ELSEVIER

Computer Physics Communications 88 (1995) 309-326

Computer Physics Communications

A new version of the multi-dimensional integration and event generation package BASES/SPRING

Setsuya Kawabata

National Laboratory for High Energy Physics, Tsukuba, Ibaraki 305, Japan

Received 26 August 1994

Abstract

The Monte Carlo integration and event generation package BASES/SPRING V1.0 S. Kawabata, Comput. Phys. Commun. 41 (1986) 127, has been upgraded so as to generate events with 50 independent variables, and to integrate functions with an alternating sign. Its ability to integrate real functions with indefinite sign is found to be useful in the numerical evaluation of interference effects among various amplitudes. Besides these changes, its program structure has been completely reformed.

NEW VERSION SUMMARY

Title of program: BASES/SPRING V5.1

Catalogue number: ADBJ

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

Reference to original program: Comput. Phys. Commun. 41 (1986) 127; *Cat. number:* AAFW

Authors of original program: S. Kawabata

Does the new version supersede the original program? no

Licensing provisions: none

Computer for which the new version is designed and others on which it is operable: HP750, FACOM, HITAC, IBM, VAX and others with a FORTRAN77 compiler.

Computer: HP750, FACOM-M780; *Installation:* National Laboratory for High Energy Physics (KEK), Tsukuba, Ibaraki, Japan

Operating system: UNIX, OSIV/F4 MSP

Programming language used: FORTRAN77

Memory required to execute with typical data: 392 Kwords

No. of bits in a word: 32

Peripherals used: disc file and printer for output

No. of lines in distributed program, including test data, etc.: 6012

Key words: Multi-dimensional integration, Monte Carlo simulation, event generation and four momentum vector generation

Nature of physical problem

The previous version of the numerical integration and event generation package BASES/SPRING has been useful to obtain total cross sections and to generate events of elementary processes in high energy physics. It is applicable to processes with up to ten independent variables. In order to study, for example, e^+e^- physics at much higher energies, we often need more than ten independent variables to describe pro-

SSDI 0010-4655(95)00028-3

cesses of our interest. As far as the numerical integration by BASES is concerned, it is easy to extend the dimension of integral. However, the event generation requires a huge memory space with the previous generation algorithm of SPRING.

Method of solution

BASES/SPRING is suited for the integration and generation of a very singular function. The number of those independent variables which give the function a singular behavior is usually small. Then, if we divide the subspace spanned by these singular variables into hypercubes, the number of hypercubes becomes not too large. Applying the previous BASES/SPRING algorithm only to this subspace and handling the other variables as additional integral variables, we could extend the dimension of integral and event generation up to 50 variables.

Typical running time

The running time is essentially determined by the complexity of the function program which gives the differential cross section of an elementary process. If we take the process $e^+e^- \to \nu\bar{\nu}\gamma$ as an example, the two dimensional integration and the generation of 10000 events require 4.4 seconds in total on a FACOM M780 computer.

LONG WRITE-UP

1. Introduction

The numerical integration and event generation package BASES/SPRING V1.0 has been applied to many elementary processes [2]. In recent years very high energy e^+e^- colliders, JLC (KEK), NLC (SLAC), and CLIC (CERN), are being studied as future plans. At the energies of these machines, the number of final state particles in some elementary processes is so high that we cannot apply the version V1.0 directly because of its limited number of dimensions (ten at maximum). Extension of the number of dimensions is easy for the integration package BASES, but it is not for the event generation package SPRING. Before discussing this point, it is useful to describe the algorithm of the version V1.0 briefly.

BASES integrates a function by means of the importance and stratified sampling method.

In order to realize the importance sampling each variable axis is divided into N_{region} *regions*, each of which is subdivided into m *subregions*. In total each variable axis has $N_g (= N_{\text{region}} \times m)$ subregions. The numbers N_{region} and m are determined from the dimension of integral N_{dim} automatically by BASES. The full integral volume is covered by a *grid* of $N_g^{N_{\text{dim}}}$ subregions. We call a subspace of the integral volume a *hypercube* which is spanned by the regions of each variable axis.

In each hypercube, sample points of a definite number N_{sample} are taken, and the integral and its variance are evaluated. The estimate of integral and its variance are calculated by summing up results of all hypercubes. We call this process an *iteration*. A cumulative estimate and its error are calculated from the results of all iterations statistically.

Execution of BASES consists of the grid optimization and the integration steps.

At the first iteration of the former step the grid is uniformly defined for each variable axis. After each iteration the grid is adjusted so as to make the sizes of subregions narrower at the parts with the larger function value and wider at the parts with the smaller one. In this way a suited grid to the integrand is obtained.

In the integration step, the probability to select each hypercube and the maximum value of the function in it are calculated as well as the estimate of integral with the frozen grid determined in the former step.

In SPRING, a hypercube is sampled according to its probability, calculated by BASES, and a point in the hypercube is selected by sample-and-reject method for which the maximum value of the function is used. If the grid is sufficiently optimized, a high efficiency of generating a set of values of the independent variables (i.e. event generation) can be achieved.

The difficulty in the straightforward extension of the number of dimensions in BASES/SPRING V1.0 can be easily seen by considering a 25 dimensional case (i.e. 25 independent variables). Even though we suppose to have only two regions for each variable axis, there are $N_{\text{cube}} = 2^{25}$ hypercubes in total. The calculation of such a huge number N_{cube} of probabilities requires much CPU time and their storage space is also too large. It seems to be unrealistic.

By assuming, however, an additional condition

to the integrand, this difficulty can be avoided. The algorithm of the new version BASES/SPRING V5.1 is described in the next section, and its program structure and usage are shown in Section 3. In Section 4 we will discuss its test run.

2. Algorithm of version 5.1

In usual practice the function of our interest has only a limited number of variables that cause its singular behavior. We call such kind of variables *wild variables*. The rest variables, on which the function depends weakly, are called *gentle variables*. If we divide only the subspace spanned by the wild variables into hypercubes, the number of hypercubes is not too large. Thus, by applying the BASES/SPRING V1.0 algorithm only to this subspace and by handling the gentle variables as additional integral variables, we can overcome the difficulty in the event generation of higher dimensions.

In order to see the difference between the old and the new algorithms, we consider the following integral of three dimensions:

$$I = \int_0^1 f(x, y, z)\, \mathrm{d}x\, \mathrm{d}y\, \mathrm{d}z.$$

In the old algorithm, the estimate of integral is given by

$$I \simeq \sum_{j=1}^{N_{\mathrm{cube}}} \frac{1}{N_{\mathrm{sample}}} \sum_{i=1}^{N_{\mathrm{sample}}} \frac{f(x_i^j, y_i^j, z_i^j)}{p(x_i^j, y_i^j, z_i^j)},$$

where x_i^j, y_i^j and z_i^j indicate ith sample point in jth hypercube, $p(x_i^j, y_i^j, z_i^j)$ is the probability density at the point for the importance sampling, and N_{sample} is the number of sample points in each hypercube.

On the other hand, if we take the new algorithm and suppose the variables x and y are wild and z is gentle, the estimate is given by

$$I \simeq \sum_{j=1}^{N_{\mathrm{cube}}} \frac{1}{N_{\mathrm{sample}}} \sum_{i=1}^{N_{\mathrm{sample}}} \frac{f(x_i^j, y_i^j, z_i)}{p(x_i^j, y_i^j, z_i)},$$

where z is sampled in the full range of z variable. Since the importance sampling takes place even for the gentle variable z, the numerical integration converges fast enough. An efficient event generation is guaranteed if the function

$$F^j(z) = \int_{j-th\,\mathrm{hypercubes}} f(x^j, y^j, z)\, \mathrm{d}x^j\, \mathrm{d}y^j$$

has similar dependences on z for all j.

The numbers of hypercubes are N_{region}^3 and N_{region}^2 in the old and the new algorithms, respectively, where N_{region} is the number of regions per variable. Generally speaking, if N_{dim} and N_{wild} are the numbers of independent variables and the wild variables ($N_{\mathrm{wild}} \leq N_{\mathrm{dim}}$), respectively, the number of hypercubes N_{cube} is given by

$$N_{\mathrm{cube}} = N_{\mathrm{region}}{}^{N_{\mathrm{dim}}} \quad \text{(old)}$$
$$= N_{\mathrm{region}}{}^{N_{\mathrm{wild}}} \quad \text{(new)}.$$

In version 5.1, the maximum number of wild variables is equal to 15.

3. Program structure and usage

In the old version, users must prepare many subprograms, e.g. `USERIN`, `FUNC`, `USROUT`, `SPINIT`, `SPEVNT`, `SPTERM` etc. according to the specification of each subprogram. This complexity seems to give a pedestrian a high threshold to use this package. Furthermore, the program flow is completely controlled by the BASES/SPRING package, and the numerical integration and the event generation are designed to be successively performed by two separate procedures. This makes it less flexible particularly in the event generation, to pass generated events to detector simulation and data analysis. In order to dissolve these inconveniencies, the program structure has been completely reformed.

3.1. Program components

When we study an elementary process, we first calculate the effective cross section by integrating the differential cross section with some kinematical cuts by `BASES`, generate events of four momentum vectors of final state particles by `SPRING`, pass through a detector simulation program to simulate the experimental data, and then

```
      C      Main Program for BASES/SPRING V5.1
      C
            IMPLICIT REAL*8(A-H,O-Z)
            EXTERNAL FUNC

            PARAMETER (MXDIM = 50 )
            COMMON /BPARM1/ XL(MXDIM),XU(MXDIM),NDIM,NWILD,IG(MXDIM),NCALL
            COMMON /BPARM2/ ACC1,ACC2,ITMX1,ITMX2
      C
            COMMON/KINEM/W,EM,ZM,ZGAM,CZ,CV,CA,FACTOR,XK,COSTH
            REAL*4 P(4)
            DATA PI,ALP,GEVNB,GENER /
                 3.1415926D0, 137.036D0, 0.38927D6, 3.0D0 /

      ********************************************************************
      *     Initialization of BASES/SPRING V5.1
      ********************************************************************
      *===> Initialization of BASES by calling BSINIT

                 CALL BSINIT

      *====>  Initialization of parameters for kinematics etc.

                 EM     = 0.511E-3
                 ZM     = 90.0
                 WM     = 78.97
                 ZGAM   = 2.6
                 ALPHA  = 1./ALP
                 RAD    = PI/180.
                 TWOPI  = 2.D0*PI
                 SQ     = SQRT(ZM**2-WM**2)
                 CZ     = ZM**2/(2.*WM*SQ)
                 CA     = CZ*(1./2.)
                 CV     = 2.*CA*(-1./2.+2.*SQ**2/ZM**2)
                 FACTOR = GENER*CZ**2*ALPHA**3/12.*GEVNB
                 COSMIN = 15.
                 COSMAX =180.-COSMIN
                 W      = 105.0D0
                 XKMIN  = 1./W

      *====>  Initialization of BASES parameters

                 NDIM  = 2
                 NWILD = 1
                 NCALL = 5000
                 ITMX1 = 10
                 ACC1  = 0.1D0
                 ITMX2 = 100
                 ACC2  = 0.05D0

                 XL(1)= XKMIN
                 XU(1)= W/2.
                 XL(2)= DCOS(COSMAX*RAD)
                 XU(2)= DCOS(COSMIN*RAD)

      *====>  Initialization of Histograms

                 CALL XHINIT(1,XL(1),XU(1),40,'Photon Energy (GeV)')
                 CALL XHINIT(2,XL(2),XU(2),50,'Cos(theta) of Photon')
                 CALL DHINIT(1,XL(2),XU(2),50,XL(1),XU(1),50,
                             ' x : cos(theta)  --  y : Photon Energy ')

      ********************************************************************
      *     Nimerical Integration by BASES V5.1
      ********************************************************************

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

                 LU     = 6

                 CALL BSINFO( LU )

                 CALL BHPLOT( LU )

      ********************************************************************
      *     Event generation by SPRING V5.1
      ********************************************************************
      *====> Initialization of additional histograms

                 CALL XHINIT( 5,  0.0D0, 4.0D1, 40,
                              'Photon Transverse Energy (GeV)')

      *====> Event generation loop

                 MXTRY  = 50
                 MXEVNT = 10000

                 DO 100 NEVNT = 1, MXEVNT

                    CALL SPRING( FUNC, MXTRY )

      *            Compute the four vectors of generated event
      *            from the kinematical variables

                    PHI    = TWOPI*DRN(DUMY)
                    PXY    = XK*SQRT(1.0 - COSTH*COSTH)
                    P(1)   = PXY*COS(PHI)
                    P(2)   = PXY*SIN(PHI)
                    P(3)   = XK*COSTH
                    P(4)   = XK

      C             WRITE(20) P

                    CALL XHFILL( 5, PXY, 1.D0)

        100      CONTINUE

                 CALL SPINFO( LU )

                 CALL SHPLOT( LU )

             STOP
             END
```

Fig. 1. Main program of the test run.

analyze the resultant simulation data to compare with real data.

If the process has a few particles in the final states, the numerical integration and the event generation do not take much computing time. It is better to carry out the simulation study in one procedure for simplicity. If the process has many final state particles, however, the numerical integration takes much computing time. It is then recommended to separate the integration part from the other parts consisting of the event generation and the detector simulation, etc. Since in the present version the numerical integration and the event generation are carried out by calling subroutines BASES and SPRING, respectively, the simulation study can be performed either in one procedure or in separate two or more procedures. In any case users are requested to prepare main and function programs. In addition to the subroutines BASES and SPRING, a set of subroutines, described below, are prepared for use. It should be noticed that all the real variables used in the system are defined to be of double precision.

BSINIT: To set the BASES/SPRING parameters equal to defaults.
BSDIMS: To set the numbers of dimensions and the wild variables, and the lower and the upper limits of the integral variables.
BSGRID: To set the sampling flags for the integral variables.
BSPARM: To set the parameters for integration.
BASES: To make numerical integration and to prepare the probability information for event generation.
BSINFO: To print a parameter list and convergency behavior of the integration.
BHPLOT: To print histograms and scatter plots, taken in the integration.
SPRING: To generate an event with weight one.
SPINFO: To print statistical information about the event generation.
SHPLOT: To print histograms and scatter plots, taken in the event generation.
BSWRIT: To store the probability information into a file.
BSREAD: To restore the probability information from a file.

An example of outputs from BSINFO, BHPLOT, SPINFO, and SHPLOT is shown and explained for the test run output in Section 4.

The program flow is fully controlled by the main program written by users. Its specification is described in the next subsection together with the usage of these subroutines.

3.2. Main program

The main program can be divided into the initialization part of BASES/SPRING, the numerical integration part, and the event generation part. A list of the main program for the test run is given in Fig. 1 as an example.

(1) Initialization of BASES / SPRING

At the beginning, BSINIT is called to set the parameters for BASES/SPRING equal to default values. Then the parameters for the integration and event generation are to be initialized to their proper values, which are passed to the system through the labeled commons BPARM1 and BPARM2. All real parameters in these two labeled commons are to be given with double precision. The content of common BPARM1 is as follows;

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

NDIM: The number of independent variables (dimension of integral) up to 50.
NWILD: The number of the wild variables up to 15.
XL(i), XL(i): The lower and the upper limits of ith integral variable. Notice that the first NWILD variables must correspond to the wild variables in each array.
IG(i): The flag to switch the sampling method for ith variable. The default setting is the importance sampling (0/others) = (uniform/importance sampling).
NCALL: The number of sample points per iteration.

The true number of sample points $N_{\text{call}}^{\text{real}}$ differs from a given number NCALL, which is automatically determined by the following algorithm.

The number of regions per variable N_{region} is determined as the maximum number which satisfies the two inequalities:

$$N_{\text{region}} = \left(\frac{N_{\text{call}}}{2}\right)^{\frac{1}{N_{\text{wild}}}} \leq 25,$$

and

$$N_{\text{region}}^{N_{\text{wild}}} < N_{\text{cube}}^{\max},$$

where $N_{\text{cube}}^{\max}$ is the maximum number of hypercubes and is set equal to 32768. The number of hypercubes is given by $N_{\text{cube}} = N_{\text{region}}{}^{N_{\text{wild}}}$, then the number of sample points per cube is $N_{\text{sample}} = N_{\text{call}}/N_{\text{cube}}$. Since the number N_{sample} is an integer, the calculated number $N_{\text{call}}^{(\text{real})} = N_{\text{sample}} \times N_{\text{cube}}$ may differ from the given number $N_{\text{call}}^{(\text{given})}$ (= NCALL).

The content of common BPARM2 is as follows;

```
COMMON/BPARM2/ACC1, ACC2, ITMX1,
  ITMX2
```

ACC1: The desired accuracy of integration for the grid optimization step in units of percent.
ACC2: The desired accuracy of integration for the integration step in units of percent.
ITMX1: The number of iterations for the grid optimization step.
ITMX2: The number of iterations for the integration step.

For those users, who do not like to set these parameters directly on the labeled commons, a set of subprograms is prepared.

To set the dimension, the number of wild variables, the lower and upper limits of integral variables,

```
CALL BSDIMS (NDIM, NWILD, XL, XU),
```

where XL and XU are arrays with the double precision real numbers.

To set the flag IG,

```
CALL BSGRID(NDIM, IG),
```

where IG is an integer array. Unless this routine is called, all integral variables are sampled by the importance sampling method as the default.

To set the number of sample points per iteration, the required accuracies and the maximum numbers of iterations both for the grid optimization and the integration steps,

```
CALL BSPARM (NCALL, ACC1, ACC2,
  ITMX1, ITMX2),
```

where ACC1 and ACC2 are to be given by the double precision real numbers.

In addition to the initialization of the BASES/SPRING parameters, parameters and constants, like center of mass energy, beam energy, kinematical cut values, the fine structure constant and the numerical value of π, etc., should be set and be passed to the function program FUNC through some labeled commons to calculate the differential cross section.

When we make histograms and scatter plots of some variables, their initialization should be done here. The system has a proper histogram package, whose description is given in Subsection 3.4.

(2) Numerical integration

In the grid optimization step, sizes of the subregions for each variable are optimized to the behavior of integrand iteration by iteration, starting from a uniform size of subregions, as briefly described in Section 1. The algorithm for the grid optimization is identical to that of the old version and VEGAS, whose more detailed description is found in Refs. [1] and [3].

In the integration step, by fixing the grid determined in the grid optimization step a cumulative estimate of the integral is calculated as well as the probability information for each hypercube spanned by the wild variables.

The both steps are terminated when the estimated accuracy of the integral reaches a required value (ACC1 / ACC2) or the number of iterations exceeds a given number (ITMX1 / ITMX2).

The estimate of the integral and its error etc.

of the integration step are returned to users in the arguments of the subroutine BASES as follows:

```
CALL BASES (FUNC, ESTIM, ERROR,
CTIME, IT1, IT2).
```

FUNC: The name of a function program, whose specification is described in Subsection 3.3.
ESTIM: A cumulative estimate of the integral.
ERROR: The standard deviation of the estimate of the integral.
CTIME: The computing time used by the integration step in seconds.
IT1: The number of iterations made in the grid optimization step.
IT2: The number of iterations made in the integration step.

The tables of the convergency behavior of the integral both for the grid optimization and the integration steps are printed out on the standard output device. We can also keep these tables in a file by calling the subroutine BSINFO, for example,

```
CALL BSINFO(LU).
```

The file allocated to the logical unit LU should be opened in the initialization step (1). The contents of the tables are described in Subsection 4.2.

When histograms and scatter plots are made, they can be printed on a logical unit LU by

```
CALL BHPLOT(LU).
```

Subroutines BSINFO and BHPLOT do not need to be called, unless outputs from them are required.

If the numerical integration and the event generation are separately performed by different procedures, the probability information made by BASES should be stored on a file by calling BSWRIT, for instance, as follows;

```
CALL BSWRIT(LN).
```

The file allocated to the logical unit LN must be opened at the initialization step (1). Furthermore, at the beginning of the event generation program and just after the initialization of BASES/SPRING described above, subroutine BSREAD should be called to restore the probability information from the file as follows;

```
CALL BSREAD(LN),
```

then the event generation can be carried out.

(3) Event generation

After the integration events can be immediately generated by calling SPRING. For each call of subroutine SPRING a set of values of independent variables is generated with weight one. The calling sequence of SPRING is as follows;

```
CALL SPRING(FUNC,""MXTRY).
```

FUNC: The name of a function program.
MXTRY: The maximum number of trials to generate one event.

At the first call of SPRING a cumulative probability distribution over all the hypercubes is calculated from the differential one, prepared by BASES. This cumulative distribution is used to sample a hypercube according to its probability.

After a hypercube is sampled in this way, a point is selected in this hypercube by the importance sampling method and is examined whether it is accepted or not. When this point is not accepted, another point is sampled in the same hypercube and tested again. If the grid is not optimized enough, there is a possibility to get into an infinite loop of generation. To avoid this occurrence the number of trials to get an event is limited to MXTRY whose recommended number is 50. The case where an event could not be generated even by MXTRY trials is called *mis-generation*. If the number of mis-generations exceeds MXTRY, a warning message is printed.

The event generation by SPRING determines only a set of values of independent variables of the function program. Users should calculate the four-momentum vectors from this set of variables.

```
:  *     Function program for the elementary process
:  *          e+ e- ==> nu(mu) nu-bar(mu) gamma
:  *       Two independent variables
:  *           XK    = X(1) : the energy of photon
:  *           COSTH = X(2) : cos(theta) of photon
:
1         REAL  FUNCTION FUNC*8(X)
:
2         IMPLICIT REAL*8(A-H,O-Z)
3         PARAMETER (MXDIM = 50 )
4         DIMENSION X(MXDIM)
:
5         COMMON/KINEM/W,EM,ZM,ZGAM,CZ,CV,CA,FACTOR,XK,COSTH
:
6             REZ(S)= S-ZM**2
7             Z2(S) = (REZ(S))**2+(ZM*ZGAM)**2
8             FUNC  = 0.
9             XK    = X(1)
10            S     = W*W
11            S1    = W*(W-2.*XK)
12            E     = W/2.
13            PP    = SQRT(E**2-EM**2)
14            COSTH = X(2)
15            D1    = XK*(E+PP*COSTH)
16            D2    = XK*(E-PP*COSTH)
:
:  C ......... MATRIX ELEMENT SQUARE STARTS
:  C
:  C                    Z-DECAY INTO MU-NUTRINO PAIR
:  C
:
17         ANS1=(S**2*CA**2+S**2*CV**2-2.*S*W*XK*CA**2-2.*S*W*XK*CV**2-3.*S*
:        . CA**2*EM**2-3.*S*CV**2*EM**2+2.*W*XK*CA**2*EM**2+2.*W*XK*CV**2*
:        . EM**2)/D1+S1*(-S**2*CA**2-S**2*CV**2+10.*S*CA**2*EM**2-2.*S*CV**
:        . 2*EM**2-16.*CA**2*EM**4+8.*CV**2*EM**4)/(2.*D1*D2)+(-S**3*CA**2-
:        . S**3*CV**2+2.*S**2*W*XK*CA**2+2.*S**2*W*XK*CV**2+2.*S**2*CA**2*
:        . EM**2+2.*S**2*CV**2*EM**2-4.*S*W*XK*CA**2*EM**2-4.*S*W*XK*CV**2*
:        . EM**2-4.*W**2*XK**2*CA**2*EM**2+4.*W**2*XK**2*CV**2*EM**2)/(2.*
:        . D1*D2)-(D2*S1*EM**2)*(CA**2+CV**2)/D1**2+D2*EM**2*(-S*CA**2-S*CV
:        . **2+2.*W*XK*CA**2+2.*W*XK*CV**2)/D1**2+S1*EM**2*(S*CA**2+S*CV**2
:        . -8.*CA**2*EM**2+4.*CV**2*EM**2)/(2.*D1**2)+S*EM**2*(S*CA**2+S*CV
:        . **2-2.*W*XK*CA**2-2.*W*XK*CV**2)/(2.*D1**2)
18         TAU=-(D1*S1)*(CA**2+CV**2)/D2+D1*(-S*CA**2-S*CV**2+2.*W*XK*CA**2+
:        . 2.*W*XK*CV**2+2.*CA**2*EM**2+2.*CV**2*EM**2)/D2-(D1*S1*EM**2)*(CA
:        . **2+CV**2)/D2**2+D1*EM**2*(-S*CA**2-S*CV**2+2.*W*XK*CA**2+2.*W*
:        . XK*CV**2)/D2**2+4.*EM**2*(CA**2+CV**2)+S1*(S*CA**2+S*CV**2-3.*CA
:        . **2*EM**2-3.*CV**2*EM**2)/D2+(S**2*CA**2+S**2*CV**2-2.*S*W*XK*CA
:        . **2-2.*S*W*XK*CV**2-3.*S*CA**2*EM**2-3.*S*CV**2*EM**2+2.*W*XK*CA
:        . **2*EM**2+2.*W*XK*CV**2*EM**2)/D2+S1*EM**2*(S*CA**2+S*CV**2-8.*
:        . CA**2*EM**2+4.*CV**2*EM**2)/(2.*D2**2)+S*EM**2*(S*CA**2+S*CV**2-
:        . 2.*W*XK*CA**2-2.*W*XK*CV**2)/(2.*D2**2)-(D2*S1)*(CA**2+CV**2)/D1+
:        . D2*(-S*CA**2-S*CV**2+2.*W*XK*CA**2+2.*W*XK*CV**2+2.*CA**2*EM**2+
:        . 2.*CV**2*EM**2)/D1+S1*(S*CA**2+S*CV**2-3.*CA**2*EM**2-3.*CV**2*EM
:        . **2)/D1+ANS1
:
:  *===> Calculate the value of function
:
19            DSDX = -FACTOR*TAU*XK/(E*Z2(S1))
20            FUNC = DSDX/E
:
:  *===> Fill histograms and scatter plot
:
21            CALL XHFILL(1,XK,DSDX)
22            CALL XHFILL(2,COSTH,DSDX)
23            CALL DHFILL(1,COSTH,XK,DSDX)
:
:
24            RETURN
25            END

      Input of this package :     71 lines
```

Fig. 2. Function program of the test run.

A typical structure of the event generation step is as follows:

```
MXTRY=50
DO 100 NEV=1, 10000
  CALL SPRING( FUNC, MXTRY)
     ....
     Calculate four momentum vectors
     ....
100 CONTIINUE
```

Since the kinematics of the process is calculated in the function program, it is better to use the result of kinematics for calculating the four-momentum vectors.

After the event generation, statistical information about the generation can be obtained by calling SPINFO as follows:

```
CALL SPINFO(LU),
```

where LU is the logical unit of an output device. Histograms and scatter plots can be also printed on the logical unit LU by

```
CALL SHPLOT(LU).
```

3.3. Function program

In the function program the value of the integrand is calculated at the sample point fed by BASES. A set of numerical values of integral variables is passed through the argument of the function program. A typical structure of function program is as follows;

```
DOUBLE PRECISION FUNCTION FUNC(X)
REAL*8 X(2)
FUNC=0.0D0
... Calculation of kinematics ...
IF(the point is outside of
   the kinematical boundary) RETURN
FUNC=is calculated from X(i) for i=1, 2
CALL XHFILL(ID, V, FUNC)
CALL DHFILL(ID, Vx, Vy, FUNC)
RETURN
END
```

A recipe for writing a function program is:

(1) Calculate the kinematical variables by which the differential cross section is described from the integral variables, X(i) for $i = 1$ to NDIM.
(2) If the point fed by BASES is found to be outside the kinematical boundary, set the value of the function equal to zero and return.
(3) If the point is inside the kinematical boundary, calculate the numerical value of the function at the point and return it as the function value.
(4) If a histogram is required, call subroutine XHFILL once.
(5) If a scatter plot is required, call DHFILL once.

An example of FUNC for the process $e^+e^- \to \nu_\mu \bar{\nu}_\mu \gamma$ is given in Fig. 2.

3.4. Histogram package

The system has a proper histogram package, by which histograms and scatter plots of any variables can be printed. A characteristic of this package is that there are two kinds of histograms, the one is called "original histograms" and another is "additional histograms".

The original histogram is quite useful to check whether the frequency distribution of generated events really reproduces the distribution calculated in the integration stage. On the other hand the additional histogram prints only the frequency distribution of the generated events. A detailed description of these two kinds of histograms is found in Subsection 4.2.

The maximum numbers of histograms and scatter plots are 50 for each. If the initialization routine XHINIT (or DHINIT) is called more than 50 times, subsequent calls are neglected. Usage of this histogram package is as follows;

(1) To initialize histograms and scatter plots the following routines are to be called:

```
   CALL XHINIT(ID#,
  · lower_limit ,
  · upper_limit , # of bins,
  ·   'Title of histogram ')
```

and

```
CALL DHINIT(ID#,
· x-lower_limit ,
· x-upper_limit , # of x bins,
· y-lower_limit ,
· y-upper_limit , # of y bins,
·   'Title of scatter plot '),
```

respectively.
(2) To fill the histograms and scatter plots the following routines are to be called in FUNC:

```
CALL XHFILL(ID#, V, FUNC)
```

and

```
CALL DHFILL(ID#,""Vx,""Vy,""FUNC)
```

respectively.
(3) To print the resultant histograms and scatter plots taken in the integration,

```
CALL BHPLOT(LU)
```

where LU is a logical unit number.
(4) To print the resultant histograms and scatter plots taken in the event generation,

```
CALL SHPLOT(LU)
```

where LU is a logical unit number.

Examples of histograms and scatter plots printed by BHPLOT and SHPLOT are shown in the test run output.

4. Test run

4.1. Test of algorithm

For the test of the new version we tate the process $e^+e^- \to \nu\bar{\nu}\gamma$ as an example, which was also used in Ref. [1]. The integral variables are energy and polar angle of the photon. As shown in histograms of the test run output, the photon energy distribution has sharp peaks at zero and the Z^0 mass, while the angular distribution has forward and backward peaks. In order to demonstrate that the new algorithm works well, we tested our program in the following three cases:

(1) Set the integration parameters, NDIM = 2, NWILD = 2, and NCALL = 5000. Since $N_{\rm dim} = N_{\rm wild}$, the integration algorithm is exactly identical to the old one.
(2) Set the parameters, NDIM = 2, NWILD = 1, IG(2) = 1, and NCALL = 5000. Only the photon energy is considered as the wild variable and the polar angle is sampled by the importance sampling.
(3) Set the parameters, NDIM = 2, NWILD = 1, IG(2) = 0, and NCALL = 5000. The polar angle is uniformly sampled.

As summarized in Table 1, cases (1) and (2) give a consistent result and almost identical speed of convergency. The result of integration is not affected by the difference of algorithms. The efficiency of event generation by the new algorithm is slightly lower than that of the old one, but still

Table 1
The results for three cases of integration parameters

Case	Numerical integration				Event generation	
	Estimate (error)	CPU time	It-1	It-2	Efficency	CPU time
(1)	$4.528033\,(\pm 0.002230)\times 10^{-2}$	3.43	10	26	75.8	0.40
(2)	$4.527550\,(\pm 0.002255)\times 10^{-2}$	3.59	10	27	69.4	0.41
(3)	$4.521658\,(\pm 0.006430)\times 10^{-2}$	10.67	10	100	11.6	1.88

Case (1) is by the old algorithm. Cases (2) and (3) are by the new version with the importance and uniform samplings for the gentle variable, respectively. It-1 and It-2 are the numbers of iterations for the grid optimization and the integration steps, respectively. The generation efficiency is defined to be the ratio of the number of accepted events to that of trials. The CPU time for generation is the computing time consumed to generate 10k events and calculate their four-momentum vectors.

high. This result shows that the new algorithm works very well.

When we sample the polar angle variable uniformly (case (3)), the convergency of the integration is much worse than the other two cases and the efficiency of event generation is very low. It is very reasonable, since the sampling of the photon polar angle is never optimized.

It is, however, noted that the uniform sampling may give much faster convergence than the importance one for those variables whose distributions are known to be nearly flat. The reason is as follows. The importance sampling is done by using the distribution taken at the previous iteration. If this distribution has some bias due to less statistics of sampling, the resultant importance sampling is also biased. Although this bias is usually canceled during many iterations, it requires much computing time to reach a stable answer.

4.2. Output from BASES / SPRING

As the test run output, we take case (2). There are four sets of outputs, the outputs from the numerical integration part by calling `BSINFO` and `BHPLOT`, and from the event generation part by calling `SPINFO` and `SHPLOT`.

(1) Output from `BSINFO`

On the first page of the output, detailed information about the integration parameters and computing time is printed.

As the integration parameters,

N_{dim}: the number of integral variables (dimension),
N_{wild}: the number of wild variables,
$N_{\mathrm{call}}^{\mathrm{given}}$: the given number of sample points per iteration,
$N_{\mathrm{call}}^{\mathrm{real}}$: the true number of sample points,
N_{g}: the number of subregions per variable,
N_{region}: the number of regions per variable,
N_{cube}: the number of hypercubes,
`XL` (i): the lower limit of ith integral variable,
`XU`(i): the upper limit of ith integral variable,
`IG`(i): the sampling flag for ith integral variable,
`ITMX1`: the maximum number of iterations for the grid optimization step,
`ACC1`: the desired accuracy in the grid optimization step,
`ITMX2`: the maximum number of iterations for the integration step, and
`ACC2`: the desired accuracy in the integration step

are listed. As the computing time information, the consumed CPU time (in units of hour, minute, and second) each for the grid optimization step, the integration step, and the overhead by the remainders is printed. Furthermore the expected computing time for generating 1000 events is estimated by assuming 70% of generation efficiency.

On the second page of the output, the convergency behavior for the grid optimization step is printed, which consists of the result of each iteration and the cumulative result up to the iteration. In the result of each iteration, the iteration number, the hit rate of the kinematically allowed region, the fraction of sample points resulting in negative function values, the estimate of the integral, and its accuracy are included. The cumulative result consists of the cumulative estimate of the integral, its error, and its accuracy. Since at the first iteration sampling points are uniformly distributed, the estimate of integral results in a smaller value than the final one, while its accuracy is worse than the final one. They converge to some final values due to the grid optimization.

On the third page of the output, the convergency behavior for the integration step is printed, whose content is identical to that for the grid optimization step. It is quite important to check if the estimate of the integral and its accuracy are stable in each iteration. If they fluctuate more than 5σ, a warning message is printed on a standard output device. This may happen either when the number of sampling points per iteration is too small or the choice of integral variables is inadequate for the behavior of the differential cross section.

(2) Output from `BHPLOT`

`BHPLOT` prints histograms and scatter plots made during the numerical integration.

The first and the last bins of a histogram are underflow and overflow bins. The first column shows the lower value of each histogram bin. The second column represents the estimated differential value and its error after the characters "+-", both of which are to be multiplied by a factor 10^{xx} shown by "E xx". On the right hand side of these columns a histogram of the differential values is drawn both in the linear scale with the character "*" and in the logarithmic scale with the character "0". If negative values exist in some bins only the linear scale histogram is displayed.

A scatter plot represents only the relative height of a function. The height of the function value is described by ten characters: 1, 2, 3,..., 8, 9, and *, while the depth (for negative function values) is displayed by the other ten characters: a, b, c,..., h, i and #. The point which results in a negative value but larger than the threshold of level "a" is indicated by the character "-", while the point describing a positive value less than the threshold of level "1" is given either by the character "+" (if a negative value exists somewhere) or by the character "." (if only positive values exist).

(3) Output from SPINFO

SPINFO prints statistics of event generation, which consists of the generation efficiency, the number of generated events, the computing time each for the event generation by SPRING, for its overhead, and for the other part consumed by the calculation of four-momentum vectors and detector simulation, etc. The maximum number of trials MXTRY and the number of mis-generations are also printed.

(4) Output from SHPLOT

SHPLOT prints histograms and scatter plots made during event generation. There are two kinds of histograms as mentioned in Subsection 3.4.

The original histogram is to be defined before the numerical integration by calling BASES. The contents of the histogram filled during the integration is compared with the frequency distribution made in the event generation. This comparison is made in the logarithmic scale, where the statistical error of each bin is represented by the characters "⟨⟩". If the error is smaller than the two character space, only the frequency is shown by the character "0". The histogram filled during the integration is represented by the character "*".

The additional histogram is to be initialized just before starting the event generation. Since there are no corresponding data made in the integration step, only the frequency distribution of events is displayed both in the linear and the logarithmic scales.

On the first page of the output, a histogram displaying the number of trials to obtain an event is always printed in the format of an additional histogram. If this histogram has a sharp peak at the first bin, the efficiency of generation is high.

Then the original histograms and the additional histograms are successively printed if they are defined.

The scatter plots printed by SHPLOT describe only the frequency distribution in terms of the relative height with ten levels.

5. Conclusion

As the test elementary process we took a simple example: $e^+e^- \to (Z^0) \to \nu\bar{\nu}\gamma$. The physical conditions for the integration are taken to be identical to those of Ref. [4] (the center of mass energy $W = 105$ GeV and the polar angle cut for the photon $165° > \theta_\gamma > 15°$), except for the energy threshold of the photon $E_\gamma > 1/W$ GeV. Besides this test the new algorithm was applied to the process $e^+e^- \to t\bar{t}Z^0$, where $t \to bf\bar{f}$ and $Z^0 \to f\bar{f}$ [5]. In this calculation, two variables were used to distinguish the different helicity combinations and another one to select decay modes of Z^0 and $W^\pm$, respectively, in addition to 20 variables for the phase space. The new version worked well even for this 23-dimensional integration and event generation.

By using the new version, we can integrate even a singular function and reproduce the behavior of the function by generating events with

weight one. There are, however, the following restrictions:

(1) The maximum number of wild variables is limited to 15.
(2) There should be no strong correlation among the wild variables. If it exists, singular points may run continuously along a diagonal line of the integral volume, and our algorithm cannot be applied. We should carefully choose the integral variables to avoid such correlations.

In spite of these limitations, this new version of BASES/SPRING will be useful for many applications in the field of very high energy physics.

Acknowledgements

The author greatly appreciates Prof. Y. Shimizu, Prof. T. Ishikawa, Dr. K. Fujii, Dr. K. Hagiwara, Dr. J. Fujimoto at KEK, Prof. K. Kato at Kogakuin University, Dr. D. Perret-Gallix at LAPP in France, and other users for their valuable comments. He also thanks Prof. S. Iwata at KEK and Prof. Y. Oyanagi at the University of Tokyo for their encouragement throughout this work.

References

[1] S. Kawabata, Comput. Phys. Commun. 41 (1986) 127.
[2] For example: K. Tobimatsu and Y. Shimizu, Prog. Theor. Phys. 74 (1985) 567; 75 (1986) 905.
S. Kuroda et al., Comput. Phys. Commun. 48 (1988) 335.
J. Fujimoto and M. Igarashi, Prog. Theor. Phys. 74 (1985) 791.
[3] G.P. Lepage, J. Comput. Phys. 27 (1978) 192.
VEGAS: Adaptive Multi-dimensional Integration Program, CLNS-80/447 (March 1980).
[4] G. Barbiellini, B. Richter and J.L. Siegrist, Phys. Lett. B 106 (1981) 414.
[5] T. Tauchi, K. Fujii, A. Miyamoto; KEK-Preprint 90–148 (November 1990).

TEST RUN OUTPUT

```
                                                    Date: 94/ 5/10  13:16
 ************************************************************
 *                                                          *
 *     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    =        2    ( 50 at max.)
     # of Wilds      :    Nwild   =        1    ( 15 at max.)
     # of sample points : Ncall   =     5000(real)     5000(given)
     # of subregions    : Ng      =       50 / variable
     # of regions       : Nregion =       25 / variable
     # of Hypercubes    : Ncube   =       25

 (2) About the integration variables
     ------+--------------+--------------+-------+-------
         i       XL(i)          XU(i)       IG(i)   Wild
     ------+--------------+--------------+-------+-------
         1    9.523810E-03    5.250000E+01      1      yes
         2   -9.659258E-01    9.659258E-01      1       no
     ------+--------------+--------------+-------+-------

 (3) Parameters for the grid optimization step
     Max.# of iterations: ITMX1 =       10
     Expected accuracy  : Acc1  =   0.1000 %

 (4) Parameters for the integration step
     Max.# of iterations: ITMX2 =      100
     Expected accuracy  : Acc2  =   0.0500 %


<<   Computing Time Information   >>

        (1) For BASES                H: M:  Sec
            Overhead            :    0: 0: 0.00
            Grid Optim. Step    :    0: 0: 0.98
            Integration Step    :    0: 0: 2.61
            Go time for all     :    0: 0: 3.59

        (2) Expected event generation time
            Expected time for 1000 events :      0.03 Sec
```

Date: 94/ 5/10 13:16

Convergency Behavior for the Grid Optimization Step

<- Result of IT	Eff	R_Neg	each iteration -> Estimate	Acc %	<- Cumulative Result Estimate(+- Error)order	Acc %	-> < CPU time > (H: M: Sec)
1	100	0.00	4.127E-02	3.801	4.127239(+-0.156879)E-02	3.801	0: 0: 0.09
2	100	0.00	4.536E-02	1.992	4.434463(+-0.078292)E-02	1.766	0: 0: 0.19
3	100	0.00	4.588E-02	0.852	4.557121(+-0.034986)E-02	0.768	0: 0: 0.29
4	100	0.00	4.557E-02	0.333	4.557166(+-0.013906)E-02	0.305	0: 0: 0.39
5	100	0.00	4.508E-02	0.274	4.529775(+-0.009239)E-02	0.204	0: 0: 0.49
6	100	0.00	4.532E-02	0.255	4.530635(+-0.007221)E-02	0.159	0: 0: 0.59
7	100	0.00	4.563E-02	0.278	4.538569(+-0.006277)E-02	0.138	0: 0: 0.68
8	100	0.00	4.538E-02	0.238	4.538356(+-0.005426)E-02	0.120	0: 0: 0.78
9	100	0.00	4.521E-02	0.242	4.534931(+-0.004861)E-02	0.107	0: 0: 0.88
10	100	0.00	4.541E-02	0.261	4.535849(+-0.004497)E-02	0.099	0: 0: 0.98

Date: 94/ 5/10 13:16

Convergency Behavior for the Integration Step

<- Result of IT	Eff	R_Neg	each iteration -> Estimate	Acc %	<- Cumulative Result Estimate(+- Error)order	Acc %	-> < CPU time > (H: M: Sec)
1	100	0.00	4.524E-02	0.258	4.523981(+-0.011689)E-02	0.258	0: 0: 1.08
2	100	0.00	4.514E-02	0.253	4.519085(+-0.008170)E-02	0.181	0: 0: 1.17
3	100	0.00	4.507E-02	0.247	4.514963(+-0.006589)E-02	0.146	0: 0: 1.27
4	100	0.00	4.513E-02	0.250	4.514549(+-0.005691)E-02	0.126	0: 0: 1.37
5	100	0.00	4.524E-02	0.264	4.516280(+-0.005136)E-02	0.114	0: 0: 1.46
6	100	0.00	4.508E-02	0.233	4.514726(+-0.004613)E-02	0.102	0: 0: 1.56
7	100	0.00	4.549E-02	0.268	4.518954(+-0.004315)E-02	0.095	0: 0: 1.66
8	100	0.00	4.522E-02	0.260	4.519278(+-0.004051)E-02	0.090	0: 0: 1.76
9	100	0.00	4.542E-02	0.258	4.521653(+-0.003830)E-02	0.085	0: 0: 1.85
10	100	0.00	4.547E-02	0.267	4.523973(+-0.003653)E-02	0.081	0: 0: 1.95
11	100	0.00	4.518E-02	0.248	4.523414(+-0.003473)E-02	0.077	0: 0: 2.05
12	100	0.00	4.524E-02	0.261	4.523487(+-0.003332)E-02	0.074	0: 0: 2.14
13	100	0.00	4.539E-02	0.256	4.524683(+-0.003203)E-02	0.071	0: 0: 2.24
14	100	0.00	4.514E-02	0.247	4.523864(+-0.003078)E-02	0.068	0: 0: 2.33
15	100	0.00	4.538E-02	0.277	4.524689(+-0.002990)E-02	0.066	0: 0: 2.43
16	100	0.00	4.520E-02	0.264	4.524436(+-0.002900)E-02	0.064	0: 0: 2.53
17	100	0.00	4.534E-02	0.275	4.524917(+-0.002825)E-02	0.062	0: 0: 2.62
18	100	0.00	4.538E-02	0.257	4.525643(+-0.002746)E-02	0.061	0: 0: 2.72
19	100	0.00	4.544E-02	0.270	4.526527(+-0.002679)E-02	0.059	0: 0: 2.82
20	100	0.00	4.533E-02	0.261	4.526821(+-0.002613)E-02	0.058	0: 0: 2.92
21	100	0.00	4.532E-02	0.270	4.527040(+-0.002556)E-02	0.056	0: 0: 3.01
22	100	0.00	4.530E-02	0.266	4.527173(+-0.002500)E-02	0.055	0: 0: 3.11
23	100	0.00	4.534E-02	0.265	4.527466(+-0.002447)E-02	0.054	0: 0: 3.21
24	100	0.00	4.514E-02	0.240	4.526793(+-0.002387)E-02	0.053	0: 0: 3.30
25	100	0.00	4.554E-02	0.272	4.527769(+-0.002344)E-02	0.052	0: 0: 3.40
26	100	0.00	4.532E-02	0.259	4.527919(+-0.002299)E-02	0.051	0: 0: 3.50
27	100	0.00	4.518E-02	0.258	4.527550(+-0.002255)E-02	0.050	0: 0: 3.59

Histogram (ID = 1) for Photon Energy (GeV)

Linear Scale indicated by "*"

x	d(Sigma)/dx	0.0E+00	1.5E-01	3.0E-01	4.5E-01
E 1	0.000	E 0			
0.001	3.189+-0.004	E -1	**********	0000000000000000000	
0.132	5.400+-0.011	E -2	*****	00000000000000000000000000000000000	
0.263	3.872+-0.004	E -2	****	0000000000000000000000000000000000	
0.395	3.333+-0.004	E -2	***	0000000000000000000000000000000000	
0.526	3.355+-0.004	E -2	***	0000000000000000000000000000000000	
0.657	3.564+-0.004	E -2	***	0000000000000000000000000000000000	
0.788	4.509+-0.007	E -2	****	0000000000000000000000000000000000	
0.920	6.206+-0.007	E -2	******	0000000000000000000000000000000000	
1.051	1.065+-0.001	E -1	*********	000000000000000000000000000000000	
1.182	2.429+-0.001	E -1	**********	0000000000000000000000	
1.313	4.737+-0.001	E -1	**********	00000000	
1.444	2.319+-0.001	E -1	**********	0000000000000000000000	
1.576	7.006+-0.014	E -2	******	000000000000000000000000000000000	
1.707	2.838+-0.043	E -2	***	0000000000000000000000000000000000	
1.838	1.422+-0.011	E -2	**	0000000000000000000000000000000000	
1.969	7.623+-0.047	E -3		0000000000000000000000000000000000	
2.101	4.806+-0.026	E -3		0000000000000000000000000000000000	
2.232	2.987+-0.134	E -3		0000000000000000000000000000000	
2.363	1.779+-0.021	E -3		0000000000000000000000000000000	
2.494	1.418+-0.016	E -3		0000000000000000000000000000000	
2.625	1.027+-0.011	E -3		0000000000000000000000000000000	
2.757	7.102+-0.069	E -4		000000000000000000000000000000	
2.888	4.934+-0.050	E -4		000000000000000000000000000000	
3.019	4.752+-0.040	E -4		000000000000000000000000000	
3.150	3.360+-0.029	E -4		000000000000000000000000000	
3.282	2.459+-0.023	E -4		000000000000000000000000	
3.413	1.800+-0.019	E -4		000000000000000000000	
3.544	1.463+-0.015	E -4		000000000000000000000	
3.675	1.127+-0.012	E -4		000000000000000000000	
3.807	7.087+-0.087	E -5		0000000000000000	
3.938	5.919+-0.072	E -5		0000000000000000	
4.069	4.704+-0.061	E -5		0000000000000000	
4.200	3.655+-0.050	E -5		0000000000000000	
4.331	3.252+-0.045	E -5		0000000000000000	
4.463	2.079+-0.033	E -5		0000000000000000	
4.594	1.428+-0.026	E -5		0000000000000000	
4.725	8.248+-0.170	E -6		00000000	
4.856	6.630+-0.141	E -6		00000000	
4.988	3.585+-0.099	E -6		0000	
5.119	1.097+-0.066	E -6		0	
E 1	0.000	E 0			

x	d(Sigma)/dx	1.0E-06	1.0E-04	1.0E-02

Logarithmic Scale indicated by "0"

Histogram (ID = 2) for Cos(theta) of Photon

Linear Scale indicated by "*"

x	d(Sigma)/dx	0.0E+00	2.0E+00	4.0E+00	6.0E+00
E -1	0.000	E 0			
-9.659	5.966+-0.011	E 0	**********	0000000000	
-9.273	3.389+-0.007	E 0	**********	0000000000000000000	
-8.887	2.440+-0.007	E 0	**********	0000000000000000000	
-8.500	1.928+-0.006	E 0	**********	0000000000000000000	
-8.114	1.547+-0.005	E 0	**********	0000000000000000000	
-7.727	1.380+-0.005	E 0	*********	00000000000000000000	
-7.341	1.206+-0.005	E 0	*********	00000000000000000000	
-6.955	1.071+-0.005	E 0	********	000000000000000000000	
-6.568	9.802+-0.041	E -1	*******	00000000000000000000	
-6.182	8.913+-0.041	E -1	*******	00000000000000000000	
-5.796	8.542+-0.041	E -1	******	000000000000000000000	
-5.409	8.083+-0.037	E -1	******	00000000000000000000	
-5.023	7.912+-0.039	E -1	*****	000000000000000000000	
-4.636	7.329+-0.033	E -1	*****	000000000000000000000	
-4.250	7.156+-0.033	E -1	*****	000000000000000000000	
-3.864	6.827+-0.031	E -1	*****	000000000000000000000	
-3.477	6.531+-0.033	E -1	*****	00000000000000000000	
-3.091	6.596+-0.031	E -1	*****	00000000000000000000	
-2.705	6.283+-0.034	E -1	****	000000000000000000000	
-2.318	6.309+-0.032	E -1	****	000000000000000000000	
-1.932	5.653+-0.031	E -1	****	00000000000000000	
-1.545	5.812+-0.032	E -1	****	000000000000000000	
-1.159	5.693+-0.033	E -1	****	00000000000000000	
-0.773	5.821+-0.035	E -1	****	000000000000000000	
-0.386	5.738+-0.029	E -1	****	000000000000000000	
0.000	5.953+-0.030	E -1	****	000000000000000000	
0.386	5.851+-0.031	E -1	****	000000000000000000	
0.773	5.872+-0.032	E -1	****	000000000000000000	
1.159	6.024+-0.029	E -1	****	000000000000000000	
1.545	6.056+-0.032	E -1	****	000000000000000000	
1.932	5.965+-0.029	E -1	****	000000000000000000	
2.318	6.198+-0.032	E -1	****	000000000000000000	
2.705	6.235+-0.029	E -1	****	000000000000000000	
3.091	6.286+-0.029	E -1	****	000000000000000000	
3.477	6.592+-0.033	E -1	*****	00000000000000000	
3.864	7.099+-0.033	E -1	*****	000000000000000000	
4.250	7.239+-0.035	E -1	*****	000000000000000000	
4.636	7.413+-0.032	E -1	*****	000000000000000000	
5.023	7.940+-0.038	E -1	*****	000000000000000000	
5.409	8.356+-0.038	E -1	******	00000000000000000	
5.796	9.330+-0.038	E -1	******	00000000000000000	
6.182	9.715+-0.041	E -1	*******	000000000000000000	
6.568	1.087+-0.004	E 0	*******	00000000000000000	
6.955	1.165+-0.005	E 0	********	000000000000000000	
7.341	1.367+-0.005	E 0	*********	00000000000000000	
7.727	1.588+-0.005	E 0	**********	000000000000000000	
8.114	1.909+-0.006	E 0	**********	000000000000000000	
8.500	2.434+-0.006	E 0	**********	000000000000000000	
8.887	3.380+-0.007	E 0	**********	000000000000000000	
9.273	5.963+-0.011	E 0	**********	0000000000	
E -1	0.000	E 0			

x	d(Sigma)/dx	1.0E-01	1.0E+00

Logarithmic Scale indicated by "0"

```
                                                        Date: 94/ 5/10  13:16
 ************************************************************
 *                                                          *
 *    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         =    69.396 Percent
 Computing time for generation =     0.410 Seconds
                for Overhead   =     0.199 Seconds
                for Others     =     0.136 Seconds
 GO time for event generation  =     0.745 Seconds
 Max. number of trials MXTRY   =        50 per event
 Number of miss-generation     =         0 times
```

```
            ************** Number of trials to get an event **************
 Total =      10000 events    "*" : No. of events in Linear scale.
    x       Lg(dN/dx)  dN/dx 0.0E+00       2.3E+03       4.5E+03       6.8E+03
+--------+----------+-------+------------+------------+-----------+-----------+
I  0.100 I 7.419E  3I   7419I**********************************************0000000I
I  0.200 I 1.779E  3I   1779I**********00000000000000000000000000000000000      I
I  0.300 I 4.800E  2I    480I***0000000000000000000000000000000               I
I  0.400 I 1.530E  2I    153I*000000000000000000000000000000                  I
I  0.500 I 5.800E  1I     58I*00000000000000000000                            I
I  0.600 I 2.900E  1I     29I*0000000000000000000                             I
I  0.700 I 1.600E  1I     16I*00000000000000000                               I
I  0.800 I 1.600E  1I     16I*00000000000000000                               I
I  0.900 I 1.000E  1I     10I*0000000000000                                   I
I  1.000 I 1.000E  1I     10I*0000000000000                                   I
I  1.100 I 6.000E  0I      6I*000000000                                       I
I  1.200 I 3.000E  0I      3I*00000                                           I
I  1.300 I 6.000E  0I      6I*000000000                                       I
I  1.400 I 1.000E  0I      1I0                                                I
I  1.500 I 2.000E  0I      2I*000                                             I
I  1.600 I 1.000E  0I      1I0                                                I
I  1.700 I 4.000E  0I      4I*0000000                                         I
I  1.800 I 0.000E  0I      0I                                                 I
I  1.900 I 1.000E  0I      1I0                                                I
I  2.000 I 1.000E  0I      1I0                                                I
I  2.100 I 0.000E  0I      0I                                                 I
I  2.200 I 1.000E  0I      1I0                                                I
I  2.300 I 0.000E  0I      0I                                                 I
I  2.400 I 0.000E  0I      0I                                                 I
I  2.500 I 1.000E  0I      1I0                                                I
I  2.600 I 1.000E  0I      1I0                                                I
I  2.700 I 1.000E  0I      1I0                                                I
I  2.800 I 0.000E  0I      0I                                                 I
I  2.900 I 0.000E  0I      0I                                                 I
I  3.000 I 0.000E  0I      0I                                                 I
I  3.100 I 0.000E  0I      0I                                                 I
I  3.200 I 0.000E  0I      0I                                                 I
I  3.300 I 0.000E  0I      0I                                                 I
I  3.400 I 0.000E  0I      0I                                                 I
I  3.500 I 0.000E  0I      0I                                                 I
I  3.600 I 0.000E  0I      0I                                                 I
I  3.700 I 0.000E  0I      0I                                                 I
I  3.800 I 0.000E  0I      0I                                                 I
I  3.900 I 0.000E  0I      0I                                                 I
I  4.000 I 1.000E  0I      1I0                                                I
I  4.100 I 0.000E  0I      0I                                                 I
I  4.200 I 0.000E  0I      0I                                                 I
I  4.300 I 0.000E  0I      0I                                                 I
I  4.400 I 0.000E  0I      0I                                                 I
I  4.500 I 0.000E  0I      0I                                                 I
I  4.600 I 0.000E  0I      0I                                                 I
I  4.700 I 0.000E  0I      0I                                                 I
I  4.800 I 0.000E  0I      0I                                                 I
I  4.900 I 0.000E  0I      0I                                                 I
I  5.000 I 0.000E  0I      0I                                                 I
I   E  1 I 0.000E  0I      0I                                                 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.
```

```
Original Histogram (ID =  1 ) for Photon Energy (GeV)
Total =     10000 events   "*" : Orig. Dist. in Log Scale.
   x      d(Sig/dx)  dN/dx 1.0E-06          1.0E-04          1.0E-02
+--------+----------+-------+----------------+----------------+----------------+
I   E  1 I 0.000E  0I      0I                                                   I
I  0.001 I 3.189E -1I   1743I**********************************************0   I
I  0.132 I 5.400E -2I    322I*****************************************0        I
I  0.263 I 3.872E -2I    210I****************************************<>        I
I  0.395 I 3.333E -2I    177I***************************************0         I
I  0.526 I 3.355E -2I    172I***************************************0         I
I  0.657 I 3.564E -2I    192I****************************************<>       I
I  0.788 I 4.509E -2I    261I*****************************************<>     I
I  0.920 I 6.206E -2I    336I******************************************<>    I
I  1.051 I 1.065E -1I    546I*********************************************0   I
I  1.182 I 2.429E -1I   1349I*************************************************0 I
I  1.313 I 4.737E -1I   2677I*************************************************** I
I  1.444 I 2.319E -1I   1263I*************************************************0 I
I  1.576 I 7.006E -2I    386I********************************************0     I
I  1.707 I 2.838E -2I    175I*****************************************0        I
I  1.838 I 1.422E -2I     81I*************************************<>          I
I  1.969 I 7.623E -3I     31I*******************************<>*                I
I  2.101 I 4.806E -3I     25I*****************************<0>                  I
I  2.232 I 2.987E -3I     15I****************************<0>                    I
I  2.363 I 1.779E -3I     10I**************************<*0>                     I
I  2.494 I 1.418E -3I      5I************************<*0>*                      I
I  2.625 I 1.027E -3I      4I***********************<*0*>                       I
I  2.757 I 7.102E -4I      4I***********************<*0 >                       I
I  2.888 I 4.934E -4I      4I***********************<*0 >                       I
I  3.019 I 4.752E -4I      5I************************< 0>                       I
I  3.150 I 3.360E -4I      4I************************< 0 >                      I
I  3.282 I 2.459E -4I      0I**********************                            I
I  3.413 I 1.800E -4I      1I******************0  >                            I
I  3.544 I 1.463E -4I      0I******************                               I
I  3.675 I 1.127E -4I      1I*******************0  >                           I
I  3.807 I 7.087E -5I      0I*****************                                 I
I  3.938 I 5.919E -5I      1I****************   0  >                          I
I  4.069 I 4.704E -5I      0I***************                                   I
I  4.200 I 3.655E -5I      0I**************                                    I
I  4.331 I 3.252E -5I      0I**************                                    I
I  4.463 I 2.079E -5I      0I***********                                      I
I  4.594 I 1.428E -5I      0I**********                                       I
I  4.725 I 8.248E -6I      0I********                                         I
I  4.856 I 6.630E -6I      0I*******                                          I
I  4.988 I 3.585E -6I      0I*****                                            I
I  5.119 I 1.097E -6I      0I*                                                I
I   E  1 I 0.000E  0I      0I                                                 I
+--------+----------+-------+----------------+----------------+----------------+
   x      d(Sig/dx)  dN/dx    "0" : Generated Events.( Arbitrary unit in Log )
```

```
Original Histogram (ID =  2 ) for Cos(theta) of Photon
Total =     10000 events   "*" : Orig. Dist. in Log Scale.
   x      d(Sig/dx)  dN/dx 1.0E-01                    1.0E+00
+--------+----------+-------+----------------------------+--------------------+
I   E -1 I 0.000E  0I      0I                                                  I
I -9.659 I 5.966E  0I   1010I**************************************************<
I -9.273 I 3.389E  0I    593I*******************************************<>     I
I -8.887 I 2.440E  0I    392I*************************************<>           I
I -8.500 I 1.928E  0I    301I*********************************<>              I
I -8.114 I 1.547E  0I    234I*****************************<>*                  I
I -7.727 I 1.380E  0I    240I*****************************<0>                  I
I -7.341 I 1.206E  0I    171I***************************<0>*                   I
I -6.955 I 1.071E  0I    183I*****************************<0>                  I
I -6.568 I 9.802E -1I    198I***************************** <>                  I
I -6.182 I 8.913E -1I    141I***************************<0>                    I
I -5.796 I 8.542E -1I    137I**************************<0>                     I
I -5.409 I 8.083E -1I    146I**************************<0>                     I
I -5.023 I 7.912E -1I    124I************************<0>                       I
I -4.636 I 7.329E -1I    121I************************<0>                       I
I -4.250 I 7.156E -1I    111I**********************<*0>                       I
I -3.864 I 6.827E -1I    110I**********************<*0>                       I
I -3.477 I 6.531E -1I    105I**********************<0>                       I
I -3.091 I 6.596E -1I     89I********************<0>*                        I
I -2.705 I 6.283E -1I     87I********************<*0>*                       I
I -2.318 I 6.309E -1I     90I********************<0>*                        I
I -1.932 I 5.653E -1I     95I**********************<*0>                       I
I -1.545 I 5.812E -1I     91I**********************<0>                       I
I -1.159 I 5.693E -1I    112I********************** <0>                       I
I -0.773 I 5.821E -1I     91I********************<0>                        I
I -0.386 I 5.738E -1I     84I********************<0*>                        I
I  0.000 I 5.953E -1I     77I******************<0*>*                         I
I  0.386 I 5.851E -1I    101I**********************<*0>                       I
I  0.773 I 5.872E -1I     75I****************<*0>**                          I
I  1.159 I 6.024E -1I     80I******************<*0>*                         I
I  1.545 I 6.056E -1I     94I********************<*0>                        I
I  1.932 I 5.965E -1I    107I************************<0>                      I
I  2.318 I 6.198E -1I    114I************************<0>                      I
I  2.705 I 6.235E -1I    107I************************<0>                      I
I  3.091 I 6.286E -1I    124I**************************<0>                     I
I  3.477 I 6.592E -1I    112I************************<0>                      I
I  3.864 I 7.099E -1I    108I************************<0>                      I
I  4.250 I 7.239E -1I    106I************************<0>                      I
I  4.636 I 7.413E -1I    116I*************************<0>                     I
I  5.023 I 7.940E -1I    141I****************************<0>                   I
I  5.409 I 8.356E -1I    115I************************<0>*                      I
I  5.796 I 9.330E -1I    159I*********************************<0>             I
I  6.182 I 9.715E -1I    119I************************<*0>**                    I
I  6.568 I 1.087E  0I    177I*****************************<0>                  I
I  6.955 I 1.165E  0I    172I****************************<0>                   I
I  7.341 I 1.367E  0I    206I*******************************<0>               I
I  7.727 I 1.588E  0I    256I************************************<>           I
I  8.114 I 1.909E  0I    324I*************************************<>          I
I  8.500 I 2.434E  0I    413I*****************************************<>      I
I  8.887 I 3.380E  0I    594I*************************************************<>I
I  9.273 I 5.963E  0I    947I**************************************************<>
I   E -1 I 0.000E  0I      0I                                                  I
+--------+----------+-------+----------------------------+--------------------+
   x      d(Sig/dx)  dN/dx    "0" : Generated Events.( Arbitrary unit in Log )
```

```
Additional Histogram (ID =  5 ) for Photon Transverse Energy (GeV)
Total =     10000 events   "*" : No. of events in Linear scale.
    x      Lg(dN/dx)  dN/dx 0.0E+00      6.3E+02      1.3E+03     1.9E+03
+--------+---------+-------+------------+------------+-----------+-----------+
I   E  1 I 0.000E  0I      0I                                                I
I  0.000 I 1.866E  3I   1866I****************************************00000000000
I  0.100 I 3.590E  2I    359I********000000000000000000000000000000          I
I  0.200 I 3.140E  2I    314I*******000000000000000000000000000000           I
I  0.300 I 6.330E  2I    633I*************00000000000000000000000000000      I
I  0.400 I 8.960E  2I    896I*******************0000000000000000000000000    I
I  0.500 I 8.600E  2I    860I*******************0000000000000000000000000    I
I  0.600 I 6.900E  2I    690I**************000000000000000000000000000000    I
I  0.700 I 6.230E  2I    623I*************00000000000000000000000000000000   I
I  0.800 I 5.240E  2I    524I***********000000000000000000000000000000000    I
I  0.900 I 5.240E  2I    524I***********000000000000000000000000000000000    I
I  1.000 I 5.250E  2I    525I***********000000000000000000000000000000000    I
I  1.100 I 5.380E  2I    538I***********000000000000000000000000000000000    I
I  1.200 I 5.490E  2I    549I***********000000000000000000000000000000000    I
I  1.300 I 5.570E  2I    557I************00000000000000000000000000000000    I
I  1.400 I 2.960E  2I    296I******000000000000000000000000000000000         I
I  1.500 I 1.140E  2I    114I***00000000000000000000000000000                I
I  1.600 I 5.200E  1I     52I**000000000000000000000000000                   I
I  1.700 I 2.800E  1I     28I*0000000000000000000000                         I
I  1.800 I 1.600E  1I     16I*00000000000000000000                           I
I  1.900 I 1.300E  1I     13I*0000000000000000                               I
I  2.000 I 4.000E  0I      4I*000000000                                      I
I  2.100 I 2.000E  0I      2I*0000                                           I
I  2.200 I 5.000E  0I      5I*0000000000                                     I
I  2.300 I 3.000E  0I      3I*0000000                                        I
I  2.400 I 2.000E  0I      2I*0000                                           I
I  2.500 I 1.000E  0I      1I0                                               I
I  2.600 I 3.000E  0I      3I*0000000                                        I
I  2.700 I 0.000E  0I      0I                                                I
I  2.800 I 1.000E  0I      1I0                                               I
I  2.900 I 1.000E  0I      1I0                                               I
I  3.000 I 1.000E  0I      1I0                                               I
I  3.100 I 0.000E  0I      0I                                                I
I  3.200 I 0.000E  0I      0I                                                I
I  3.300 I 0.000E  0I      0I                                                I
I  3.400 I 0.000E  0I      0I                                                I
I  3.500 I 0.000E  0I      0I                                                I
I  3.600 I 0.000E  0I      0I                                                I
I  3.700 I 0.000E  0I      0I                                                I
I  3.800 I 0.000E  0I      0I                                                I
I  3.900 I 0.000E  0I      0I                                                I
I   E  1 I 0.000E  0I      0I                                                I
+--------+---------+-------+-------------+-------------+-------------+---+
    x      Lg(dN/dx)  dN/dx 1.0E+00       1.0E+01       1.0E+02
                       "0" : No. of Events in Log. scale.
```

```
Scat_Plot (ID =  1 ) for  x : cos(theta)  --  y : Photon Energy

  E  1   +--------------------------------------------------+
 5.145   I                                                  I
 5.040   I                                                  I
 4.935   I                                                  I
 4.830   I                                                  I
 4.725   I                                                  I
 4.620   I                                                  I
 4.515   I                                                  I
 4.410   I                                                  I
 4.305   I                                                  I
 4.200   I                                                  I
 4.095   I                                                  I
 3.990   I                                                  I
 3.885   I                                              .   I
 3.780   I                                                  I
 3.675   I                                                 .I
 3.570   I                                                  I
 3.465   I                                                  I
 3.360   I                                              .   I
 3.255   I                                          .       I
 3.150   I                            .       .       .     I
 3.045   I.                                       . .    .  I
 2.940   I                          .                       I
 2.835   I        .                                 .     . I
 2.730   I ..                                   .     .     I
 2.625 Y I  .   .                .                 .        I
 2.520   I.         .                                      .I
 2.416   I....              .                          ...  I
 2.311   I                         . .                .   . I
 2.206   I. ..   ..    .          .                .     . .I
 2.101   I. .    .  .   .. .                     .   ..... .I
 1.996   I..   .. .          .        .      . .. .. ... ...I
 1.891   I....  ...              ......  . . .   .... ... ....I
 1.786   I..............  .... .. .. ... ...... ...........I
 1.681   I1.................  . ........ .. ..............I
 1.576   I11.................... ... ...... ..........11I
 1.471   I43111.....................................11224I
 1.366   I*63322112111111111111111..1.111111.111111.2121346*I
 1.261   I9542112111111.1...1...1.....1..1.11.111111111122358I
 1.156   I4211.1....................................1.11223I
 1.051   I11................. ...... ....................111I
 0.946   I1.... ........................... ..............1I
 0.841   I1............  ....  .............. ..........1I
 0.736   I........ .... ...... ... ...... ....... .. .........I
 0.631   I............. ... ......  ...... ............I
 0.526   I.......... ..... . ... . ... ..... . ... .........I
 0.421   I........  .. . ........ ..... .... .. .........I
 0.316   I....... .... ........  .......... ..... ..........I
 0.211   I1........ .......................................1I
 0.106   I1............... ............. ...............1.1I
 0.001   I185322211211211..1....1.........1.111....1.11222358I
  E  1   +--------------------------------------------------+
 Low-                          X
 Edge     -------------------------
          99888776665554433322111000001112233344555666778889
  E -1    ..................................................
          62851739517406284073951730371593704826047159371582
 Low-     57801245689023567901345780875431097653209865421087
 Edge     92603714815926037048159260629518407306295184173062
```