5))**2   , AMASCT(1)**2)
      Q2MAX=MIN(S+AMASS2(3)-2*W*ENGYCT(1,1), AMASCT(2)**2)
      IF(Q2MIN.GE.Q2MAX) GOTO 999
      IF     (IRESNS.EQ. 0) THEN
       Q2  = Q2MIN+(Q2MAX-Q2MIN)*X(4)
                 AJACOB = AJACOB*(Q2MAX-Q2MIN)
      ELSE IF(IRESNS.EQ.-1) THEN
       Q2  = Q2MIN*(Q2MAX/Q2MIN)**X(4)
                 AJACOB = AJACOB*Q2*LOG(Q2MAX/Q2MIN)
      ELSE IF(IRESNS.EQ. 1) THEN
       AM    = ARESNS(1)
       AM2   =AM*AM
       AMG   =ARESNS(1)*ARESNS(2)
       THEMIN=ATAN((Q2MIN-AM2)/AMG)
       THEMAX=ATAN((Q2MAX-AM2)/AMG)
       THE   =THEMIN+(THEMAX-THEMIN)*X(4)
       Q2    =AMG*TAN(THE)+AM2
                 AJACOB=AJACOB*(THEMAX-THEMIN)
     .*                 ((Q2-AM2)**2+AMG**2)/AMG
      ELSE
       GOTO 999
      END IF
*************
* ENERGY3   *
*************
       E3   = max(0.d0,(S+AMASS2(3)-Q2)/2.D0/W)
       IF(E3.GT.ENGYCT(2,1)) GOTO 999
       IF(E3.LT.ENGYCT(1,1)) GOTO 999
       P3   = SQRT( (E3-AMASS1(3))*(E3+AMASS1(3)) )
       else if(icos3.eq.1) then
```

```
*************
* ENERGY3   *
*************
       e3min=MAX(AMASS1(3),ENGYCT(1,1)
     . ,(S+AMASS2(3)-AMASCT(2)**2)/2.D0/W)
       E3MAX=MIN(ENGYCT(2,1)
     . ,(S+AMASS2(3)-(AMASS1(4)+AMASS1(5))**2)/2.D0/W
     . ,(S+AMASS2(3)-AMASCT(1)**2)/2.D0/W)
       E3=E3MIN*(E3MAX/E3MIN)**X(4)
                 AJACOB = AJACOB*E3*log(E3max/E3min)*2.D0*W
       P3   = SQRT( (E3-AMASS1(3))*(E3+AMASS1(3)) )
*************
* Q2        *
*************
       q2=s+amass2(3)-2*w*e3
       ELSE
       GOTO 999
       end if
*************
* ENERGY1   *
*************
       E1   = (S+AMASS2(1)-AMASS2(2))/2.D0/W
       P1   = SQRT( (E1-AMASS1(1))*(E1+AMASS1(1)) )
*************
* ENERGY2   *
*************
       E2   = (S+AMASS2(2)-AMASS2(1))/2.D0/W
*************
* COS(the3) *
*************
      IF     (ICOS3.EQ.0) THEN
       COS3=COSCUT(1,1) + (COSCUT(2,1)-COSCUT(1,1))*X(3)
       SIN3=SQRT( (1-COS3)*(1+COS3) )
       D1  = 2.D0*(E1*E3-P1*P3*COS3)
       D2  = 2.D0*(E2*E3+P1*P3*COS3)
                 AJACOB = AJACOB*(COSCUT(2,1)-COSCUT(1,1))
      ELSE IF(ICOS3.EQ.1) THEN
       XI   = (E1+P1)/AMASS1(1)
       ABDB = 1+AMASS2(1)/P1/(E1+P1)
       TEMAX= (ABDB+COSCUT(2,1))/(ABDB-COSCUT(2,1))
       TEMIN= (ABDB+COSCUT(1,1))/(ABDB-COSCUT(1,1))
       YMAX = (LOG(TEMAX)/LOG(XI)+2)/4.D0
       YMIN = (LOG(TEMIN)/LOG(XI)+2)/4.D0
       if(coscut(1,1).lt.-1.d0+1.d-10)ymin=0
       if(coscut(2,1).gt. 1.d0-1.d-10)ymax=1
       Y    = YMIN+(YMAX-YMIN)*X(3)
                 AJACOB = AJACOB*(YMAX-YMIN)
       TETA = XI**(4*Y-2)
       COS3 = ABDB*(TETA-1)/(TETA+1)
c      IF(ABS(COS3).GT.1) COS3=SIGN(1.D0,COS3)
       SIN3 = SQRT( (1-COS3)*(1+COS3) )
       D1   = 4*E3*E1/(TETA+1)
```

```
       D2   = TETA*D1
       D13  = 4   *E1/(TETA+1)
       D23  = TETA*D13
                 AJACOB = AJACOB*LOG(XI)/2.D0/E1/P1
     .*                   D13*D23
      ELSE
       GOTO 999
      END IF
************************
* PHI4 in 4-5 CM frame *
************************
      PHI4=X(2)*2*PI+PHI3
                 AJACOB = AJACOB*2*PI

************************
* COS4 in 4-5 CM frame *
************************
      COS4=-1+2*X(1)
      SIN4=SQRT( (1-COS4)*(1+COS4) )
                 AJACOB = AJACOB*2

* Set four-vectors
*Particle 1
      PE(1,1) = 0
      PE(2,1) = 0
      PE(3,1) = P1
      PE(4,1) = E1

*Particle 2
      PE(1,2) = 0
      PE(2,2) = 0
      PE(3,2) =-P1
      PE(4,2) = E2

*Particle 3
      PE(1,3) = P3*SIN3*COS(PHI3)
      PE(2,3) = P3*SIN3*SIN(PHI3)
      PE(3,3) = P3*COS3
      PE(4,3) = E3

*Particle 4 in 4-5 CM frame
      E4      = (Q2+AMASS2(4)-AMASS2(5))/2.D0/SQRT(Q2)
      P4      = SQRT( (E4-AMASS1(4))*(E4+AMASS1(4)) )
      PE(1,4) = P4*SIN4*COS(PHI4)
      PE(2,4) = P4*SIN4*SIN(PHI4)
      PE(3,4) = P4*COS4
      PE(4,4) = E4

*Particle 5 in 4-5 CM frame
      E5      = (Q2+AMASS2(5)-AMASS2(4))/2.D0/SQRT(Q2)
      PE(1,5) =-PE(1,4)
      PE(2,5) =-PE(2,4)
```

```
      PE(3,5) =-PE(3,4)
      PE(4,5) = E5

* boost from 4-5 CM frame to 1-2 CM frame (Lab-frame)
      PBSTCL(1)=-PE(1,3)
      PBSTCL(2)=-PE(2,3)
      PBSTCL(3)=-PE(3,3)
      PBSTCL(4)= SQRT(P3*P3+Q2)
      CALL WTOLAB(PE(1,4),PE(1,5),PBSTCL, PE(1,4),PE(1,5))
c     call pboost(pe(1,4),pbstcl, pe(1,4))
c     call pboost(pe(1,5),pbstcl, pe(1,5))
* cuts in Lab-frame
      COS4LB=PE(3,4)/SQRT(PE(1,4)**2+PE(2,4)**2+PE(3,4)**2)
      IF(COS4LB.LT.COSCUT(1,2) .OR. COS4LB.GT.COSCUT(2,2)) GOTO 999
      COS5LB=PE(3,5)/SQRT(PE(1,5)**2+PE(2,5)**2+PE(3,5)**2)
      IF(COS5LB.LT.COSCUT(1,3) .OR. COS5LB.GT.COSCUT(2,3)) GOTO 999
      IF(PE(4,4).LT.ENGYCT(1,2).OR.PE(4,4).GT.ENGYCT(2,2)) GOTO 999
      IF(PE(4,5).LT.ENGYCT(1,3).OR.PE(4,5).GT.ENGYCT(2,3)) GOTO 999

*Set invariants
      PP(1,1) = AMASS2(1)
      PP(1,2) = (S-AMASS2(1)-AMASS2(2))/2.D0
      PP(1,3) = D1/2.D0
      PP(1,4) = PE(4,1)*PE(4,4)-PE(1,1)*PE(1,4)
     .         -PE(2,1)*PE(2,4)-PE(3,1)*PE(3,4)
      PP(1,5) = PE(4,1)*PE(4,5)-PE(1,1)*PE(1,5)
     .         -PE(2,1)*PE(2,5)-PE(3,1)*PE(3,5)

      PP(2,1) = PP(1,2)
      PP(2,2) = AMASS2(2)
      PP(2,3) = D2/2.D0
      PP(2,4) = PE(4,2)*PE(4,4)-PE(1,2)*PE(1,4)
     .         -PE(2,2)*PE(2,4)-PE(3,2)*PE(3,4)
      PP(2,5) = PE(4,2)*PE(4,5)-PE(1,2)*PE(1,5)
     .         -PE(2,2)*PE(2,5)-PE(3,2)*PE(3,5)

      PP(3,1) = PP(1,3)
      PP(3,2) = PP(2,3)
      PP(3,3) = AMASS2(3)
      PP(3,4) = PE(4,3)*PE(4,4)-PE(1,3)*PE(1,4)
     .         -PE(2,3)*PE(2,4)-PE(3,3)*PE(3,4)
      PP(3,5) = PE(4,3)*PE(4,5)-PE(1,3)*PE(1,5)
     .         -PE(2,3)*PE(2,5)-PE(3,3)*PE(3,5)

      PP(4,1) = PP(1,4)
      PP(4,2) = PP(2,4)
      PP(4,3) = PP(3,4)
      PP(4,4) = AMASS2(4)
      PP(4,5) = (Q2-AMASS2(4)-AMASS2(5))/2.D0

      PP(5,1) = PP(1,5)
      PP(5,2) = PP(2,5)
```

```
      PP(5,3) = PP(3,5)
      PP(5,4) = PP(4,5)
      PP(5,5) = AMASS2(5)
*Set jacobian
      YACOB=FACT*AJACOB
     ./(2*PI)/(32*PI2)/(32*PI2)
     .*BETA(Q2       /S ,AMASS2(3)/S )
     .*BETA(AMASS2(4)/Q2,AMASS2(5)/Q2)

c     write(6,*)pe(4,1)+pe(4,2)
c     write(6,*)pe(4,3)+pe(4,4)+pe(4,5)
c     write(6,*)pe(1,3)+pe(1,4)+pe(1,5)
c     write(6,*)pe(2,3)+pe(2,4)+pe(2,5)
c     write(6,*)pe(3,3)+pe(3,4)+pe(3,5)
c     write(6,*)amass1
c     write(6,*)amass2
c     write(6,*)pbstcl
c     write(6,*)sqrt(pe(4,3)**2-pe(1,3)**2-pe(2,3)**2-pe(3,3))
c     write(6,*)sqrt(pe(4,4)**2-pe(1,4)**2-pe(2,4)**2-pe(3,4))
c     write(6,*)sqrt(pe(4,5)**2-pe(1,5)**2-pe(2,5)**2-pe(3,5))
      RETURN
999   CONTINUE
      JUMP=1
      YACOB=0
      RETURN
      END
```

Source list 3.7 An example of `KINEM`

○ . . . . ○

3.4 Test of generated source code

The main program `gauge.f` to check gauge invariance at one point in the integration volume is produced by `GRACE`.

After the execution, as shown in the output 3.1, the squared values of the amplitude at a point in the phase space in different gauges, covariant gauge and unitary gauge are shown, which is the output of the test program. The number of calculated Feynman graphs is different in these gauges. The relative error is printed in the output. We usually require about 14 digits agreement in double precision and about 32 digits in quadruple precision [2]. In the output one can also see contribution of each graph to the result. [3]

```
                    GRACE   Ver.   1.0


          * 5120     E+ E-  => W+  W-  A       TREE


   W =  1000.00     EM =   .511E-03    IN GEV
   RMN =  .100E-02


 (1) KINEMATICAL CUTS :

                ETH =   80.0        GEV
               CSMX =    1.000
               CSMN =   -1.000

      WHERE  ETH = THRESHOLD ENERGY FOR Q20 AND Q10
             CSMN AND CSMX ARE ANGLE CUT FOR CS AND CSQ


 X      =  .45 .45 .45 .45
 JSELG =  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 IGAUAB =  0 AGAUGE =  1.000000000000000E+20
 IGAUWB =  0 AGAUGE =  1.000000000000000E+20
 IGAUZB =  0 AGAUGE =  1.000000000000000E+20
 IGAUGL =  0 AGAUGE =  1.000000000000000E+20
   ANS1   =  1.34979414428365
 # GRAPHS =  18
 X      =  .45 .45 .45 .45
 JSELG =  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 IGAUAB =  1 AGAUGE =  2.0
 IGAUWB =  2 AGAUGE =  3.0
 IGAUZB =  3 AGAUGE =  4.0
 IGAUGL =  4 AGAUGE =  5.0
 ANS2     =  1.34979414428364
 # GRAPHS =  28
 ANS1/ANS2 - 1 =  5.551115123125782E-15

 INTEGRATED VALUE OF SQUARE OF EACH GRAPH
 GRAPH          ABSOLUTE          RELATIVE
     1 :     .23935707E-04     .17732857E-04
```

[2] In quadruple precision, we have checked it on FACOM mainframe computer, Sun sparc workstation and HITAC 3050 workstation

[3] See also the footnote in 2.6 for this test and the width of particle.

2	:	.29254454E-04	.21673271E-04
3	:	.30405158E-10	.22525774E-10
4	:	.24923917E+02	.18464976E+02
5	:	.30462254E+02	.22568074E+02
6	:	.16871906E-06	.12499614E-06
7	:	.33536397E+02	.24845564E+02
8	:	.40988511E+02	.30366490E+02
9	:	.14039649E+01	.10401326E+01
10	:	.11486965E+01	.85101606E+00
11	:	.34248206E+02	.25372910E+02
12	:	.50292151E+03	.37259127E+03
13	:	.37723272E+03	.27947426E+03
14	:	.18116921E-07	.13421988E-07
15	:	.30004545E+02	.22228978E+02
16	:	.82173348E+02	.60878430E+02
17	:	.42105481E+01	.31194002E+01
18	:	.92509178E+03	.68535767E+03
19	:	.14010416E-11	.10379669E-11
20	:	.46106302E+03	.34158025E+03
21	:	.37723271E+03	.27947426E+03
22	:	.46034419E-09	.34104770E-09
23	:	.10452751E-13	.77439594E-14
24	:	.31070077E+02	.23018382E+02
25	:	.10159998E-13	.75270725E-14
26	:	.83905907E+02	.62162002E+02
27	:	.28065832E+01	.20792676E+01
28	:	.92394308E+03	.68450666E+03
TOTAL	:	.13497941E+01	

Output 3.1 Result from the gauge invariance check

3.5 Numerical integration

The GRACE system generates a set of FORTRAN subprograms, MAINBS, USERIN, and FUNC, which are necessary for the Monte Carlo integration program package BASES. [4] In this section a description of the integration program package BASES and these subprograms generated by GRACE are given in the following order:

(1) **Program structure of** BASES
Relation among BASES and those subprograms generated by GRACE is described in section 3.5.1.

(2) **Initialization subprogram** USERIN in section 3.5.2.

(3) **Function program of integrand** FUNC in section 3.5.3.

(4) **Histogram package** in 3.5.4.

(5) **Output from** BASES in 3.5.5.

3.5.1 Program structure of BASES

In Fig.3.2 is shown the structure of program BASES. Among them, MAINBS, KINIT, KINEM and KFILL generated by GRACE might need modification by user.

Program flow

The main program MAINBS controls the program flow of integration as follows:

(A) Initialization In subprogram BSINIT, the parameters to control BASES are set to the default values. After that, in the subprogram KINIT called by USERIN, they are initialized to those defined by the user. If some fundamental parameters, like number of dimensions of integral, are not set in KINIT, the program will terminate.
An example of KINIT is already given in section 3.3, and details is given in section 3.5.2.

(B) The grid optimization and integration steps

(1) For each hypercube, N_{trial} points are sampled in the following way;

(a) Sample a small region in the hypercube and sample a point in the small region for each variable.

(b) Call function subprogram FUNC to calculate the differential cross section at the sampled point.

and the estimate and variance of the integral are calculated.

[4] Here and in the followings, the names USERIN and/or FUNC sometimes refer to the group of modules called from them.

(2) Sums of the estimates and variances over all hypercubes are taken to calculate the estimate and error of the integral.

(3) If the integration converges, then go to step (C).

(4) If the integration does not yet converge, then;

For the grid optimization step,
call the subprogram BSGDEF to adjust each width of grids and then go to step (B.1).

For the integration step,
go to step (B.1).

(C) Termination of task
Print out the result.

Main program MAINBS

If the user wants to calculate the cross section for a set of parameters, i.e., for a set of energies, one can do it by modifying MAINBS. To do this, one introduces a loop in MAINBS and defines the parameter in the KINIT for each loop iteration. Such a parameter dependent on the loop would be transferred through a common block coded by the user.

A binary file bases.data is created by invoking integ which saves the probability information as the output to the unit 23. This file bases.data is used in the event generation stage. It should be noted that the parameter file stores the last results if one has introduced loop control in MAINBS.

In the program MAINBS, the name of function program should be given by an external statement and subprogram BSMAIN is called, which is a steering routine of the integration. If the name of function program differs from FUNC, it should be declared with the real name and substituted as the first argument at the call of subroutine BASES.

The results of integration and the produced histograms are stored in file bases.result by calling BSINFO and BHPLOT.

Subroutine BASES

The subroutine BASES called in MAINBS executes the Monte-Carlo integration. It is called as follows:

```
      CALL BASES( FUNC, ESTIM, SIGMA,  CTIME, IT1, IT2 )
```

The meaning of arguments is shown below. If the user wants to make summary table of the results and so on, one edits MAINBS by referring these arguments.

FUNC	input	The name of a function program.
ESTIM	output	A cumulative estimate of the integral.
ERROR	output	The standard deviation of the estimate of the integral.
CTIME	output	The computing time used by the integration step in seconds.
IT1	output	The number of iterations made in the grid optimization step.
IT2	output	The number of iterations made in the integration step.

3.5.2 Initialization subprogram KINIT

At the beginning of the integration job, the subroutine KINIT is called by USERIN to initialize the parameters both for the integration and calculating the integrand.

A sample of KINIT is shown in list 3.6 in the last section.

(1) **Initialization of kinematics**
The definition of center-of-mass energy, the values of cutoff, and so on are given here. Also, some variables which are to be referred in KINEM would be computed here (and are to be transferred to KINEM by common areas), since it can avoid the same computation at every calling of FUNC.

(2) **Initialization of the integration parameters**
The parameters for integration are summarized in the commons /BPARM1/ and /BPARM2/, where all real variables are to be given by the double precision. Below, we present these common areas and the meaning of variables in the common areas.

○ .... ○

```
PARAMETER ( MXDIM = 50 )
COMMON /BPARM1/ XL(MXDIM), XU(MXDIM), NDIM ,NWILD, IG(MXDIM), NCALL
```

○ .... ○

`XL(`i`) (`i` = 1, NDIM)`	The lower bound of i-th variable.
`XU(`i`) (`i` = 1, NDIM)`	The upper bound of i-th variable.
`NDIM`	The dimension of the integral.
`NWILD`	The number of wild variables (at least one and at most 15).
`IG(`i`) (`i` = 1, NDIM)`	The flag for the grid optimization. If `IG(`i`)` $= 1(0)$, the grid for the i-th variable is (not) optimized. If the integrand is approximately constant for a variable, it may give better convergence than varying widths to set the grid uniform for this variable. The default flag is "1" (optimization).
`NCALL`	The number of sampling points per iteration.

The number of real sampling points differs from a given number `NCALL`, which is automatically determined by the following algorithm. It is noticed that the order of variables `XU(`i`)`, `XL(`i`)` and `IG(`i`)` ($i = 1$, `NDIM`), should start with the wild variables.

$N_{call}^{(given)}$	=	1,000			$N_{call}^{(given)}$	=	5,000		
N_{wild}	N_s	N_{cube}	N_g	$N_{call}^{(real)}$	N_{wild}	N_s	N_{cube}	N_g	$N_{call}^{(real)}$
1	25	25	50	1,000	1	25	25	50	5,000
2	22	484	44	968	2	25	625	50	5,000
3	7	343	49	686	3	13	2,197	39	4,394
4	4	256	48	768	4	7	2,401	49	4,802
5	3	243	48	972	5	4	1,024	48	4,096
6	2	64	50	960	6	3	729	48	4,374
7	2	128	50	896	7	3	2,187	48	4,374
8	2	256	50	768	8	2	256	50	4,864
9	1	1	50	1,000	9	2	512	50	4,608
10	1	1	50	1,000	10	2	1024	50	4,096
$N_{call}^{(given)}$	=	10,000			$N_{call}^{(given)}$	=	20,000		
N_{wild}	N_s	N_{cube}	N_g	$N_{call}^{(real)}$	N_{wild}	N_s	N_{cube}	N_g	$N_{call}^{(real)}$
1	25	25	50	10,000	1	25	25	50	20,000
2	25	625	50	10,000	2	25	625	50	20,000
3	17	4,913	34	9,826	3	21	9,261	42	18,522
4	8	4,096	48	9,182	4	10	10,000	50	20,000
5	5	3,125	50	9,375	5	6	7,776	48	15,552
6	4	4,096	48	8,192	6	4	4,096	48	16,384
7	3	2,187	48	8,448	7	3	2,187	48	19,693
8	2	256	50	9,984	8	3	6,561	48	19,683
9	2	512	50	9,728	9	2	512	50	19,968
10	2	1,024	50	9,216	10	2	1,024	50	19,456

The number of subregions per variable N_s is determined by the maximum number which satisfies the two inequalities:

$$N_s = (\frac{N_{call}}{2})^{\frac{1}{N_{wild}}} \leq 25 \quad \text{and} \quad N_s^{N_{wild}} < 32768.$$

The number of hypercubes is given by $N_{cube} = N_s^{N_{wild}}$, then the number of sampling points per hypercube is $N_{trial} = N_{call}/N_{cube}$. Since the number N_{trial} is an integer, the calculated number $N_{call}^{(real)} = N_{trial} \times N_{cube}$ may differ from the given number $N_{call}^{(given)}$.

The table gives the numbers of real sampling points $N_{call}^{(real)}$ depending on the given numbers of sampling points $N_{call}^{(given)}$ and the numbers of wild variables N_{wild}.

```
○..........................................................○

       COMMON /BPARM2/ ACC1, ACC2, ITMX1, ITMX2

○..........................................................○
```

`ACC1`	The accuracy (%) for the grid optimization step (default 0.2 %).
`ACC2`	The accuracy (%) for the integration step (default 0.05 %).
`ITMX1`	The maximum iteration number of the grid optimization step (default 15).
`ITMX2`	The maximum iteration number of the integration step (default 100).

(3) **Initialization of Histograms and Scatter plots**
To make a histogram and a scatter plot, `XHINIT` and/or `DHINIT` are to be called for the initialization. The meaning of arguments and details will be given in section 3.5.4.

Normally, in the code generated automatically, the initialization of histograms is done for each `X( )` distribution, E_i distribution and $\cos\theta_i$ distribution where E_i is the energy of final i-th particle and $\cos\theta_i$ is the final i-th particle's angle with respect to the beam axis.

3.5.3 Function program of the integrand

The function program calculates the value of integrand at the sampling point fed by `BASES`.

A set of numerical values of the integration variables at a sampling point is passed through the argument of function program. A typical structure of the function program is given in the source list 3.8 where the dimension of integration is five.

1) Calculate the kinematical variables, by which the differential cross section is described, from the integration variables, `X(`i`)` for i = 1, `NDIM`. This is done in `KINEM`.

2) If, in the last step, a sampling point is found to be outside of the kinematical boundary, set the value of function equal to zero and return.

3) If the point is inside the kinematical boundary, calculate the numerical value of the differential cross section and set the value of function equal to the calculated value.

4) If a histogram or a scatter plot is required, call subprogram XHFILL or subprogram DHFILL once. This is done in KFILL.

5) If the kinematics is multi-valued with respect to the input values of X(i), special care is required for the communication between FUNC and KINEM. See an example below.

○..○

```
      DOUBLE PRECISION FUNCTION FUNC(X)
      DOUBLE PRECISION X(5)
      FUNC = 0.0

      ... Calculation of the kinematics ...
      IF( the point is outside the kinematical boundary ) RETURN

      FUNC = is calculated from X(i) for i = 1, 5.

      CALL XHFILL( ID, V, FUNC )
      CALL DHFILL( ID, VX, VY, FUNC)

      RETURN
      END
```

Source list 3.8 Typical structure of FUNC

○..○

An example of FUNC for the process $e^+e^- \to W^+W^-\gamma$ is given in the source list 3.9. The interaction with KINEM is also described in section 3.3.3. The structure of this example is as follows:

1) The array X stores the values of integration variables. Another array XX keeps its values.

2) Total number of external particles NEXTRN and XX are transferred to subprogram KINEM. The tables of momenta P and inner-products of them PP, and normalization factor YACOB are received from KINEM.

3) P and PP are copied to the common variables PExxxx and PPROD, respectively, and they are used in the amplitude calculation.

4) Subprogram AMPTBL calculates the amplitudes and makes the tables of them.

5) Summation over the spin states by calling the subprogram AMPSUM.

6) Fill the histograms and scatter pots by calling KFILL in which the subprograms XHFILL and DHFILL are called.

7) The variable JUMP

   If the sampling point is out of the kinematical boundary, JUMP is set to a non zero integer in KINEM. For this case, the amplitude computation is skipped.

8) The variables NREG and IREG

   When the kinematics contains a multi-valued function, *i.e.* one sampling point in the integration volume corresponds to several points in the phase space, the variables NREG and IREG take the control as is described in section 3.3.3.

○...○

```
* FILE "func.f" is generated by GRACE System (Minami-Tateya Group)
* Grace Version   1. 1          1994-Aug-19
************************************************************************
      FUNCTION FUNC(X)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8     FUNC

      PARAMETER ( MXDIM = 50 )
      COMMON / LOOP0 / LOOP
      COMMON / BASE1 / XL(MXDIM),XU(MXDIM),NDIM,NWILD,
     &                 IG(MXDIM),NCALL
      COMMON / BASE2 / ACC1,ACC2,ITMX1,ITMX2
      COMMON / BASE3 / SI,SI2,SWGT,SCHI,SCALLS,ATACC,NSU,IT,WGT

      REAL*8   X(MXDIM)

      INCLUDE 'incl1.f'
      INCLUDE 'inclk.f'

      COMMON /AMSPIN/JHS(NEXTRN), JHE(NEXTRN), ASPIN
      REAL*8     ANS0, ANS

*  P : Table of four momenta
* PP : Table of inner products
      REAL*8     XX(MXDIM),P(4,NEXTRN),PP(NEXTRN,NEXTRN)
      COMMON /SP4VEC/ VEC(4,NEXTRN)
*=======================================================================
*         Initialization
*=======================================================================

      ANSUM = 0.0D0

      DO  5 I = 1, NDIM
         XX(I) = X(I)
    5 CONTINUE
```

```
      NREG  = 1
      DFT   = 0.D0
*=======================================================================
*          Kinematics
*=======================================================================

      DO 1000 IREG = 1 , MXREG

         IF( IREG .GT. NREG ) GO TO 1000

        CALL KINEM(NEXTRN, XX, P, PP, YACOB,NREG,IREG,JUMP)

*-----------------------------------------------------------------------
*          Reset the temporal buffer for the region 1
*-----------------------------------------------------------------------
      IF( IREG .EQ. 1 ) THEN
          DFT = 0.D0
          DO 180 K = 1, NEXTRN
          DO 180 J = 1, 4
             VEC(J,K) = 0.D0
  180     CONTINUE
      ENDIF

         IF( JUMP .NE. 0 ) GO TO 1000

*-----------------------------------------------------------------------
*          For user's cut
*-----------------------------------------------------------------------

C     CALL USRCUT(JUMP)
C     IF( JUMP .NE. 0 ) GOTO 1000

*-----------------------------------------------------------------------
*          Four momenta of external particles
*-----------------------------------------------------------------------

      DO 20 I = 1, 4
*           1:  EL-  INITIAL   LPRTCL  MASS=AMEL
        PE0001(I) = P(I,  1)
*           2:  EL+  INITIAL   LANTIP  MASS=AMEL
        PE0002(I) = P(I,  2)
*           3:  AB   FINAL     LPRTCL  MASS=AMAB
        PE0003(I) = P(I,  3)
*           4:  WB+  FINAL     LPRTCL  MASS=AMWB
        PE0004(I) = P(I,  4)
*           5:  WB-  FINAL     LANTIP  MASS=AMWB
        PE0005(I) = P(I,  5)
   20 CONTINUE

*-----------------------------------------------------------------------
*          Inner products of momenta of external particles
*-----------------------------------------------------------------------
```

```
      DO 30 J = 1, NEXTRN
      DO 30 I = 1, NEXTRN
        PPROD(I, J) = PP(I, J)
   30 CONTINUE

*=========================================================================
*          Amplitude calculation
*=========================================================================
*       =============
        CALL AMPTBL
*       =============
*       ===================
        CALL AMPSUM(ANS0)
*       ===================

        FKNORM = YACOB*ASPIN
        ANS    = ANS0*FKNORM
        ANSUM  = ANSUM + ANS

*-------------------------------------------------------------------------
      CALL KFILL(NEXTRN, NDIM, X, P, PP, ANS)

*-------------------------------------------------------------------------
*          Save four momenta and probabilities of the region 1
*-------------------------------------------------------------------------
      IF( IREG .EQ. 1 ) THEN
          DFT = ANS
          DO 420 K = 1, NEXTRN
          DO 420 J = 1, 4
             VEC(J,K) = P(J,K)
  420     CONTINUE
      ENDIF

*=========================================================================
*          Update summary table
*=========================================================================

        ANSP(0) = ANSP(0) + WGT*ANS
        DO 60 IGR = 1, JGRAPH
          ANSP(IGR)=ANSP(IGR) + WGT*YACOB*ASPIN*ANCP(IGR)
   60   CONTINUE
        NKCALL = NKCALL + 1
        IF(NKCALL.GT.10000) THEN
          NKCALL = NKCALL - 10000
          FKCALL = FKCALL + 10000
        ENDIF

 1000 CONTINUE

      FUNC = ANSUM
```

```
*-------------------------------------------------------------------
*         Put the final 4 vectors into the arrays VEC()
*-------------------------------------------------------------------
*     IF( FUNC .GT. 0.D0 ) THEN
*         IF( DFT/FUNC .LT. DRN(DUM)) THEN
*             DO 850 K = 1, NEXTRN
*             DO 850 J = 1, 4
*                VEC(J,K) = P(J,K)
* 850         CONTINUE
*         ENDIF
*     ENDIF

      RETURN
      END
```

Source list 3.9 An example of FUNC

o...o

3.5.4 Histogram package

The program package BASES/SPRING has its own histogram package, whose characteristics are as follows;

1) To initialize the histograms and scatter plots, the following routines are to be called in KINIT.

```
  CALL XHINIT( ID#,
 .             lower_limit, upper_limit, # of bins, ' Title '),
and
  CALL DHINIT( ID#,
 .              x_lower_limit, x_upper_limit, # of x bins,
 .              y_lower_limit, y_upper_limit, # of y bins,
 .                                                ' Title '),
```

respectively.

The ID and bin numbers are to be given by an integer value, and the lower and upper limits are to be given by the double precision values. The maximum bin number both for histograms and scatter plots is 50, which is defined by the paper size. When too many histograms or scatter plots are initialized, the first NHIST-1 histograms and NSCAT scatter plots are initialized and the others are neglected.

2) To fill the histograms or the scatter plots on a scalar computer the following filling routines are called in the function KFILL which is called from FUNC:

```
CALL XHFILL( ID#, V, FUNC )       for each histogram
CALL DHFILL( ID#, VX, VY, FUNC)   for each scatter plot
```

3) The outputs of histograms and scatter plots can display even a *negative* function as well as the positive definite function.

4) In `GRACE` the histograms are written on files `bases.result` and `spring.result`.

3.5.5 Output from `BASES`

The outputs consist of the following items.

1) **Parameters for BASES**

   After returning from `USERIN`, the parameters given there are printed out, some of which are numbers of the integration variables, the wild variable and the sampling points per iteration, N_{dim}, N_{wild} and $N_{call}^{(given)}$. From these numbers, the number of the small-regions per variable N_g, that of the sub-regions per variable N_s, that of real sampling points per iteration $N_{call}^{(real)}$, and that of hypercubes N_{cube} are calculated and printed. Further, for each integration variable, the lower and upper limits, `XL(`i`)` and `XU(`i`)`, the grid optimization flag `IG(`i`)`, and the kind of variable (*i.e.* wild or not) are printed. And finally the maximum iteration number and the expected accuracy both for the grid optimization and the integration steps are printed. An example of this output is given in the output 3.2.

○ .... ○

```
                                                    Date: 94/ 8/19  20:43
          **********************************************************
          *                                                        *
          *     BBBBBBB     AAAA     SSSSSS   EEEEEE   SSSSSS      *
          *     BB    BB   AA  AA   SS    SS  EE      SS    SS     *
          *     BB    BB  AA    AA  SS        EE      SS           *
          *     BBBBBBB   AAAAAAAA   SSSSSS   EEEEEE   SSSSSS      *
          *     BB    BB  AA    AA        SS  EE            SS     *
          *     BB    BB  AA    AA  SS    SS  EE      SS    SS     *
          *     BBBB BB   AA    AA   SSSSSS   EEEEEE   SSSSSS      *
          *                                                        *
          *                   BASES Version 5.1                    *
          *           coded by S.Kawabata KEK, March 1994          *
          **********************************************************

       <<   Parameters for BASES    >>

        (1) Dimensions of integration etc.
            # of dimensions :    Ndim    =        5   ( 50 at max.)
            # of Wilds      :    Nwild   =        4   ( 15 at max.)
            # of sample points : Ncall   =    41472(real)    50000(given)
            # of subregions    : Ng      =       48 / variable
            # of regions       : Nregion =       12 / variable
            # of Hypercubes    : Ncube   =    20736
```

```
(2) About the integration variables
    ------+---------------+---------------+-------+-------
       i       XL(i)           XU(i)       IG(i)   Wild
    ------+---------------+---------------+-------+-------
       1     .000000E+00    1.000000E+00     1      yes
       2     .000000E+00    1.000000E+00     1      yes
       3     .000000E+00    1.000000E+00     1      yes
       4     .000000E+00    1.000000E+00     1      yes
       5     .000000E+00    1.000000E+00     1       no
    ------+---------------+---------------+-------+-------

(3) Parameters for the grid optimization step
    Max.# of iterations: ITMX1 =        5
    Expected accuracy   : Acc1  =    .2000 %

(4) Parameters for the integration step
    Max.# of iterations: ITMX2 =        5
    Expected accuracy   : Acc2  =    .0100 %
```

Output 3.2 General information of the integration

o . . . . o

2) **Convergency behavior**

BASES issues some messages when the convergence is not well established.

However the logic to check the convergence is rather simple and the message might not be always precise. More information can be obtained by reading the output carefully when the print flag is set to issue details of integration steps.

According to the print flag the two kinds of convergency behaviors can be obtained, one is for the grid optimization step and another is for the integration step. The print format consists of the result of each iteration and the cumulative result and the computing time used.

In the result of each iteration, the sampling efficiency (the percentage of the points inside of the kinematical boundary), the ratio of the numbers of the negative valued sampling points to the total number of sampling points in unit of percent, the estimate of integral of the iteration and the estimated accuracy in unit of percent are shown.

In the cumulative result, the cumulative estimates of integral and error are listed up in addition to the accuracy in the unit of percent. The computing time in this table is measured from the beginning of the grid optimization step till the end of the current iteration, which does not contain the time of overhead but that used for estimating integral.

In the convergency behavior for the grid optimization step, it should be checked that the accuracy for each iteration does decrease iteration by iteration and converge to a stable value. If not the case, it is recommended to increase the number of sampling points N_{call} and try again. When the increment of number of sampling points does not help to improve the behavior, the current choice of the integration variables may not be suitable for the behavior of integrand. Examples of convergency behavior both for the grid optimization and integration steps are given in the outputs 3.3 and 3.4, respectively.

```
                                                       Date: 94/ 8/19  20:43
               Convergency Behavior for the Grid Optimization Step
------------------------------------------------------------------------------
<- Result of  each iteration ->  <-     Cumulative Result     -> < CPU  time >
 IT Eff R_Neg   Estimate  Acc %  Estimate(+- Error )order  Acc % ( H: M: Sec )
------------------------------------------------------------------------------
  1 100   .00  4.203E+00  5.516  4.203258(+- .231843)E 00  5.516    0: 3:53.85
  2 100   .00  3.907E+00  2.289  3.944922(+- .083419)E 00  2.115    0: 7:45.87
  3 100   .00  3.966E+00  1.583  3.958399(+- .050164)E 00  1.267    0:11:39.86
  4 100   .00  3.944E+00   .966  3.949475(+- .030344)E 00   .768    0:15:30.03
  5 100   .00  4.014E+00  1.369  3.964637(+- .026563)E 00   .670    0:19:23.40
------------------------------------------------------------------------------
```

Output 3.3 Convergency behavior for the grid optimization step

```
                                                       Date: 94/ 8/19  20:43
               Convergency Behavior for the Integration Step
------------------------------------------------------------------------------
<- Result of  each iteration ->  <-     Cumulative Result     -> < CPU  time >
 IT Eff R_Neg   Estimate  Acc %  Estimate(+- Error )order  Acc % ( H: M: Sec )
------------------------------------------------------------------------------
  1 100   .00  3.933E+00   .910  3.933202(+- .035802)E 00   .910    0:23:15.67
  2 100   .00  3.978E+00  1.435  3.945768(+- .030331)E 00   .769    0:27: 7.03
  3 100   .00  3.953E+00   .845  3.948889(+- .022452)E 00   .569    0:31: 0.03
  4 100   .00  3.928E+00   .926  3.943177(+- .019106)E 00   .485    0:34:51.83
  5 100   .00  3.942E+00   .849  3.942924(+- .016594)E 00   .421    0:38:42.65
------------------------------------------------------------------------------
```

Output 3.4 Convergency behavior for the integration step

The accuracy of each iteration must be stable in the integration step. When the integration variables does not suit for the integrand, it fluctuates iteration by iteration and may jump suddenly to a big value in the worst case.

In the interactive mode the convergency behavior is printed iteration by iteration, while it is printed only for the final 50 iterations in the batch mode. This mode is to be selected at the installation time by setting the flag "`INTV`" in the routine `BSMAIN`.

3) **Histograms and scatter plots**

If histograms and scatter plots are initialized in KINIT and filled in KFILL, their results are printed at each end of the grid optimization step and the integration step according to the print flag. In the output 3.5 we show only the histogram ID = 3 for saving the space of this manual.

The first and the last bins of histogram represent values of the underflow and the overflow bins, respectively. The first column shows the lower edge value of each histogram bin. The second column represents the estimated differential value and error after the characters "+-", both of which are to be multiplied by a factor "E xx" shown as order. On the right hand side of these columns a histogram of the differential values is drawn both in the linear scale with "*" and in the logarithmic scale with "O". If negative values exist in some bins only the linear scale histogram is shown.

The scatter plot represents only the relative height of the function. The height of the function value is described by ten characters; 1, 2, 3, ..., 8, 9 and *, while the depth (for the negative values) is displayed by ten characters; a, b, c, d, ..., h, i and #. The point which has a negative value but larger than the value of the level "a" is indicated by "-". On the other hand, the point describing a positive value but less than the level "1" is given either by "+" (if a negative value exists somewhere) or by "." (if only the positive values exist). In the output 3.6 an example of scatter plot is shown.

```
Histogram (ID =  3 ) for X( 3) SPECTRUM
                                Linear Scale indicated by "*"
    x      d(Sigma)/dx     0.0E+00      8.3E+00      1.7E+01     2.5E+01
+-------+-----------------+------------+------------+-----------+-----------+
I  E  0 I  .000        E  0I                                                 I
I  .000 I 2.595+-  .051 E  1I****************************************000000000I
I  .020 I 8.402+-  .279 E  0I*************00000000000000000000000000          I
I  .040 I 5.954+-  .265 E  0I**********00000000000000000000000000             I
I  .060 I 4.817+-  .229 E  0I********00000000000000000000000000               I
I  .080 I 3.415+-  .171 E  0I******0000000000000000000000000                  I
I  .100 I 3.698+-  .167 E  0I******00000000000000000000000000                 I
I  .120 I 3.053+-  .148 E  0I*****0000000000000000000000000                   I
I  .140 I 2.898+-  .154 E  0I*****0000000000000000000000000                   I
I  .160 I 2.443+-  .160 E  0I****000000000000000000000000                     I
I  .180 I 2.917+-  .155 E  0I*****0000000000000000000000000                   I
I  .200 I 2.183+-  .134 E  0I****00000000000000000000000                      I
I  .220 I 1.874+-  .127 E  0I***00000000000000000000000                       I
I  .240 I 2.161+-  .129 E  0I****00000000000000000000000                      I
I  .260 I 1.924+-  .123 E  0I***00000000000000000000000                       I
I  .280 I 1.589+-  .110 E  0I***0000000000000000000000                        I
I  .300 I 1.543+-  .100 E  0I***000000000000000000000                         I
I  .320 I 1.387+-  .107 E  0I***00000000000000000000                          I
I  .340 I 1.316+-  .086 E  0I**000000000000000000000                          I
I  .360 I 1.239+-  .091 E  0I**00000000000000000000                           I
I  .380 I 1.282+-  .091 E  0I**000000000000000000000                          I
I  .400 I 1.021+-  .069 E  0I**0000000000000000000                            I
I  .420 I 9.757+-  .814 E -1I**000000000000000000                             I
I  .440 I 9.046+-  .458 E -1I**000000000000000000                             I
I  .460 I 8.947+-  .466 E -1I**000000000000000000                             I
I  .480 I 9.287+-  .679 E -1I**000000000000000000                             I
I  .500 I 8.258+-  .790 E -1I**00000000000000000                              I
I  .520 I 9.083+-  .589 E -1I**000000000000000000                             I
I  .540 I 9.549+-  .668 E -1I**000000000000000000                             I
I  .560 I 9.718+-  .739 E -1I**000000000000000000                             I
I  .580 I 9.076+-  .840 E -1I**000000000000000000                             I
I  .600 I 1.193+-  .088 E  0I**00000000000000000000                           I
I  .620 I 1.401+-  .093 E  0I***00000000000000000000                          I
I  .640 I 1.428+-  .119 E  0I***000000000000000000000                         I
I  .660 I 1.097+-  .101 E  0I**0000000000000000000                            I
I  .680 I 1.947+-  .123 E  0I***00000000000000000000000                       I
I  .700 I 1.979+-  .126 E  0I***00000000000000000000000                       I
I  .720 I 2.099+-  .122 E  0I****00000000000000000000000                      I
I  .740 I 2.214+-  .138 E  0I****00000000000000000000000                      I
I  .760 I 1.784+-  .125 E  0I***00000000000000000000000                       I
I  .780 I 2.376+-  .141 E  0I****000000000000000000000000                     I
I  .800 I 2.483+-  .144 E  0I****000000000000000000000000                     I
I  .820 I 2.566+-  .149 E  0I****0000000000000000000000000                    I
I  .840 I 2.883+-  .152 E  0I*****0000000000000000000000000                   I
I  .860 I 2.803+-  .150 E  0I*****000000000000000000000000                    I
I  .880 I 3.448+-  .175 E  0I******0000000000000000000000000                  I
I  .900 I 3.906+-  .176 E  0I******00000000000000000000000000                 I
I  .920 I 4.146+-  .196 E  0I*******00000000000000000000000000                I
I  .940 I 5.760+-  .247 E  0I*********000000000000000000000000000             I
I  .960 I 8.741+-  .291 E  0I**************0000000000000000000000000          I
I  .980 I 2.612+-  .052 E  1I****************************************000000000I
I  E  0 I  .000        E  0I                                                 I
+-------+-----------------+-------------------+-------------------+---------+
    x      d(Sigma)/dx     1.0E-01             1.0E+00             1.0E+01
                                Logarithmic Scale indicated by "O"
```

Output 3.5 An example of histogram

```
o . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . o

Scat_Plot (ID =  6 ) for X( 3)-X( 4) DISTRIBUTION

  E  0    +--------------------------------------------------+
  .980    I8212.1............... .. . ........1.1.....1..1137I
  .960    I8332..1111.1.......... ...................11.114*I
  .940    I832.................. .  .. .............1.1.212*I
  .920    I821.11.1.................................1.1.13*I
  .900    I722111.1.....1.................. ........11.11236I
  .880    I821..1.............. ..................1.....1118I
  .860    I7111.........1.......... ..............1..1.1.1236I
  .840    I821.11.1.1..1........... ...... .1....1.....11227I
  .820    I831.11.1.....1......... ...... ....1.....1.1...117I
  .800    I821211....11..... .......1 ....1.......111...1.227I
  .780    I421.1.1..1...... ...............1. .....11.111.128I
  .760    I711....111..1..........1..1...........1....1111124I
  .740    I421.1.1..1............ ..........1........111116I
  .720    I5212.1...11............ ..... ......1..........16I
  .700    I811..211............. .. ....... ...1.......11117I
  .680    I5211.11...........................1.....1.....26I
  .660    I8111..........1....... ... .....1......1..1.1..137I
  .640    I41...1..1.................... .....1...1..1111216I
  .620    I72.1.1........... ........................11127I
  .600    I7221.1.1...1....... ...... .. .. ...........1.1137I
  .580    I7211.1.........1....... .. .........1.....1.111.15I
  .560    I52...11.1........ .....................1....1.136I
  .540    I52111........11..... ...... ..........1...1....116I
  .520    I5112.....1......1.. ...... ..... .............1224I
  .500 Y  I5221....1.....1.. ......................11.1.1138I
  .480    I72...1............... . . ....... .......1....1125I
  .460    I5.21.11.....1...1.....................1.......126I
  .440    I611......1....... ..................11........2.17I
  .420    I7111.11.11.............................1......126I
  .400    I622...... .....1........   . .................1125I
  .380    I73..1.11............ . . .... .........1....111114I
  .360    I51111....1.......1.... . .........1......1..1.1.18I
  .340    I8211..1... 1........... ..........1........21118I
  .320    I4211.....1.......... .. ....... ........1.1111..26I
  .300    I52111.11.......1........ . . .......1... ....1.227I
  .280    I611..1...1...11..1........ .... ...1...1..1.111127I
  .260    I8111................................... 1...1.215I
  .240    I62.1.1....1.... ..... ... . .............1..1.234I
  .220    I9111.12............ .....................1..11115I
  .200    I63.111..1......... ... ...........11....1..1...128I
  .180    I6211..1..1..................  ...........1.1.1128I
  .160    I512..1......1........1.........  .....1.11...1.127I
  .140    I8221.....1......1........1.... .........1..1.21217I
  .120    I83111..1.1..1...........1...  .1........1121.11139I
  .100    I7122..11..... ........... ... . .....1...1...12127I
  .080    I821.1....1............   .....................237I
  .060    I72121111.....1.......... ............1......11147I
  .040    I93311111.1............... ... .....1..1...1.11.11*I
  .020    I9211..1.......1.... .....................1.12.1238I
  .000    I8232...1.1..1.......................1...1...1..327I
  E  0    +--------------------------------------------------+
 Low-                              X
 Edge
           00000000000000000000000000000000000000000000000000
  E  0     ..................................................
           00000011111222223333344444555555666677777788899999
  Low-     01367923579245791457914580135791358014579235801367
  Edge     09909909999009999099990900999999990090999099009909
```

Output 3.6 An example of scatter plot

○ .... ○

5) **List of computing time**

As well as the message, a list of computing time is printed at the end of the job as shown in a output 3.7.

When the integration has been achieved by a single job, the items (1) and (2) are exactly the same. If the integration is performed by several jobs, the computing time is only for the current job, while that for total calculation includes all computing time from the beginning. The expected event generation time is printed at the item (3). From this value, the computing time limit for the event generation will be evaluated.

○ .... ○

```
<<   Computing Time Information   >>

          (1) For BASES              H: M:  Sec
              Overhead           :   0: 0: 0.05
              Grid Optim. Step   :   0:19:23.40
              Integration Step   :   0:19:19.25
              Go time for all    :   0:38:42.71

          (2) Expected event generation time
              Expected time for 1000 events :      7.99 Sec
```

Output 3.7 Computing time information

○ .... ○

6) **Final result of integration**

The results of integration can be read from the output shown in the list 3.4. Since the value of the integral and its error are given by the arguments of BASES, when the user needs some formatted output for the results, one can make it by editing MAINBS. For the reference to the arguments of BASES, one can find information in section 3.5.1.

7) **Probability information**

Before terminating the integration job, BASES generates a data file by the routine BSWRIT, where

1) Probability information
   Probability of each hypercube, according to which a hypercube is sampled in the event generation.
2) The maximum values of integrand
   The maximum value of integrand in each hypercube is stored, by which the sampling point are tested by using a uniform random number.

3) Contents of histograms
   In the event generation, those histograms are printed out comparing to the distribution of generated events which are defined in the integration by BASES. For this purpose, the contents of histograms taken in the BASES are stored in this file.

Although there are several versions of BASES/SPRING, *e.g.* the original BASES /SPRING, BASES25/SPRING25, and BASES50/SPRING50, the data format of this file *does* depend on the version. The newest one is BASES50/SPRING50 and is recommended to use. We call BASES50/SPRING50 as BASES/SPRING throughout this manual. *Be careful not to use the different versions for* BASES *and* SPRING.

3.6 Event generation

An advantage of BASES/SPRING packages is that if a differential cross section is integrated by BASES the four vectors of final state particles are easily generated with weight one by using SPRING. (See section 2.8 in Ref.[8].) In this section, a description of SPRING is given in the following order.

(1) Input for SPRING

(2) Program structure of SPRING

(3) Specifications of the subprograms to be prepared

(4) Output from SPRING

The event generation by SPRING is normally quite fast. But if calculation of the integrand requires much computing time, both the integration and the event generation takes much time. For such a case we recommend to use a vector computer if available. A vector version of SPRING will be described in Ref.[8].

3.6.1 Input for SPRING

There are two inputs for SPRING. One is a file of the probability information for each hypercube, which is produced by the integration package BASES. In this file the following data are saved:

(a) The probability of sampling each hypercube.

(b) The maximum value of integrand in each hypercube.

(c) The contents of histograms and scatter plots.

(d) The control data for BASES.

SPRING with a different version from that of BASES should not be used for the event generation, since the data format of this file *does* depend on the version as mentioned in the previous section. The most new one is BASES50/SPRING50 and is recommended to use.

At the beginning of the generation job, the following parameters are read from the logical unit 5:

(1) The number of events to be generated, which is stored in the variable MXEVNT.

The event generation loop is terminated not only by the generation of given numbers of events, but also by too many failure of the generation.

3.6.2 Program structure of SPRING

In Fig.3.2, the program structure of SPRING is shown, where the subprograms in the `solid box` are generated by GRACE automatically. Others are included in the BASES/SPRING library or CHANEL library.

The parameter MXTRY defines the maximum number of trials for getting an accepted event, which makes the event generation free from an infinite loop described later in this subsection.

The program flow in MAINSP is as follows;

(A) Initialization

   (1) The subprogram BSINIT is called.

   (2) By the subroutine BSREAD the probability information of all hypercubes and the contents of histograms and scatter plots are read from a binary file.

   (3) USERIN is called for initialization of histograms *etc.* and kinematics.

   (4) The probability distribution read from the file is changed into the cumulative distribution.

(B) Event generation loop
The loop is controlled by variables MXTRY and MXEVNT.

   (1) A hypercube (say the i-th hypercube) is sampled according to its probability by a random number generated by a function DRN.

   (2) A point is sampled in a small region in the i-th hypercube, sampled in the step (B.1).

   (3) The value of the integrand at the sampled point ζ is calculated by calling FUNC.

   (4) If the sampled point ζ satisfies the condition

$$\frac{f(\zeta)}{p(\zeta)}/Max.[\frac{f(x_i)}{p(x_i)}] < \eta \text{ (= a uniform random number)},$$

   then this point is accepted as an event, and go out of the event generation loop.

   (5) If the sampled point is not accepted and the number of trials to get an event is less than the given value of MXTRY, the histogram information for the point is cleared by the subroutine SHCLER and go to the step (B.2).

   (6) If the number of trials is larger than the given value, this hypercube is abandoned, and go to the step (B.1).

(C) Record and analysis of generated events
When a point is accepted as an event, the user is responsible for the record and/or

the analysis of the generated events. If the four vectors of generated events are going to be written on a file, the file should be opened and closed by the user in MAINSP before and after the loop of event generation. It is recommended to record the event inside the loop of generation. Just after the call of SPRING one can access the four vectors of the event by referring the common area /SP4VEC/ which is filled in FUNC (List 3.8).

```
○..........................................................○

      COMMON /SP4VEC/ VEC(4,NEXTRN)

○..........................................................○
```

This common area is explicitly given in MAINSP. Here, the variable VEC(i,j) stores j-th particle's four vector where i= $1, 2, 3, 4$ correspond to p_x, p_y, p_z, E, respectively. The ordering of j is the same in the input file supplied at the graph generation and it includes both initial and final particles.

(D) Check the number of events
Increment the number of generated event and test the remaining computing time. If the number of events is less than the given number or there remains enough computing time for generating one event, go to the step (B.1).

(E) Termination
Before terminating the job, histograms and scatter plots are printed by SHPLOT.

As described in the step (B.5), the parameter MXTRY plays an important role. Without limiting the maximum number of trials to get an event, the generation loop may come into an infinite loop. This parameter is set in the main program MAINSP and default number is equal to 50.

3.6.3 Subprograms to be prepared

GRACE automatically generates main programs and subprograms for the integration by BASES though a part of them might need modification by the user. These subprograms are also necessary for the event generation by SPRING except for MAINBS. As their specifications can be found in subsections 3.5.1, 3.5.2 and 3.5.3. we will not repeat them here unless there exist difference between their specifications in BASES and SPRING.

No change

Subprograms USERIN, KINIT and KINEM, used in BASES step, does not need to be modified. Especially the subprogram USERIN should be identical to that used in BASES.

MAINSP

Main program `MAINSP` is produced by `GRACE` in a complete form. The user sometimes needs to edit it for the record of events, definition of output file, or the change of parameters `MXTRY` and/or `MXEVNT`. The user might touch it when the name of parameter file created by `BASES` differs from the default.

FUNC

When the integrand is a single-valued function, it *should not be changed.* But if it is a two-valued function, the last part of the function code must be activate, which part is normally commented out just after the source generation by `GRACE`. The example of this case is shown in the source list 3.14.

When the kinematics is described by a many-valued function, a sample point in the integration volume corresponds to several distinct points in the phase space, for each of which differential cross section is calculated. In the integration the values of differential cross section at these points are simply summed and the sum is given as the function value `FUNC`, while in the event generation a point among these points must be sampled according to their probabilities.

The example in the source list 3.14 and 3.12 shows the two-valued function case. For the first point the four vectors and numerical value of the differential cross section are stored in an arrays `VEC`(j, k) and variable `DFT` at the do loop `420` in the list 3.12. If the ratio of `DFT` and `FUNC` is less than a random number, the second point in the phase space is taken as a sampled point, where `FUNC` is the sum of the differential cross section values at these two points. This method can be easily extended to a many-valued function case.

```
o..............................................................o

 1000 CONTINUE

      FUNC = ANSUM

*--------------------------------------------------------------
*       Put the final 4 vectors into the arrays VEC()
*--------------------------------------------------------------
      IF( FUNC .GT. 0.D0 ) THEN
          IF( DFT/FUNC .LT. DRN(DUM)) THEN
              DO 850 K = 1, NEXTRN
              DO 850 J = 1, 4
                 VEC(J,K) = P(J,K)
  850         CONTINUE
          ENDIF
      ENDIF

      RETURN
      END
```

Source list 3.10 The last part of `FUNC`

○ . . . . ○

3.6.4 Output from SPRING

The output from SPRING consists of the general information, histogram output, the number of trials distribution and the four vector output. There are two kinds of histogram output, one is the original histogram and other is the additional histogram.

General information

After generating events, the following information is printed:

○ . . . . ○

```
                                                 Date: 94/ 8/19  21:43
   **********************************************************
   *                                                        *
   *    SSSSS   PPPPPP   RRRRRR   IIIII  N    NN   GGGGG    *
   *   SS   SS  PP   PP  RR   RR   III   NN   NN  GG   GG   *
   *   SS       PP   PP  RR   RR   III   NNN  NN  GG        *
   *    SSSSS   PPPPPP   RRRRR     III   NNNN NN  GG  GGGG  *
   *        SS  PP       RR  RR    III   NN NNNN  GG   GG   *
   *   SS   SS  PP       RR   RR   III   NN  NNN  GG   GG   *
   *    SSSSS   PP       RR    RR IIIII  NN   NN   GGGGG    *
   *                                                        *
   *                  SPRING Version 5.1                    *
   *           coded by S.Kawabata KEK, March 1994          *
   **********************************************************

Number of generated events    =     10000
Generation efficiency         =    28.144 Percent
Computing time for generation =   202.360 Seconds
               for Overhead   =      .340 Seconds
               for Others     =      .090 Seconds
GO time for event generation  =   202.790 Seconds
Max. number of trials MXTRY   =        50 per event
Number of miss-generation     =        95 times
```

Output 3.9 General information of the event generation

○ . . . . ○

When the number of trials to generate one event exceeds the number MXTRY, this outbreak is counted as the number of mis-generation. If this number is not negligible small, something happens in the event generation, *e.g.* mis-match between the integrand and the probability information of the input file, or the grids determined by BASES are not enough optimized. This can be also checked by the number of trials distribution described later.

Histograms

There are two kinds of histograms.

One is the original histogram, which is defined in the integration stage by BASES. The contents of these histograms produced in the integration are read from the input file and are compared with the frequency distribution taken in the event generation. This comparison is done in the logarithmic scale, where the statistical error of each bin is represented by "< >". If error is smaller than the two character space, only the frequency is shown by "O". The histogram obtained by BASES is represented by "*". An example of the original histogram is shown in the output 3.10, which can be compared with the histogram shown in the output 3.5 of section 3.5.5.

Another is to report the trial number distribution of SPRING shown is output 3.11.

Number of trials distribution

The number of trials distribution is printed out at the final stage, by which we can see how efficient the event generation was. The first column represents the number of trials to obtain one event and the number of events is shown in the third column. An example for the process $e^+e^- \rightarrow W^+W^-\gamma$ is shown in the output 3.12, where about 80% of events are generated with the first trial. If this distribution has a long tail, this means generation efficiency is low, then the following points should be tested:

(1) The grids determined by BASES is not optimized well. If this is the case, try integration again with more sampling points (by setting NCALL larger than the current number).

(2) The integrand does not match for the probability distribution in the input file. Check whether the subprograms USERIN and FUNC are exactly identical to those used in the integration.

(3) The integration could not give a good answer due to unsuitable integration variables for the integrand. In this case, improvement of the kinematics is required.

```
Original Histogram (ID =  1 ) for X( 1) SPECTRUM
Total =     10000 events   "*" : Orig. Dist. in Log Scale.
   x      d(Sig/dx)  dN/dx 1.0E-01          1.0E+00          1.0E+01
+-------+----------+-------+----------------+----------------+---------------+
I  E  0 I  .000E  0I      0I                                                 I
I  .000 I 5.967E  1I   3116I**********************************************0  I
I  .020 I 9.278E  0I    495I********************************<>               I
I  .040 I 5.512E  0I    284I****************************<>                   I
I  .060 I 3.640E  0I    178I*************************<>                      I
I  .080 I 2.779E  0I    137I***********************<>                        I
I  .100 I 2.201E  0I    132I**********************<0>                        I
I  .120 I 1.989E  0I     98I********************<0>                          I
I  .140 I 1.461E  0I     69I*****************<0>                             I
I  .160 I 1.293E  0I     56I****************<0>                              I
I  .180 I 1.147E  0I     58I****************<0>                              I
I  .200 I 1.039E  0I     46I**************<0>                                I
I  .220 I 9.586E -1I     50I***************<0>                               I
I  .240 I 7.552E -1I     33I************<0>                                  I
I  .260 I 8.063E -1I     33I************<0>*                                 I
I  .280 I 7.528E -1I     43I**************<0>                                I
I  .300 I 6.514E -1I     37I*************<0>                                 I
I  .320 I 7.198E -1I     33I************<0>                                  I
I  .340 I 6.336E -1I     27I**********<*0>                                   I
I  .360 I 5.355E -1I     36I************< 0>                                 I
I  .380 I 4.975E -1I     21I********<*0>                                     I
I  .400 I 5.382E -1I     23I*********<0*>                                    I
I  .420 I 4.678E -1I     23I*********<0*>                                    I
I  .440 I 5.533E -1I     25I*********<*0>                                    I
I  .460 I 5.703E -1I     29I***********<0>                                   I
I  .480 I 5.827E -1I     29I***********<0>                                   I
I  .500 I 5.571E -1I     26I**********<0*>                                   I
I  .520 I 5.209E -1I     30I***********<0 >                                  I
I  .540 I 9.908E -1I     34I************<0>**                                I
I  .560 I 5.112E -1I     23I*********<0*>                                    I
I  .580 I 5.276E -1I     31I***********<*0>                                  I
I  .600 I 4.570E -1I     30I***********<0 >                                  I
I  .620 I 6.059E -1I     32I***********<*0>                                  I
I  .640 I 5.438E -1I     24I*********<*0>                                    I
I  .660 I 5.402E -1I     29I***********<0>                                   I
I  .680 I 6.301E -1I     44I**************<0>                                I
I  .700 I 8.053E -1I     25I*********<*0>***                                 I
I  .720 I 6.336E -1I     26I**********<0*>                                   I
I  .740 I 8.339E -1I     33I************<0>*                                 I
I  .760 I 9.425E -1I     40I*************<0*>                                I
I  .780 I 9.959E -1I     43I**************<0>                                I
I  .800 I 1.144E  0I     66I*****************<0>                             I
I  .820 I 1.454E  0I     59I****************<0>*                             I
I  .840 I 1.528E  0I     78I******************<0>                            I
I  .860 I 1.966E  0I     82I*******************<>*                           I
I  .880 I 2.564E  0I    125I**********************<>                         I
I  .900 I 2.735E  0I    152I************************<>                       I
I  .920 I 3.601E  0I    168I************************<>                       I
I  .940 I 5.259E  0I    279I****************************<>                   I
I  .960 I 9.247E  0I    470I********************************<>               I
I  .980 I 5.871E  1I   2940I*********************************************<>  I
I  E  0 I  .000E  0I      0I                                                 I
+-------+----------+-------+----------------+----------------+---------------+
    x     d(Sig/dx)  dN/dx    "0" : Generated Events.( Arbitrary unit in Log )
```

Output 3.10 An example of the original histogram

```
    ************* Number of trials to get an event *************
Total =      10095 events   "*" : No. of events in Linear scale.
    x     Lg(dN/dx)  dN/dx  .0E+00      1.5E+03      3.0E+03     4.5E+03
+-------+----------+------+------------+------------+-----------+-----------+
I   .100 I 4.668E  3I   4668I***************************************0000000   I
I   .200 I 2.174E  3I   2174I*******************00000000000000000000000       I
I   .300 I 1.131E  3I   1131I**********00000000000000000000000000000          I
I   .400 I 6.180E  2I    618I******00000000000000000000000000000              I
I   .500 I 3.180E  2I    318I***00000000000000000000000000000                 I
I   .600 I 2.220E  2I    222I**0000000000000000000000000000                   I
I   .700 I 1.580E  2I    158I**00000000000000000000000000                     I
I   .800 I 9.100E  1I     91I*000000000000000000000000                        I
I   .900 I 9.200E  1I     92I*000000000000000000000000                        I
I  1.000 I 7.000E  1I     70I*00000000000000000000000                         I
I  1.100 I 4.800E  1I     48I*000000000000000000000                           I
I  1.200 I 4.400E  1I     44I*00000000000000000000                            I
I  1.300 I 3.500E  1I     35I*0000000000000000000                             I
I  1.400 I 2.500E  1I     25I*00000000000000000                               I
I  1.500 I 3.100E  1I     31I*000000000000000000                              I
I  1.600 I 1.900E  1I     19I*000000000000000                                 I
I  1.700 I 1.900E  1I     19I*000000000000000                                 I
I  1.800 I 2.000E  1I     20I*0000000000000000                                I
I  1.900 I 2.300E  1I     23I*00000000000000000                               I
I  2.000 I 1.100E  1I     11I*0000000000000                                   I
I  2.100 I 1.700E  1I     17I*000000000000000                                 I
I  2.200 I 1.300E  1I     13I*0000000000000                                   I
I  2.300 I 1.400E  1I     14I*00000000000000                                  I
I  2.400 I 1.300E  1I     13I*0000000000000                                   I
I  2.500 I 9.000E  0I      9I*00000000000                                     I
I  2.600 I 6.000E  0I      6I*000000000                                       I
I  2.700 I 7.000E  0I      7I*0000000000                                      I
I  2.800 I 1.100E  1I     11I*0000000000000                                   I
I  2.900 I 8.000E  0I      8I*00000000000                                     I
I  3.000 I 4.000E  0I      4I*0000000                                         I
I  3.100 I 9.000E  0I      9I*00000000000                                     I
I  3.200 I 2.000E  0I      2I*000                                             I
I  3.300 I 4.000E  0I      4I*0000000                                         I
I  3.400 I 1.200E  1I     12I*0000000000000                                   I
I  3.500 I 6.000E  0I      6I*000000000                                       I
I  3.600 I 3.000E  0I      3I*00000                                           I
I  3.700 I 4.000E  0I      4I*0000000                                         I
I  3.800 I 3.000E  0I      3I*00000                                           I
I  3.900 I 2.000E  0I      2I*000                                             I
I  4.000 I 5.000E  0I      5I*00000000                                        I
I  4.100 I 5.000E  0I      5I*00000000                                        I
I  4.200 I 1.000E  0I      1I0                                                I
I  4.300 I 2.000E  0I      2I*000                                             I
I  4.400 I 4.000E  0I      4I*0000000                                         I
I  4.500 I 7.000E  0I      7I*0000000000                                      I
I  4.600 I 4.000E  0I      4I*0000000                                         I
I  4.700 I 2.000E  0I      2I*000                                             I
I  4.800 I  .000E  0I      0I                                                 I
I  4.900 I 4.000E  0I      4I*0000000                                         I
I  5.000 I 2.000E  0I      2I*000                                             I
I   E  1 I 9.500E  1I     95I*000000000000000000000000                        I
+-------+----------+------+------------+------------+-----------+-----------+
    x     Lg(dN/dx)  dN/dx 1.0E+00      1.0E+01     1.0E+02      1.0E+03
                            "0" : No. of Events in Log. scale.
```

Output 3.11 The number of trials distribution

Appendix A

Kinematics

Below is the list of built-in kinematics in the system. This will includes more candidates in the future. The name of section shows the code number of the kinematics.

Normally, if one uses built-in kinematics, the cross section is given in unit of pb.

code number	contents
`2001`	2-body $\rightarrow$ 2 body in CM frame No t-channel singularity.
`2002`	2-body $\rightarrow$ 2 body in CM frame With t-channel singularity. (forward peak)
`2003`	2-body $\rightarrow$ 2 body in CM frame With t-channel singularity. (forward-backward peak)
`3001`	2-body $\rightarrow$ 3 body in CM frame , Sequential decay type $1+2 \rightarrow 3+(4+5) \rightarrow 3+4+5$. Simple phase space.
`3002`	2-body $\rightarrow$ 3 body in CM frame , Sequential decay type $1+2 \rightarrow 3+(4+5) \rightarrow 3+4+5$. Particle-3 is a radiative photon from initial particles.
`3003`	2-body $\rightarrow$ 3 body in CM frame , Sequential decay type $1+2 \rightarrow 3+(4+5) \rightarrow 3+4+5$. Particle 4 and 5 make a resonance.
`3004`	2-body $\rightarrow$ 3 body in CM frame , Sequential decay type $1+2 \rightarrow 3+(4+5) \rightarrow 3+4+5$. Invariant mass of 4 and 5 behaves $\sim 1/M^2$.
`3005`	2-body $\rightarrow$ 3 body in CM frame , Particle 5 is produced at central by 'fusion'. $1 \rightarrow 3+A, \quad 2 \rightarrow 4+B, \quad A+B \rightarrow 5$.
`3006`	2-body $\rightarrow$ 3 body in CM frame , Radiative processes $1+2 \rightarrow 3(\gamma)+4+5$, both initial and final radiation can be treated.
`3007`	2-body $\rightarrow$ 3 body in CM frame , Double-radiative processes $1+2 \rightarrow 3(\gamma)+4(\gamma)+5$
`3008`	2-body $\rightarrow$ 3 body in CM frame , Three phton processes $1+2 \rightarrow 3(\gamma)+4(\gamma)+5(\gamma)$
`3009`	2-body $\rightarrow$ 3 body in CM frame , General purpos kinematics. It can be used for almost all processes except radiative one.
`4001`	2-body $\rightarrow$ 4 body in CM frame, a pair of sequential decay type $1+2 \rightarrow (3+4)+(5+6) \rightarrow 3+4+5+6$ No t-channel singularity.
`4002`	2-body $\rightarrow$ 4 body in CM frame, a pair of sequential decay type $1+2 \rightarrow (3+4)+(5+6) \rightarrow 3+4+5+6$ With t-channel singularity.
`4003`	2-body $\rightarrow$ 4 body in CM frame, 'fusion' type $1+2 \rightarrow (3+A)+(4+B); A+B \rightarrow 5+6$
`4004`	2-body $\rightarrow$ 4 body in CM frame, General purpos kinematics.

For the 4-vector notation, we use (p_x, p_y, p_z, E) ordering. Generally, the kinematics assumes

$$p_1 + p_2 \to p_3 + p_4 + \cdots \qquad (2-\text{body scattering}),$$
$$p_1 \to p_2 + p_3 + p_4 + \cdots \qquad (\text{decay of a particle}),$$

and so on. Here the assignment of particles keeps the order of particles in the input data at the stage of graph generation.

The Lorentz invariant phase space for final n-body ($A \to 1+2+3+\cdots+n$) is defined by

$$d\tilde{\Gamma}_n = (2\pi)^4 \delta^{(4)}\left(\sum_{in} p - \sum_{out}^{n} p\right) \prod_{out}^{n} \frac{d^3 p_j}{(2\pi)^3 2E_j} \equiv \frac{1}{(2\pi)^{3n-4}} d\Gamma_n$$

$$d\Gamma_n = \delta^{(4)}\left(\sum_{in} p - \sum_{out}^{n} p\right) \prod_{out}^{n} \frac{d^3 p_j}{2E_j}.$$

The chain relation for the phase space is useful($0 < k < n-1$):

$$d\Gamma_n(A \to 1+2+3+\cdots) = d\Gamma_{k+1}(A \to 1+\cdots+k+q)\frac{dQ^2}{2\pi} d\Gamma_{n-k}(q \to (k+1)+(k+2)+\cdots)$$

where $q^2 = Q^2$.

When p_b and p_c are in the center-of-mass system and $p_a = p_b + p_c$,i.e., $\vec{p_a} = \vec{0}$, we use the following notations

$$d\Gamma_2 = d\Gamma_{\mathrm{CM}}(a;bc) = \delta^{(4)}(p_a - p_b - p_c)\frac{d^3 p_b}{2E_b}\frac{d^3 p_c}{2E_c}$$

and write it by angular variables:

$$d\Gamma_{\mathrm{CM}}(a;bc) = \frac{\beta(a;bc)}{8} d\Omega_{\mathrm{CM}}(a;bc) = \frac{\beta(a;bc)}{8} d\cos\theta_{b,(bc)} d\phi_{b,(bc)}$$

Here

$$\begin{aligned} \beta(a;bc) &= \frac{2P}{E_a} \\ &= \frac{1}{E_a}\sqrt{(E_a+m_b+m_c)(E_a-m_b-m_c)(E_a+m_b-m_c)(E_a-m_b+m_c)} \end{aligned}$$

and the suffix (bc) denotes that the angles θ_b, ϕ_b are defined in the center-of-mass system. Angles in the laboratory frame have no suffix.

A.1 2001

Kinematics `2002` and `2003` is variation of this one and the user can also consult them in the following sections.

Description of the kinematics

This is the simple kinematics for 2 to 2 process in the center-of mass system. Integration variables are naturally the polar angle θ and azimuthal angle ϕ with respect to the incoming particles.

If there is no strong peak to some directions, e.g., to the forward direction, this works well. Also user can introduce forward and backward angle cutoff as options.

We use the frame where incoming particles collide along z-axis, so that the momenta of initial particles are assigned as follows:

$$\begin{array}{ll} \text{Particle-1} & (0,0,+P,E_1) \\ \text{Particle-2} & (0,0,-P,E_2) \end{array}$$

where P is the positive value determined by m_1, m_2, W. The angles θ and ϕ represent the direction of particle-3. Here, relative velocity v_{rel} is assumed to be 2 for the collision highly relativistic particles. If colliding particles are slow, the user needs modification in `kinit.f`.

Phase space is in the center-of-mass system and it is given by

$$d\tilde{\Gamma}_2 = \frac{1}{(2\pi)^2} d\Gamma_{\rm CM}(12;34) = \frac{\beta(12;34)}{8(2\pi)^2} d\cos\theta d\phi.$$

Meaning of `X()` for BASES integration

$$\begin{array}{ll} \texttt{X(1)} & \cos\theta = 2\texttt{X(1)} - 1 \\ \texttt{X(2)} & \phi = 2\pi\texttt{X(2)} \end{array}$$

Options in `kinit.f`

Default value is shown in parenthesis.

1. Physical parameter section
   - `W` R*8 (200.0) : Center of mass energy.
   - `COSCUT(1)` R*8 (-1.0) : Minimum of $\cos\theta$.
   - `COSCUT(2)` R*8 (+1.0) : Maximum of $\cos\theta$.
2. Physics control section
   - `NDIM` I*4 (2) : If you want to suppress ϕ-integration, replace the definition of `NDIM` by the following:

     ```
     NDIM = 1
     ```

     Then ϕ is fixed to 0.0.
3. `BASES` control section

- `ITMX1` I*4 (5) : These three values control `BASES` integration. See document of `BASES` for details.
- `ITMX2` I*4 (5) :
- `NCALL` I*4 (5000) :
- `NX` I*4 (50) : Value `NX` control the histograms.

A.2 2002

This is the kinematics similar to `2001` and only the difference is quoted here.

Description of the kinematics

In this kinematics, with flag `ICOST=+1`, the cross section is assumed to have $\sim 1/t$ singularity, where $t = (p1 - p3)^2$. New integration variable is introduced as

$$\begin{aligned} D &= -t + m_1^2, \\ &= 2(E_1E_3 - PP_3\cos\theta) - m_3^2. \end{aligned}$$

The variable is transformed into

$$\begin{aligned} dD &= 2PP_3 d\cos\theta, \\ dD/D &= d(\log D). \end{aligned}$$

Then the phase space in `2001` is replaced by

$$\begin{aligned} d\cos\theta &= \frac{D}{2PP_3} d(\log D), \\ &= \frac{D}{2PP_3}\log(D_{max}/D_{min})d\eta \end{aligned}$$

where $D = D_{min}(D_{max}/D_{min})^\eta$ for $0 < \eta < 1$.

Meaning of `X()` for BASES integration

`X(1)`	Momentum transfer in t-channel, $D = -(p1 - p3)^2 + m_1^2$. $D = D_{min}(D_{max}/D_{min})^{X(1)}$
`X(2)`	the same as in `2001`.

Options in `kinit.f`

Physics control section

- `ICOST` I*4 (1) : Treatment of $\cos\theta$.

A.3 2003

This is the kinematics similar to `2001` and only the difference is quoted here.

Description of the kinematics

In this kinematics, with flag `ICOST=-1`, the cross section is assumed to have $\sim 1/t$ and $\sim 1/u$ singularity, where $t = (p1 - p3)^2$ and $u = (p1 - p4)^2$. New integration variable , D, is introduced as the same as `2002`, but $\cos\theta$ is symmetrized around 90°.

Meaning of `X()` for BASES integration

`X(1)`	Momentum transfer in t-channel, $D = -(p1 - p3)^2 + m_1^2$.
	$D = D_{min}(D_{max}/D_{min})^{X(1)}$, D is symmetrized around X(1)=0.5.
`X(2)`	the same as in `2001`.

Options in `kinit.f`

Physics control section

- `ICOST` I*4 (1) : Treatment of $\cos\theta$.

A.4 3001

Kinematics `3002`, `3003`, `3004`, and `3009` are variation of this one and the user can also consult them in the following sections.

Description of the kinematics

This is the kinematics for 2 to 3 process in the center-of mass system. Here the final state first splits into particle-3 and the system of particles 4 and 5. After that the latter decays into particle-4 and particle-5.

$$1 + 2 \longrightarrow 3 + q \qquad q \longrightarrow 4 + 5$$

Integration variables are the polar angle θ and azimuthal angle ϕ for the first split, those angles for the second split, and the invariant mass of particles 4 and 5. Angles for the first split are defined with respect to the incoming particles and those for the second split are defined in the center-of-mass system of particles 4 and 5 with respect to the momentum direction of the system.

If there is a mass singularity for the two of particles in the final state, it is recommended to assign the two particles to particles 4 and 5. Also user can introduce cutoff for angles and minimum energies as options.

For the 4-vector notation, we use (p_x, p_y, p_z, E) ordering. We use the frame where incoming particles collide along z-axis, so that the momenta of initial particles are assigned as follows:

Particle-1	$(0, 0, +P, E_1)$
Particle-2	$(0, 0, -P, E_2)$

where P is the positive value determined by m_1, m_2, W. Here, relative velocity v_{rel} is assumed to be 2 for the collision highly relativistic particles. If colliding particles are slow, the user needs modification in `kinit.f`.

The angles θ_3 and ϕ_3 represent the direction of particle-3. Invariant mass of 4 and 5, Q^2 is another variable.

$$Q^2 = (p_4 + p_5)^2$$

In the center of mass system of particles 4 and 5, angles $\theta_{4,(45)}$ and $\phi_{4,(45)}$ represent the direction of particle-4. The system of particles 4 and 5 are boosted backward to the momentum direction of particle-3.

Phase space is in the center-of-mass system and it is given by

$$\begin{aligned} d\tilde{\Gamma}_3 &= \frac{1}{2\pi} d\Gamma_2(1 + 2 \to 3 + q) dQ^2 d\Gamma_2(q \to 4 + 5) \\ &= \frac{\beta(12; 3q)\beta(q; 45)}{8^2(2\pi)^5} d\cos\theta_3 d\phi_3 dQ^2 d\cos\theta_{4,(45)} d\phi_{4,(45)}. \end{aligned}$$

Here there is no singular behavior, and `IRESN=0` and `ICOS3=0` are assumed.

Meaning of `X()` for BASES integration

`X(1)`	$\cos\theta_{4,(45)} = 2$`X(1)`$- 1$ Polar angle in the CM system of particles 4 and 5.
`X(2)`	$\phi_{4,(45)} = 2\pi$`X(2)` Azimuthal angle in the CM system of particles 4 and 5.
`X(3)`	$\cos\theta_3 = 2$`X(3)`$- 1$ Polar angle of particle-3
`X(4)`	Invariant mass of particles 4 and 5, Q^2. Normal. $Q^2 = Q^2_{min} + (Q^2_{max} - Q^2_{min})$`X(4)`
`X(5)`	$\phi_3 = 2\pi$`X(5)` Azimuthal angle of particle-3

Options in `kinit.f`

Default value is shown in parenthesis.

1. Physical parameter section

- `W` R*8 (200.0) : Center of mass energy.
- `COSCUT(1,1)` R*8 (-1.0) : Minimum of $\cos\theta_3$.
- `COSCUT(2,1)` R*8 (+1.0) : Maximum of $\cos\theta_3$.
- `COSCUT(1,2)` R*8 (-1.0) : Minimum of $\cos\theta_4$. Here, θ_4 is the polar angle of particle-4 with respect to the beam axis.
- `COSCUT(2,2)` R*8 (+1.0) : Maximum of $\cos\theta_4$.
- `COSCUT(1,3)` R*8 (-1.0) : Minimum of $\cos\theta_5$. Here, θ_5 is the polar angle of particle-5 with respect to the beam axis.
- `COSCUT(2,3)` R*8 (+1.0) : Maximum of $\cos\theta_5$.
- `ENGYCT(1,1)` R*8 (m_3) : Minimum of E_3.
- `ENGYCT(2,1)` R*8 (W) : Maximum of E_3.
- `ENGYCT(1,2)` R*8 (m_4) : Minimum of E_4.
- `ENGYCT(2,2)` R*8 (W) : Maximum of E_4.
- `ENGYCT(1,3)` R*8 (m_5) : Minimum of E_5.
- `ENGYCT(2,3)` R*8 (W) : Maximum of E_5.
- `AMASCT(1)` R*8 ($m_4 + m_5$) : Minimum of Q. Q is the mass of the system of particles 4 and 5.
- `AMASCT(2)` R*8 ($W - m_3$) : Maximum of Q.
- `ARESNS(1)` R*8 (0.0) : Mass of resonance, this and the next parameter are meaningful only when `IRESN=+1`.
- `ARESNS(2)` R*8 (0.0) : Width of resonance.

2. Physics control section

   - `NDIM` I*4 (5) : If you want to suppress ϕ_3-integration, replace the definition of `NDIM` by the following:

     ```
     NDIM = 4
     ```

     Then ϕ_3 is fixed to 0.0.
   - `IRESN` I*4 (0) : Treatment of Q^2. (Do not change.)
   - `ICOS3` I*4 (0) : Treatment of θ_3. (Do not change.)

3. `BASES` control section

   - `ITMX1` I*4 (5) : These three values control `BASES` integration. See document of `BASES` for details.
   - `ITMX2` I*4 (5) :
   - `NCALL` I*4 (5000) :
   - `NX` I*4 (50) : Value `NX` control the histograms.

Related modules

WTOLAB

A.5 3002

This is the kinematics similar to 3001 and only the difference is quoted here.

Description of the kinematics

Here, the particle-3 favors the beam direction, e.g, the photon radiated from initial particles. So in this case, $m_3 = 0$ and $m_1 = m_2$ is assumed.

When the flag ICOS3=1, the kinematics is changed to the case for the radiation of particle-3 along beam axis. If ICOS3=1, the flag IRESN is neglected. Here, the energy of particle-3, E_3, is used instead of Q^2 using

$$Q^2 = W^2 - 2WE_3 + m_3^2 \quad (m_3 = 0)$$

and E_3 is converted into

$$\frac{dE_3}{E_3} = d(\log E_3)$$

to absorb the $1/E_3$ behavior(soft singularity) which appears in the photon radiation.

The angle θ_3 is changed to absorb the collinear singularity, which appears in the form of

$$\frac{1}{D_1 D_2}, \quad D_1 = 2p_1p_3, \quad D_2 = 2p_2p_3.$$

We introduce a variable

$$\tau = \frac{1 + v\cos\theta_3}{1 - v\cos\theta_3} \quad , \quad v = \sqrt{1 - 4m_1^2/W^2},$$

$$y = \frac{1}{4}\left(2 + \frac{\log\tau}{\log\xi}\right) , \quad \xi = \sqrt{\frac{1+v}{1-v}}.$$

Here the correspondence is that $y = (0, 1)$ to $\cos\theta_3 = (-1, 1)$ and

$$D_1 = \frac{4E_1E_3}{1+\tau}, \quad D_2 = D_1\tau.$$

Then the phase space in 3001 is replaced by

$$dQ^2 = 2WE_3\log(E_{3,max}/E_{3,min})dx_4$$

where $E_3 = E_{3,min}(E_{3,max}/E_{3,min})^{x4}$ and

$$d\cos\theta_3 = \log(\xi)\frac{D_1D_2}{E_3^2}\frac{1}{2E_1P_1}dy.$$

Meaning of `X()` for BASES integration

`X(1)`, `X(2)`, `X(5)` are the same in `3001`.

`X(3)`	Related to $\cos\theta_3$ through the variable y above. $y = y_{min} + (y_{max} - y_{min})\texttt{X(3)}$
`X(4)`	Energy of particle-3. $E_3 = E_{3,min}(E_{3,max}/E_{3,min})^{\texttt{X(4)}}$

Options in `kinit.f`

Physics control section

- `IRESN` I*4 (0) : Treatment of Q^2. (Do not change.)
- `ICOS3` I*4 (1) : Treatment of θ_3. (Do not change.)

A.6 3003

This is the kinematics similar to `3001` and only the difference is quoted here.

Description of the kinematics

When the flag `IRESN=+1`, the amplitude is assumed to have a resonance behavior as

$$\frac{1}{(Q^2 - M_R^2)^2 + M_R^2\Gamma_R^2}.$$

Then the variable is transformed into

$$\frac{dQ^2}{(Q^2 - M_R^2)^2 + M_R^2\Gamma_R^2} = \frac{dt}{M_R\Gamma_R} \qquad \left(t = \arctan\frac{Q^2 - M_R^2}{M_R\Gamma_R}\right)$$

where M_R and Γ_R are the mass and width of the resonance.

Then the phase space in `3001` is replaced by

$$dQ^2 = \frac{(Q^2 - M_R^2)^2 + M_R^2\Gamma_R^2}{M_R\Gamma_R}dt.$$

Meaning of `X()` for BASES integration

`X(1)`, `X(2)`, `X(3)`, `X(5)` are the same in `3001`.

`X(4)`	Invariant mass of particles 4 and 5, Q^2. Resonance. $t = \arctan((Q^2 - M_R^2)/(M_R\Gamma_R))$ $t = t_{min} + (t_{max} - t_{min})\texttt{X(4)}$ $Q^2 = M_R^2 + M_R\Gamma_R\tan t$

Options in `kinit.f`

Physical parameter section

- `ARESNS(1)` R*8 (M_W) : Mass of resonance, M_R.
- `ARESNS(2)` R*8 (Γ_W) : Width of resonance, Γ_R.

Physics control section

- `IRESN` I*4 (1) : Treatment of Q^2. (Do not change.)
- `ICOS3` I*4 (0) : Treatment of θ_3. (Do not change.)

A.7 3004

This is the kinematics similar to `3001` and only the difference is quoted here.

Description of the kinematics

When the flag `IRESN=-1`, the amplitude is assumed to contain a pole as $1/Q^2$, and the variable is transformed into

$$\frac{dQ^2}{Q^2} = d(\log Q^2).$$

Then the phase space in `3001` is replaced by

$$dQ^2 = Q^2 d(\log Q^2) = Q^2 \log(Q^2_{max}/Q^2_{min})dt$$

where $Q^2 = Q^2_{min}(Q^2_{max}/Q^2_{min})^t$ for $0 < t < 1$.

Meaning of `X()` for BASES integration

`X(1)`, `X(2)`, `X(3)`, `X(5)` are the same in `3001`.

`X(4)`	Invariant mass of particles 4 and 5, Q^2. $1/Q^2$ behavior assumed. $Q^2 = Q^2_{min}(Q^2_{max}/Q^2_{min})^{\texttt{X(4)}}$

Options in `kinit.f`

Physics control section

- `IRESN` I*4 (-1) : Treatment of Q^2. (Do not change.)
- `ICOS3` I*4 (0) : Treatment of θ_3. (Do not change.)

A.8 3005

Description of the kinematics

This is the kinematics for 2 to 3 process in the center-of mass system. Here a particle 3 emmit particle A and a particle 4 emmit particle B. After that particles A and B collide into particles 5;

$$1 \longrightarrow 3 + A \quad , \quad 2 \longrightarrow 4 + B \quad , \quad A + B \longrightarrow 5$$

Integration variables are the polar angle θ and energy of particles 3 and 4, and azimuthal angle ϕ of particles 3. Angles of 3 and 4 are defined with respect to the incoming particles.

For the 4-vector notation, we use (p_x, p_y, p_z, E) ordering. We use the frame where incoming particles collide along z-axis, so that the momenta of initial particles are assigned as follows:

$$\begin{array}{ll} \text{Particle-1} & (0, 0, +P, E_1) \\ \text{Particle-2} & (0, 0, -P, E_2) \end{array}$$

where P is the positive value determined by m_1, m_2, W. Here, relative velocity v_{rel} is assumed to be 2 for the collision highly relativistic particles. If colliding particles are slow, the user needs modification in `kinit.f`.

New integration variable is intruduced as

$$\begin{aligned} Q^2 &= -(p_1 - p_3)^2 \\ &= 2p_1p_3 - m_1^2 - m_3^2 \\ q_0 &= E_1 - E_3 \end{aligned}$$

Phase space is in the center-of-mass system and it is given by

$$d\tilde{\Gamma}_3 = \frac{1}{8(2\pi)^5} \frac{dE_3 d\cos\theta_3 dE_4 d\cos\theta_4 d\phi_3}{\sin\theta_3 \sin\theta_4 |\sin\phi_4|}$$

In this kinematics, with `ICOS3=1` the cross section is assumed to have $\sim 1/Q^2$ singularity. The variable is transformed into

$$\begin{array}{lll} dQ^2 & = 2PP_3 d\cos\theta_3, & (\text{for ICOS3=0}) \\ dQ^2/Q^2 & = d(\log Q^2). & (\text{for ICOS3=1}) \end{array}$$

Then phase space is replaced by

$$\begin{array}{lll} d\cos\theta_3 & = \frac{1}{2PP_3} dQ^2, & (\text{for ICOS3=0}) \\ & = \frac{Q^2}{2PP_3} \log(Q^2_{max}/Q^2_{min}) d\eta & (\text{for ICOS3=1}) \end{array}$$

where $Q^2 = Q^2_{min}(Q^2_{max}/Q^2_{min})^\eta$ for $0 < \eta < 1$.

Meaning of `X()` for BASES integration

`X(1)`	Energy of $q = p_1 - p_3$
`X(2)`	$Q^2 = -q^2$
	$Q^2 = Q^2_{min} + (Q^2_{max} - Q^2_{min})X(2)$ for ICOS3=0
	$Q^2 = Q^2_{min}(Q^2_{max}/Q^2_{min})^{X(2)}$ for ICOS3=1
`X(3)`	$\cos\theta_4 = 2\mathtt{X(3)} - 1$
	Polar angle of particle-4
`X(4)`	Energy of particle-4
`X(5)`	$\phi_3 = 2\pi\mathtt{X(5)}$
	Azimuthal angle of particle-3

Options in `kinit.f`

Default value is shown in parenthesis.

1. Physical parameter section

   - `W` R*8 (200.0) : Center of mass energy.
   - `COSCUT(1,1)` R*8 (-1.0) : Minimum of $\cos\theta_3$.
   - `COSCUT(2,1)` R*8 (+1.0) : Maximum of $\cos\theta_3$.
   - `COSCUT(1,2)` R*8 (-1.0) : Minimum of $\cos\theta_4$. Here, θ_4 is the polar angle of particle-4 with respect to the beam axis.
   - `COSCUT(2,2)` R*8 (+1.0) : Maximum of $\cos\theta_4$.
   - `COSCUT(1,3)` R*8 (-1.0) : Minimum of $\cos\theta_5$. Here, θ_5 is the polar angle of particle-5 with respect to the beam axis.
   - `COSCUT(2,3)` R*8 (+1.0) : Maximum of $\cos\theta_5$.
   - `ENGYCT(1,1)` R*8 (m_3) : Minimum of E_3.
   - `ENGYCT(2,1)` R*8 ($\frac{s+m_3-(m_4+m_5)^2}{2W}$) : Maximum of E_3.
   - `ENGYCT(1,2)` R*8 (m_4) : Minimum of E_4.
   - `ENGYCT(2,2)` R*8 ($\frac{s+m_4-(m_5+m_3)^2}{2W}$) : Maximum of E_4.
   - `ENGYCT(1,3)` R*8 (m_5) : Minimum of E_5.
   - `ENGYCT(2,3)` R*8 ($\frac{s+m_5-(m_3+m_4)^2}{2W}$) : Maximum of E_5.

2. Physics control section

   - `ICOS3` I*4 (0) : Treatment of θ_3.

3. `BASES` control section

   - `ITMX1` I*4 (5) : These three values control `BASES` integration. See document of `BASES` for details.

- `ITMX2` I*4 (5) :
- `NCALL` I*4 (5000) :
- `NX` I*4 (50) : Value `NX` control the histograms.

A.9 3006

This is the kinematics for radiative processes.

Description of the kinematics

This is the kinematics for 2 to 3 process in the center-of mass system. A particle 3 is assumed to be a photon. To treat both initial and final state radiations, a phase space with respect to photon angles is divided into three regions;

$$\begin{aligned}
\sigma &= \int_S \frac{d\sigma}{d\Omega_3} d\Omega_3, \\
&= \int_{S_{12}} \frac{d\sigma}{d\Omega_3} d\Omega_3 + \int_{S_4} \frac{d\sigma}{d\Omega_3} d\Omega_3 + \int_{S_5} \frac{d\sigma}{d\Omega_3} d\Omega_3, \\
S_{12} &= \{\hat{P}_3 | min\{\theta_{13}, \theta_{23}, \theta_{43}, \theta_{53}\} = \theta_{13} \ or \ \theta_{23}\}, \\
S_4 &= \{\hat{P}_3 | min\{\theta_{13}, \theta_{23}, \theta_{43}, \theta_{53}\} = \theta_{34}\}, \\
S_5 &= \{\hat{P}_3 | min\{\theta_{13}, \theta_{23}, \theta_{43}, \theta_{53}\} = \theta_{35}\},
\end{aligned}$$

where $\theta_{ij} = cos^{-1}(\hat{P}_i \cdot \hat{P}_i)$ and $\hat{P}_i$ is a unit (three) vector along a three momentum of a particle-i.

In the region S_{12}

In the region S_{12}, the kinematics 3003 is used. Difference is quoted here.

Options in `kinit.f`

Default value is shown in parenthesis.

1. Physical parameter section
   - `COSOPN` R*8 (1) : opening angle cut between photon and particle 4,5
2. Physics control section
   - `IFRAD` I*4 (3) : 1=initial state radiadiation, 2=final state radiation,3=initial+final state radiation.

In the region S_4

In the region S_4, the final state first splits into particle-5 and the system particle-5 and photon (particle-3). After that the latter decays into particle-5 and particle-3.

$$1+2 \longrightarrow 5+q \quad , \quad q \longrightarrow 3(photon)+5$$

Integration variables are the polar angle θ and azimuthal angle ϕ for the first split, those angles for the second split, and the invariant mass of particles 3 and 5. Angles for the first split are defined with respect to the incoming particles and those for the second split are defined in the center-of-mass system of particles 4 and 5 with respect to the momentum direction of the system.

For the 4-vector notation, we use (p_x, p_y, p_z, E) ordering. We use the frame where incoming particles collide along z-axis, so that the momenta of initial particles are assigned as follows:

$$\begin{array}{ll} \text{Particle-1} & (0,0,+P,E_1) \\ \text{Particle-2} & (0,0,-P,E_2) \end{array}$$

where P is the positive value determined by m_1, m_2, W. Here, relative velocity v_{rel} is assumed to be 2 for the collision highly relativistic particles. If colliding particles are slow, the user needs modification in `kinit.f`.

The angles θ_4 and ϕ_4 represent the direction of particle-4. Invariant mass of 3 and 5, Q^2 is another variable.

$$Q^2 = (p_3+p_5)^2$$

In the center of mass system of particles 3 and 5, angles $\theta_{3,(35)}$ and $\phi_{3,(35)}$ represent the direction of particle-3. The system of particles 3 and 5 are boosted backward to the momentum direction of particle-4.

Phase space is in the center-of-mass system and it is given by

$$\begin{aligned} d\tilde{\Gamma}_3 &= \frac{1}{2\pi} d\Gamma_2(1+2 \to 4+q) dQ^2 d\Gamma_2(q \to 3+5) \\ &= \frac{\beta(12;4q)\beta(q;35)}{8^2(2\pi)^5} d\cos\theta_3 d\phi_3 dQ^2 d\cos\theta_{4,(45)} d\phi_{4,(45)}. \end{aligned}$$

To treat colinear singularity, new variables are introduced;

$$\frac{dQ^2}{Q^2} = d(\log Q^2),$$

then,

$$dQ^2 = Q^2 d(\log Q^2) = Q^2 \log(Q^2_{max}/Q^2_{min}) dt$$

where $Q^2 = Q^2_{min}(Q^2_{max}/Q^2_{min})^t$ for $0 < t < 1$. Moreover,

$$\begin{aligned}
\cos\theta_3^* &= (\frac{\sqrt{Q^2}E_3}{E_3^*} - E_q)/P_q, \\
E_3^* &= \frac{Q^2 - m_5^2}{2\sqrt{Q^2}}, \\
E_q &= \frac{s + Q^2 - m_4^2}{2\sqrt{s}}, \\
P_q &= \sqrt{E_q^2 - Q^2},
\end{aligned}$$

where θ_3^* is polar angle of particle-3 in particle-3 and -5 rest frame, E_3 is an energy of particle-3 in a labframe. E_3 is used as integration valiable instead of θ_3^*, then

$$d(\cos\theta_3^*) = \frac{\sqrt{Q^2}}{E_3^* P_q} d(E_3).$$

Futher modification

$$\begin{aligned}
\frac{dE_3}{E_3} &= d(\log E_3), \\
&= \log(E_3^{max}/E_3^{min})dt,
\end{aligned}$$

for $0 < t < 1$ has been done.

When the flag `icos4=1`, the amplitude is assumed to contain a pole as $1/t^2 = 1/(P_1 - P_4)^2$. The treatment similer to `kinem2002` has been done.

When the flag `icos4=2`, the amplitude is assumed to contain a pole $1/t^2 = 1/(P_1 - P_4)^2$ and $1/u^2 = 1/(P_1 - P_5)^2$. The treatment similer to `kinem2003` has been done.

Meaning of `X()` for BASES integration

`X(1)`	$\cos\theta_4 = 2\mathtt{X(1)} - 1$	:ICOS4=0
	$D = -(p1 - p4)^2 + m_1^2$,	
	$D = D_{min}(D_{max}/D_{min})^{X(1)}$	:ICOS4=1
	D is symmetrized around X(1)=0.5	:ICOS4=2
`X(2)`	$\phi_3 = 2\pi\mathtt{X(2)}$	
	Azimuthal angle in the CM system of particles 3 and 5.	
`X(3)`	$Q^2 = (P_3 + P_5)^2 = Q^2_{min}(Q^2_{max}/Q^2_{min})^{X(3)}$	
`X(4)`	$E_3 = E_3^{min}(E_3^{max}/E_3^{min})^{X(4)}$	
	Energy of particle-3 (photon)	
`X(5)`	$\phi_4 = 2\pi\mathtt{X(5)}$	
	Azimuthal angle of particle-4	

Options in `kinit.f`

Default value is shown in parenthesis.

1. Physical parameter section

   - `W` R*8 (200.0) : Center of mass energy.
   - `COSCUT(1,1)` R*8 (-1.0) : Minimum of $\cos\theta_3$.
   - `COSCUT(2,1)` R*8 (+1.0) : Maximum of $\cos\theta_3$.
   - `COSCUT(1,2)` R*8 (-1.0) : Minimum of $\cos\theta_4$. Here, θ_4 is the polar angle of particle-4 with respect to the beam axis.
   - `COSCUT(2,2)` R*8 (+1.0) : Maximum of $\cos\theta_4$.
   - `COSCUT(1,3)` R*8 (-1.0) : Minimum of $\cos\theta_5$. Here, θ_5 is the polar angle of particle-5 with respect to the beam axis.
   - `COSCUT(2,3)` R*8 (+1.0) : Maximum of $\cos\theta_5$.
   - `ENGYCT(1,1)` R*8 (1.D-3) : Minimum of E_3.
   - `ENGYCT(2,1)` R*8 ($\frac{s+m_3-(m_4+m_5)^2}{2W}$) : Maximum of E_3.
   - `ENGYCT(1,2)` R*8 (m_4) : Minimum of E_4.
   - `ENGYCT(2,2)` R*8 ($\frac{s+m_4-(m_5+m_3)^2}{2W}$) : Maximum of E_4.
   - `ENGYCT(1,3)` R*8 (m_5) : Minimum of E_5.
   - `ENGYCT(2,3)` R*8 ($\frac{s+m_5-(m_3+m_4)^2}{2W}$) : Maximum of E_5.
   - `COSOPN)` R*8 (1) : opening angle cut between photon and particle 4,5

2. Physics control section

   - `ICOS4` I*4 (0) : Treatment of θ_4.
   - `IFRAD` I*4 (3) : 1=initial state radiadiation, 2=final state radiation,3=initial+final state radiation.

3. `BASES` control section

   - `ITMX1` I*4 (5) : These three values control `BASES` integration. See document of `BASES` for details.
   - `ITMX2` I*4 (5) :
   - `NCALL` I*4 (5000) :
   - `NX` I*4 (50) : Value `NX` control the histograms.

In the region S_5

$4 \leftrightarrow 5$ of previous section.

A.10 3007

This is the kinematics for double-radiative processes. It is similar to `3006` and only the difference is quoted here

Description of the kinematics

This is the kinematics for 2 to 3 process in the center-of mass system. Particle 3 and 4 are assumed to be photons. A energy ordaring is required;

$$E_3 < E_4$$

A.11 3008

This is the kinematics for three-photon process. It is similar to `3006` and only the difference is quoted here

Description of the kinematics

This is the kinematics for 2 to 3 process in the center-of mass system. Particle 3,4 and 5 are assumed to be photons. A energy ordaring is required;

$$E_3 < E_4 < E_5$$

A.12 3009

This is the kinematics similar to `3001` and only the difference is quoted here.

Description of the kinematics

This kinematics can treat processes in which particle 4 and 5 come from two independent resonances (for example, $b\bar{b}$ from a Higgs boson and a Z-boson). Moreover it can treat some singularities of angluare distribution of particle-4.

In this kinematics, with flag `ICOS4=1or2`, the cross section is assumed to have flat rapidity distrobution. New integration variable η is introduced as

$$-\log(1+2/\epsilon) < \eta < \log(1+2/\epsilon),$$

$$\epsilon = 2m_4^2/(P_4+P_5)^2.$$

By using this variable, $\cos\theta_4$ can be expressed as;

$$\cos\theta_4 = (1+\epsilon)\tanh\eta.$$

$\cos\theta_4$ is measured with respect to $-P_3$.

With flag `ICOST=-1`, the cross section is assumed to have $\sim 1/t$ singularity, where $t = (p1 - p4)^2$. New integration variable is introduced as

$$\begin{aligned} D &= -t, \\ &= 2(E_1 E_4 - P P_4 \cos\theta_4). \end{aligned}$$

$\cos\theta_4$ is measured with respect to P_1. The variable is transformed into

$$\begin{aligned} dD &= 2PP_4 d\cos\theta, \\ dD/D &= d(\log D). \end{aligned}$$

Then the phase space in `3001` is replaced by

$$\begin{aligned} d\cos\theta_4 &= \frac{D}{2PP_4} d(\log D), \\ &= \frac{D}{2PP_4}\log(D_{max}/D_{min}) d\eta \end{aligned}$$

where $D = D_{min}(D_{max}/D_{min})^\eta$ for $0 < \eta < 1$.

With flag `ICOST=-2`, $\cos\theta_4$ is symmetrized around 90°.

Options in `kinit.f`

Default value is shown in parenthesis.

1. Physical parameter section
   - `ARESNS(1,1)` R*8 (0.0) : Mass of first resonance.
   - `ARESNS(2,1)` R*8 (0.0) : Width of first resonance.
   - `ARESNS(1,2)` R*8 (0.0) : Mass of second resonance.
   - `ARESNS(2,2)` R*8 (0.0) : Width of second resonance.
2. Physics control section
   - `IRESNS` I*4 (0) : Treatment of Q^2. * no-singularity : IRESNS= 0 * narrow resonance (single): IRESNS= 1 * narrow resonance (single) * + 1/Q2 singularit : IRESNS=-1 * narrow resonance (double): IRESNS= 2 * narrow resonance (double) * + 1/Q2 singularit : IRESNS=-2 * Q2=S peak : IRESNS= 3 * 1/Q2 singularity only : IRESNS=-3 * narrow resonance (single) * + Q2=S peak : IRESNS= 4
   - `ICOS4` I*4 (0) : Treatment of θ_4. * no-singularity : ICOS4= 0 * z-axis -¿ -p3 * 1/t singularity : ICOS4= 1 * 1/t + 1/u singularity : ICOS4= 2 * z-axis -¿ p1 * 1/t singularity : ICOS4=-1 * 1/t + 1/u singularity : ICOS4=-2

A.13 4001

Kinematics `4002` is variation of this one and the user can also consult them in the following sections.

Description of the kinematics

This is the kinematics for 2 to 4 process in the center-of mass system. Here the final state first splits into the system of particles 3 and 4 and the system of particles 5 and 6. After that both system decays:

$$1+2 \longrightarrow q_1+q_2 \quad , \quad q_1 \longrightarrow 3+4 \quad , \quad q_2 \longrightarrow 5+6$$

Integration variables are the polar angle θ and azimuthal angle ϕ for the first split, those angles for the two secondary splits, and the invariant masses of q_1 and q_2. Angles for the first split are defined with respect to the incoming particles and those for the secondary splits are defined in their own center-of-mass system with respect to the momentum direction of the system.

If there is a mass singularity for the two of particles in the final state, it is recommended to assign the two particles to form a pair above. Also user can introduce cutoff for angles and minimum energies as options.

For the 4-vector notation, we use (p_x, p_y, p_z, E) ordering. We use the frame where incoming particles collide along z-axis, so that the momenta of initial particles are assigned as follows:

$$\begin{array}{ll} \text{Particle-1} & (0,0,+P,E_1) \\ \text{Particle-2} & (0,0,-P,E_2) \end{array}$$

where P is the positive value determined by m_1, m_2, W. Here, relative velocity v_{rel} is assumed to be 2 for the collision highly relativistic particles. If colliding particles are slow, the user needs modification in **`kinit.f`**.

The angles θ_{q1} and ϕ_{q1} represent the direction of the system of particles 3 and 4. Invariant masses are another variables.

$$Q_1^2 = q_1^2 = (p_3+p_4)^2 \quad , \quad Q_2^2 = q_2^2 = (p_5+p_6)^2$$

In the center of mass system of particles 3 and 4, angles $\theta_{3,(34)}$ and $\phi_{3,(34)}$ represent the direction of particle-3. Similarly, $\theta_{5,(56)}$ and $\phi_{5,(56)}$ are defined. The systems of particles 3 and 4,and 5 and 6 are boosted backward to the laboratory frame later.

Phase space is in the center-of-mass system and it is given by

$$\begin{aligned} d\tilde{\Gamma}_4 &= \frac{1}{(2\pi)^8} d\Gamma_2(1+2 \to q_1+q_2) dQ_1^2 dQ_2^2 d\Gamma_2(q_1 \to 3+4) d\Gamma_2(q_2 \to 5+6) \\ &= \frac{\beta(12;q_1q_2)\beta(q_1;34)\beta(q_2;56)}{8^3(2\pi)^8} d\cos\theta_{q1} d\phi_{q1} \\ &\times \; dQ_1^2 d\cos\theta_{3,(34)} d\phi_{3,(34)} dQ_2^2 d\cos\theta_{5,(56)} d\phi_{5,(56)}. \end{aligned}$$

For the treatment of Q_j^2, one can use one of three ways by setting the flags `IRESNS(j), (j=1,2)`.

- Flat distribution to Q_j^2, as in `3001` for the system of particle 4 and 5. The flag is `IRESNS(j)=0`.
- Resonance distribution to Q_j^2, as in `3003` for the system of particle 4 and 5. The flag is `IRESNS(j)=1`. If this mode is selected, also the user should supply the values of mass and width.
- $1/Q_j^2$ distribution, as in `3004` for the system of particle 4 and 5. The flag is `IRESNS(j)=-1`.

Here there is no singular behavior for t channel, and `ICOSQ3=0` is assumed.

Meaning of `X()` for BASES integration

`X(1)`	Invariant mass of particles 3 and 4, Q_1^2. Dependent on `IRESNS(1)`. The same as `X(4)` in `3001`, `3003`, `3004` for `IRESNS(1)=0,1,-1`.
`X(2)`	Invariant mass of particles 5 and 6, Q_2^2. Dependent on `IRESNS(2)`. The same as `X(4)` in `3001`, `3003`, `3004` for `IRESNS(2)=0,1,-1`.
`X(3)`	$\cos\theta_{3,(34)} = 2\texttt{X(3)} - 1$ Polar angle in the CM system of particles 3 and 4.
`X(4)`	$\phi_{3,(34)} = 2\pi\texttt{X(4)}$ Azimuthal angle in the CM system of particles 3 and 4.
`X(5)`	$\cos\theta_{5,(56)} = 2\texttt{X(5)} - 1$ Polar angle in the CM system of particles 5 and 6.
`X(6)`	$\phi_{5,(56)} = 2\pi\texttt{X(6)}$ Azimuthal angle in the CM system of particles 5 and 6.
`X(7)`	$\cos\theta_{q1} = 2\texttt{X(7)} - 1$ Polar angle of particle-3
`X(8)`	$\phi_{q1} = 2\pi\texttt{X(8)}$ Azimuthal angle of q_1

Options in `kinit.f`

Default value is shown in parenthesis.

1. Physical parameter section
   - `W` R*8 (200.0) : Center of mass energy.
   - `COSCUT(1,1)` R*8 (-1.0) : Minimum of $\cos\theta_3$. This and angles below are all in the laboratory frame.

- `COSCUT(2,1)` R*8 (+1.0) : Maximum of $\cos\theta_3$.
- `COSCUT(1,2)` R*8 (-1.0) : Minimum of $\cos\theta_4$.
- `COSCUT(2,2)` R*8 (+1.0) : Maximum of $\cos\theta_4$.
- `COSCUT(1,3)` R*8 (-1.0) : Minimum of $\cos\theta_5$.
- `COSCUT(2,3)` R*8 (+1.0) : Maximum of $\cos\theta_5$.
- `COSCUT(1,4)` R*8 (-1.0) : Minimum of $\cos\theta_6$.
- `COSCUT(2,4)` R*8 (+1.0) : Maximum of $\cos\theta_6$.
- `ENGYCT(1,1)` R*8 (m_3) : Minimum of E_3.
- `ENGYCT(2,1)` R*8 (W) : Maximum of E_3.
- `ENGYCT(1,2)` R*8 (m_4) : Minimum of E_4.
- `ENGYCT(2,2)` R*8 (W) : Maximum of E_4.
- `ENGYCT(1,3)` R*8 (m_5) : Minimum of E_5.
- `ENGYCT(2,3)` R*8 (W) : Maximum of E_5.
- `ENGYCT(1,4)` R*8 (m_6) : Minimum of E_6.
- `ENGYCT(2,4)` R*8 (W) : Maximum of E_6.
- `AMASCT(1,1)` R*8 ($m_3 + m_4$) : Minimum of Q_1. Q_1 is the mass of the system of particles 3 and 4.
- `AMASCT(2,1)` R*8 ($W - m_5 - m_6$) : Maximum of Q_1.
- `AMASCT(1,2)` R*8 ($m_5 + m_6$) : Minimum of Q_2. Q_2 is the mass of the system of particles 5 and 6.
- `AMASCT(2,2)` R*8 ($W - m_3 - m_4$) : Maximum of Q_2.
- `ARESNS(1,1)` R*8 (0.0) : Mass of resonance, this and the next parameter are meaningful only when `IRESNS(1)=+1`.
- `ARESNS(2,1)` R*8 (0.0) : Width of resonance.
- `ARESNS(1,2)` R*8 (0.0) : Mass of resonance, this and the next parameter are meaningful only when `IRESNS(2)=+1`.
- `ARESNS(2,2)` R*8 (0.0) : Width of resonance.

2. Physics control section

- `NDIM` I*4 (8) : If you want to suppress ϕ_{q1}-integration, replace the definition of `NDIM` by the following:

```
NDIM = 7
```

  Then ϕ_{q1} is fixed to 0.0.
- `IRESNS(1)` I*4 (0) : Treatment of Q_1^2. See the description of kinematics for the meaning.

- IRESNS(2) I*4 (0) : Treatment of Q_2^2. See the description of kinematics for the meaning.
- ICOSQ3 I*4 (0) : Treatment of θ_{q1}. (Do not change.)

3. BASES control section

   - ITMX1 I*4 (5) : These three values control BASES integration. See document of BASES for details.
   - ITMX2 I*4 (5) :
   - NCALL I*4 (5000) :
   - NX I*4 (50) : Value NX control the histograms.

Related modules

WTOLAB

A.14 4002

This is the kinematics similar to 4001 and only the difference is quoted here.

Description of the kinematics

In this kinematics, with flag ICOSQ3=1 , new integration variable is introduced as

$$\begin{aligned} T &= -(p_1 - Q_1)^2, \\ &= 2(E_1 E_{q_1} - P P_{q_1} \cos\theta_{q_1}) - m_1^2 - Q_1^2. \end{aligned}$$

Moreover with flag ICOSQ3=2 , the cross section is assumed to have $\sim 1/T$ singularity. The variable is transformed into

$$\begin{aligned} \mathrm{dT} &= 2PP_{q_1} d\cos\theta_{q_1}, & \text{(for ICOSQ3=1)} \\ \mathrm{dT/T} &= d(\log T). & \text{(for ICOSQ3=2)} \end{aligned}$$

Then the phase space in 4001 is replaced by

$$\begin{aligned} d\cos\theta_{q_1} &= \frac{1}{2PP_{q_1}} d\cos\theta_{q_1}, & \text{(for ICOSQ3=1)} \\ &= \frac{T}{2PP_{q_1}} \log(T_{max}/T_{min}) d\eta & \text{(for ICOSQ3=2)} \end{aligned}$$

where $T = T_{min}(T_{max}/T_{min})^\eta$ for $0 < \eta < 1$.

Meaning of X() for BASES integration

Except for X(7), they are the same in 4001.

X(7) Momentum transfer square, $T = -(p_1 - Q_1)^2$.
$T = T_{min} + (T_{max} - T_{min})\mathtt{X(7)}$ (for ICOSQ3=1)
$T = T_{min}(T_{max}/T_{min})^{\mathtt{X(7)}}$ (for ICOSQ3=2)

Options in `kinit.f`

Physics control section

- `ICOSQ3` I*4 (1) : Treatment of θ_{q1}. (Do not change.)

A.15 4003

Description of the kinematics

This is the kinematics for 2 to 4 process in the center-of mass system. Here a particle 3 emmit particle A and a particle 4 emmit particle B. After that particles A and B collide into particles 5 and 6;

$$1 \longrightarrow 3 + A \quad , \quad 2 \longrightarrow 4 + B \quad , \quad A + B \longrightarrow 5 + 6$$

Integration variables are the polar angle θ and azimuthal angle ϕ of particles 3 and 4, those angles of particles 5 and 6 in their rest frame, energies of particles 3 and 4, and the invariant masses of 5 and 6. Angles of 3 and 4 are defined with respect to the incoming particles and those of 5 and 6 are defined in their own center-of-mass system with respect to the momentum direction of the system.

User can introduce cutoff for angles and minimum energies as options.

For the 4-vector notation, we use (p_x, p_y, p_z, E) ordering. We use the frame where incoming particles collide along z-axis, so that the momenta of initial particles are assigned as follows:

$$\begin{array}{ll} \text{Particle-1} & (0, 0, +P, E_1) \\ \text{Particle-2} & (0, 0, -P, E_2) \end{array}$$

where P is the positive value determined by m_1, m_2, W. Here, relative velocity v_{rel} is assumed to be 2 for the collision highly relativistic particles. If colliding particles are slow, the user needs modification in `kinit.f`.

New variables are introduces as ;

$$\begin{aligned} Q_{1,2}^2 = -q_{1,2}^2 &= -(p_1 - p_{3,4})^2, \\ &= 2(EE_{3,4} \mp P_{1,2}P_{3,4}\cos\theta_{3,4}), \end{aligned}$$

and

$$q_{1,2}^0 = E - P_{3,4}.$$

Invariant mass is another variables.

$$Q_3^2 = q_3^2 = (p_5 + p_6)^2.$$

Phase space is in the center-of-mass system and it is given by

$$
\begin{aligned}
d\tilde{\Gamma}_4 &= \frac{1}{16(2\pi)^6}\frac{1}{P_3P_4}\frac{1}{\sqrt{\Omega^2-\Psi^2}}dQ_1^2dQ_2^2dq_1^0dq_2^0d\phi_3 \\
&\times \beta(q_1q_2:56)dQ_1^2d\cos\theta_{5,(56)}d\phi_{6,(56)},
\end{aligned}
$$

where

$$
\begin{aligned}
\Psi &= W^2-2W(P_3+P_4)+2(E_3E_4-P_3P4\cos\theta_3\cos\theta_4) \\
&+m_3^2+m_4^2-Q_3^2, \\
\Omega &= 2E_3E_4\sin\theta_3\sin\theta_4.
\end{aligned}
$$

For the treatment of Q_3^2, one can use one of three ways by setting the flags `IRESNS`.

- Flat distribution to Q_3^2, as in `3001` for the system of particle 4 and 5. The flag is `IRESNS=0`.
- Resonance distribution to Q_3^2, as in `3003` for the system of particle 4 and 5. The flag is `IRESNS=1`. If this mode is selected, also the user should supply the values of mass and width.
- $1/Q_3^2$ distribution, as in `3004` for the system of particle 4 and 5. The flag is `IRESNS=1`.

When the flag `ICOS3,4=1`, the kinematics is changed to the case for the radiation of particle-3,4 along beam axis.

$$dQ_{1,2}^2/Q_{1,2}^2 = d(\log Q_{1,2}^2).$$

Meaning of `X()` for BASES integration

`X(1)`	Energy of $q_1 = E_1 - E_3$
`X(2)`	Energy of $q_2 = E_2 - E_4$
`X(3)`	Momentun transfer squaer, $Q_1^2 = -(p_1 - p_3)^2$
	$Q_1^2 = Q_{1,min}^2 + (Q_{1,max}^2 - Q_{1,min}^2)X(3)$ for ICOS3=0
	$Q_1^2 = Q_{1,min}^2 (Q_{1,max}^2 / Q_{1,min}^2)^{X(3)}$ for ICOS3=1
`X(4)`	Momentun transfer squaer, $Q_2^2 = -(p_2 - p_4)^2$
	$Q_2^2 = Q_{2,min}^2 + (Q_{2,max}^2 - Q_{2,min}^2)X(4)$ for ICOS4=0
	$Q_2^2 = Q_{2,min}^2 (Q_{2,max}^2 / Q_{2,min}^2)^{X(4)}$ for ICOS4=1
`X(5)`	$\cos\theta_{5,(56)} = 2\mathtt{X(5)} - 1$
	Polar angle in the CM system of particles 5 and 6.
`X(6)`	$\phi_{5,(56)} = 2\pi\mathtt{X(6)}$
	Azimuthal angle in the CM system of particles 5 and 6.
`X(7)`	Invariant mass of particles 5 and 6, Q_3^2.
	Dependent on `IRESNS` . The same as `X(4)`
	in `3001`, `3003`, `3004` for `IRESNS(2)=0,1,-1`.
`X(8)`	$\phi_3 = 2\pi\mathtt{X(8)}$
	Azimuthal angle of particle-3

Options in `kinit.f`

Default value is shown in parenthesis.

1. Physical parameter section

   - `W` R*8 (200.0) : Center of mass energy.
   - `COSCUT(1,1)` R*8 (-1.0) : Minimum of $\cos\theta_3$. This and angles below are all in the laboratory frame.
   - `COSCUT(2,1)` R*8 (+1.0) : Maximum of $\cos\theta_3$.
   - `COSCUT(1,2)` R*8 (-1.0) : Minimum of $\cos\theta_4$.
   - `COSCUT(2,2)` R*8 (+1.0) : Maximum of $\cos\theta_4$.
   - `COSCUT(1,3)` R*8 (-1.0) : Minimum of $\cos\theta_5$.
   - `COSCUT(2,3)` R*8 (+1.0) : Maximum of $\cos\theta_5$.
   - `COSCUT(1,4)` R*8 (-1.0) : Minimum of $\cos\theta_6$.
   - `COSCUT(2,4)` R*8 (+1.0) : Maximum of $\cos\theta_6$.
   - `ENGYCT(1,1)` R*8 (m_3) : Minimum of E_3.
   - `ENGYCT(2,1)` R*8 (W) : Maximum of E_3.
   - `ENGYCT(1,2)` R*8 (m_4) : Minimum of E_4.
   - `ENGYCT(2,2)` R*8 (W) : Maximum of E_4.

- `ENGYCT(1,3)` R*8 (m_5) : Minimum of E_5.
- `ENGYCT(2,3)` R*8 (W) : Maximum of E_5.
- `ENGYCT(1,4)` R*8 (m_6) : Minimum of E_6.
- `ENGYCT(2,4)` R*8 (W) : Maximum of E_6.
- `AMASCT(1)` R*8 ($m_5 + m_5$) : Minimum of Q_3. Q_1 is the mass of the system of particles 3 and 4.
- `AMASCT(2)` R*8 ($W - m_3 - m_4$) : Maximum of Q_3.
- `ARESNS(1)` R*8 (m_W) : Mass of resonance, this and the next parameter are meaningful only when `IRESNS(1)=+1`.
- `ARESNS(2)` R*8 (Γ_W) : Width of resonance.

2. Physics control section

   - `NDIM` I*4 (8) : If you want to suppress ϕ_{q1}-integration, replace the definition of `NDIM` by the following:

     ```
     NDIM = 7
     ```

     Then ϕ_3 is fixed to 0.0.
   - `IRESNS` I*4 (0) : Treatment of Q_3^2. See the description of kinematics for the meaning.
   - `ICOS3` I*4 (1) : Treatment of θ_3.
   - `ICOS4` I*4 (0) : Treatment of θ_4.

3. `BASES` control section

   - `ITMX1` I*4 (5) : These three values control `BASES` integration. See document of `BASES` for details.
   - `ITMX2` I*4 (5) :
   - `NCALL` I*4 (5000) :
   - `NX` I*4 (50) : Value `NX` control the histograms.

Related modules

PBOOST ROXMTX MINVR2 MVMULT

Bibliography

[1] K. Aoki *et al.*, *Suppl. Prog. Theor. Phys.* **73**, 1982.

[2] T. Muta, *"Foundations of Quantum Chromodynamics"*, World Scientific, 1987.

[3] T. Kaneko, in *"New Computing Techniques in Physics Research"*, ed. D. Perret-Gallix and W. Wojcik, p.555, 1990, Edition du CNRS, Paris,
T. Kaneko and H. Tanaka, in *"Proc. of the Second Workshop on JLC"*, ed. S. Kawabata, p.250, 1991, KEK Proceeding 91-10,
T. Kaneko *et al.*, in *"New Computing Techniques in Physics Research II"*, ed. D. Perret-Gallix, p.659, 1992, Edition du CNRS, Paris.

[4] H. Tanaka, *Comput. Phys. Commun.* **58**(1990)153.

[5] H. Tanaka, T. Kaneko and Y. Shimizu, *Comput. Phys. Commun.* **64**(1991)149.

[6] S. Kawabata, *Comput. Phys. Commun.* **41**(1986)127;
S.Kawabata, BASES/SPRING v.5.1, *in preparation.*

[7] J.Fujimoto *et al.*, *Suppl. Prog. Theor. Phys.* **100**, 1991.

[8] GRACE manual, MINAMI-TATEYA group, KEK report 92-19, (1993).