An aerial photograph of a rural landscape. In the background, a large, dark green mountain rises against a blue sky with light clouds. The middle ground shows a valley with a patchwork of green and brown fields, interspersed with small buildings and roads. In the foreground, a large school complex is visible, featuring several large buildings, a prominent circular structure, and a parking lot. The overall scene is a mix of natural and built environments.

*Numerical Integration  
and  
Event Generation  
in  
High Energy Physics*

Sept. 3rd '2009  
Grace School at KEK  
Setsuya Kawabata



# *Contents*

---

***1. Introduction***

***2. Monte Carlo Integration Method***

***3. Numerical Integration Package BASES***

***4. Event Generation***

***5. Event Generation Package SPRING***

***6. Advantages of BASES/SPRING***

***7. Vulnerability of BASES***

***8. A question from User***



# 1. Introduction

---

**Why Monte Carlo integration ?**

**For designing an Experiment**

**Need to know**

**The event rate of an interesting elementary process**



**Numerical calculation of the cross section**



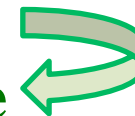
**Integration of the differential Cross section**

**Example:**

$e^+ e^- \rightarrow e^+ e^-$  : only 1 variable, e.g.  $\cos\theta$

$e^+ e^- \rightarrow e^+ e^- g$  : 4 independent variables

**Analytical Integration is impossible**





# *1. Introduction (2)*

---

## **Why Event generation ?**

**In physics analysis**

**Need to know**

**The detection efficiency for the interesting event**

**Not all of events can be detected by the detector due to**

- a) Trigger efficiency**
- b) Geometrical boundary of detector**
- c) Various cuts to select the interesting events**

**against background events.**

**These effects cannot be described by any analytic function**

**To know the efficiency**

**We generate events with weight one and  
pass them through the detector simulation,  
then analyze them as if they are real data.**



# 1. Introduction (3)

**How calculate the cross section ?**

1. Draw all Feynman diagrams

2. Write down the amplitudes,  $T = \sum T_i$   
and get the matrix element squared,  $\tau = \sum_{spin,pol} \sum_{i,j} T_i^\dagger T_j$

3. Add the kinematics part

4. Integrate over the phase space

**Each step is achieved by**

1.  $\Rightarrow$  by hand

2.  $\Rightarrow$  by symbolic manipulation program REDUCE etc.

3.  $\Rightarrow$  by hand

4.  $\Rightarrow$  by an adaptive Monte Carlo integration package



## 2. Monte Carlo Integration Method

For simplicity, consider an integral

$$I = \int_a^b f(x) dx$$

### (1) Crude Monte Carlo Method

Meaning of the integral

$$\begin{aligned} I &= \int_a^b f(x) dx = \text{size of Area} \quad \text{[Small green area diagram]} \\ &= (b-a) \overline{f(x)} = \lim_{N \rightarrow \infty} \frac{b-a}{N} \sum_{i=1}^N f(x_i) \end{aligned}$$

The estimate of the integral

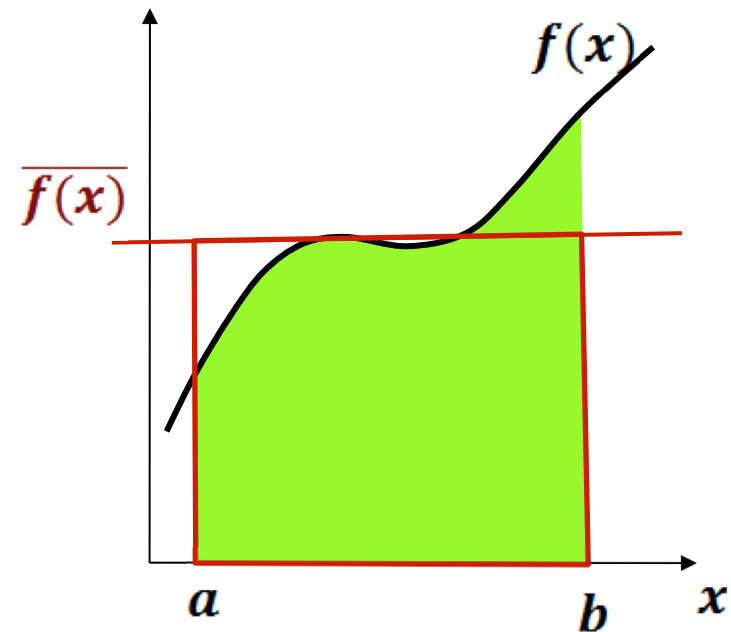
$$I \cong \frac{b-a}{N} \sum_{i=1}^N f(x_i)$$

The variance

$$\begin{aligned} \text{var.}(I) &= \left(\frac{b-a}{N}\right)^2 N \text{var.}(f(x)) \\ &= \frac{1}{N} \left[ (b-a) \int_a^b \{f(x)\}^2 dx - I^2 \right] \end{aligned}$$

The error of the estimate

$$\varepsilon = \sqrt{\text{var.}(I)} \propto \frac{c}{\sqrt{N}}$$





## 2. Monte Carlo Integration Method (2)

### (2) Stratified Sampling Method

Divide the integration region  $[a, b]$  into  $m$  sub-regions

$[a_1, b_1], [a_2, b_2], \dots, [a_m, b_m]$ , where  $a = a_1 < b_1 = a_2 < b_2 = a_3 < \dots < b_m = b$ .

The integral becomes

$$I = \int_a^b f(x) dx = \sum_{j=1}^m \int_{a_j}^{b_j} f(x) dx$$

Estimate of the integral

$$I \cong \sum_{j=1}^m \frac{b_j - a_j}{N_j} \left[ \sum_{i=1}^{N_j} f(x_i) \right]$$

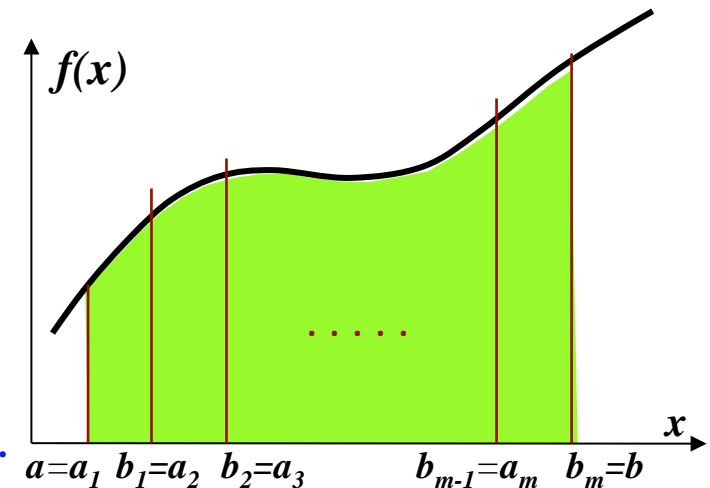
$N_j$  uniform random numbers are used for the  $j$ -th sub-region.

Variance

$$I \cong \sum_{j=1}^m \left( \frac{b_j - a_j}{N_j} \right)^2 N_j \text{var.}(f(x_j))$$

To make the variance small,

- Into the sub-regions with large variance, throw more sample points, i.e. make  $N_j$  large
- Make the width of such sub-region  $[a_j, b_j]$  narrower, so as to make  $\text{var.}(f(x_j))$  small.





## 2. Monte Carlo Integration Method (3)

### (3) Importance Sampling Method

Modify the integral as

$$I = \int_a^b f(x) dx = \int_a^b \frac{f(x)}{p(x)} p(x) dx \quad \text{and require} \quad \int_a^b p(x) dx = 1 ,$$

then the integral can be read as

*The expectation value of function  $\frac{f(x)}{p(x)}$   
with the probability density  $p(x)$ .*

Estimate of the integral

$$I = E_{xp} \left[ \frac{f(\zeta)}{p(\zeta)} \right] = \frac{1}{N} \sum_{i=1}^N \frac{f(\zeta_i)}{p(\zeta_i)}$$

where  $\mathbf{z}$  is the random number with the probability density  $p(x)$ .

Variance

$$\begin{aligned} \text{var.}(I) &= \left( \frac{1}{N} \right)^2 \text{var.} \left[ \frac{f(\zeta)}{p(\zeta)} \right] = \left( \frac{1}{N} \right)^2 \sum_{i=1}^N \left[ \frac{f(\zeta_i)}{p(\zeta_i)} - \frac{1}{N} \sum_{j=1}^N \frac{f(\zeta_j)}{p(\zeta_j)} \right]^2 \\ &= \frac{1}{N} \left[ \int_a^b \frac{f(x)^2}{p(x)} dx - I^2 \right] \end{aligned}$$





## 2. Monte Carlo Integration Method (4)

### (3) Importance Sampling Method (continued)

If the probability density were given by

$$p(x) = \frac{|f(x)|}{\int_a^b f(y)dy}$$

the variance could become zero:

$$\text{var.}(I) = \frac{1}{N} \left[ \int_a^b \frac{f(x)^2}{p(x)} dx - I^2 \right] = \frac{1}{N} \left[ \int_a^b \frac{f(x)^2}{|f(x)|} \left\{ \int_a^b f(y)dy \right\} dx - I^2 \right] = 0$$

In practice, take those prob. density which satisfies ;

$$\frac{|f(x)|}{p(x)} \cong \text{const.} \quad \Rightarrow$$

*The prob. density  $p(x)$   
is chosen to be large  
(many sample points are thrown)  
at the point where  $|f(x)|$  is large.*

**Program package BASES uses this importance sampling method as VEGAS does.**



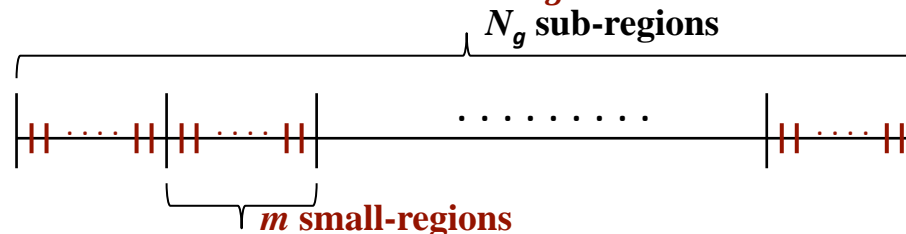
### 3. Numerical Integration Package BASES

For simplicity, consider the one dimensional integration:

$$I = \int_0^1 f(x) dx = \int_0^1 \frac{f(x)}{p(x)} p(x) dx$$

■ Probability density  $p(x)$

■ Divide the integration region  $[0, 1]$  into  $N_g$  sub-regions



Divide each sub-region into  $m$  small-regions. Total number of small-regions :  $N_S = N_g \times m$

■ Sampling Method

- Each small-region is sampled with the probability  $\frac{1}{N_S}$ .
- Probability density in each small-region is uniform.

Suppose  $\left\{ \begin{array}{l} \Delta x_{kl} : \text{Width of the } l\text{-th small region in the } k\text{-th sub-region} \\ p(x_{kl}) : \text{Probability density at a point } x_{kl} \text{ in the small region } \Delta x_{kl} \end{array} \right.$

then  $\frac{1}{N_S} = p(x_{kl}) \Delta x_{kl}$ .

Probability density:

■  $p(x_{kl}) = \frac{1}{N_S \Delta x_{kl}}$ , which satisfies  $\sum_{k=1}^{N_g} \sum_{l=1}^m p(x_{kl}) = \sum_{k=1}^{N_g} \sum_{l=1}^m \frac{1}{N_g m} = 1$



## 3. Numerical Integration Package BASES (2)

### ■ Adjustment of widths of small-regions

#### Importance sampling

$$\frac{|f(x)|}{p(x)} = |f(x)|\Delta x N_s \cong \text{const.}$$

⇒ Adjust widths so as to be

$$|f(x)|/Dx \cong \text{const.}$$

#### Procedure to adjust widths

- (1) At the beginning,  
uniform width for all small-regions.
- (2) During one iteration of estimation,  
a histogram of  $\frac{|f(x)|}{p(x)}$  is made.
- (3) After each iteration,

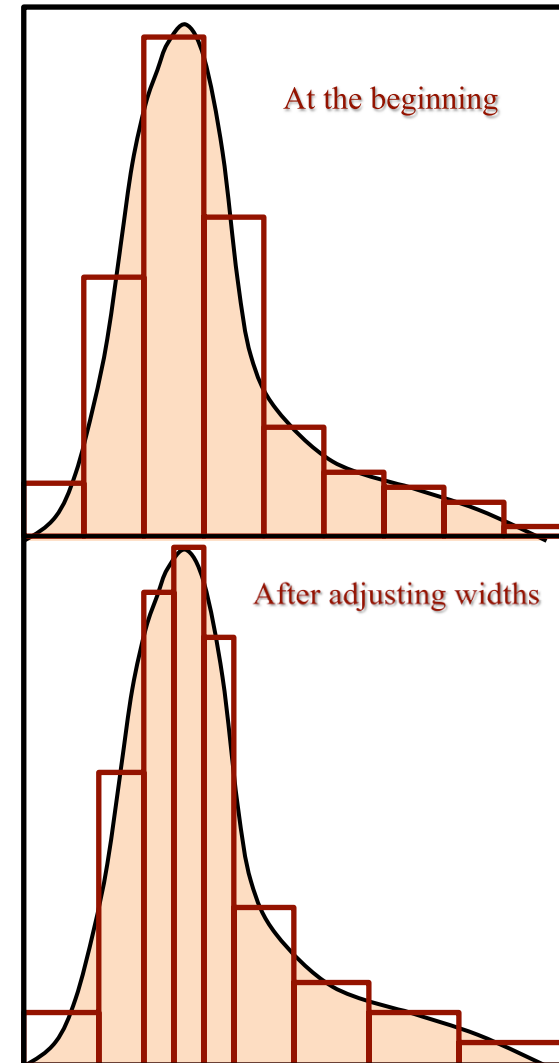
#### The widths are adjusted as follows:

$S$  : Total area of histogram

$N_s$  : the number of small-regions

The new boundaries of small-regions are determined so that

$$\text{All histogram bins have an equal area } \frac{S}{N_s}.$$





## 3. Numerical Integration Package BASES (3)

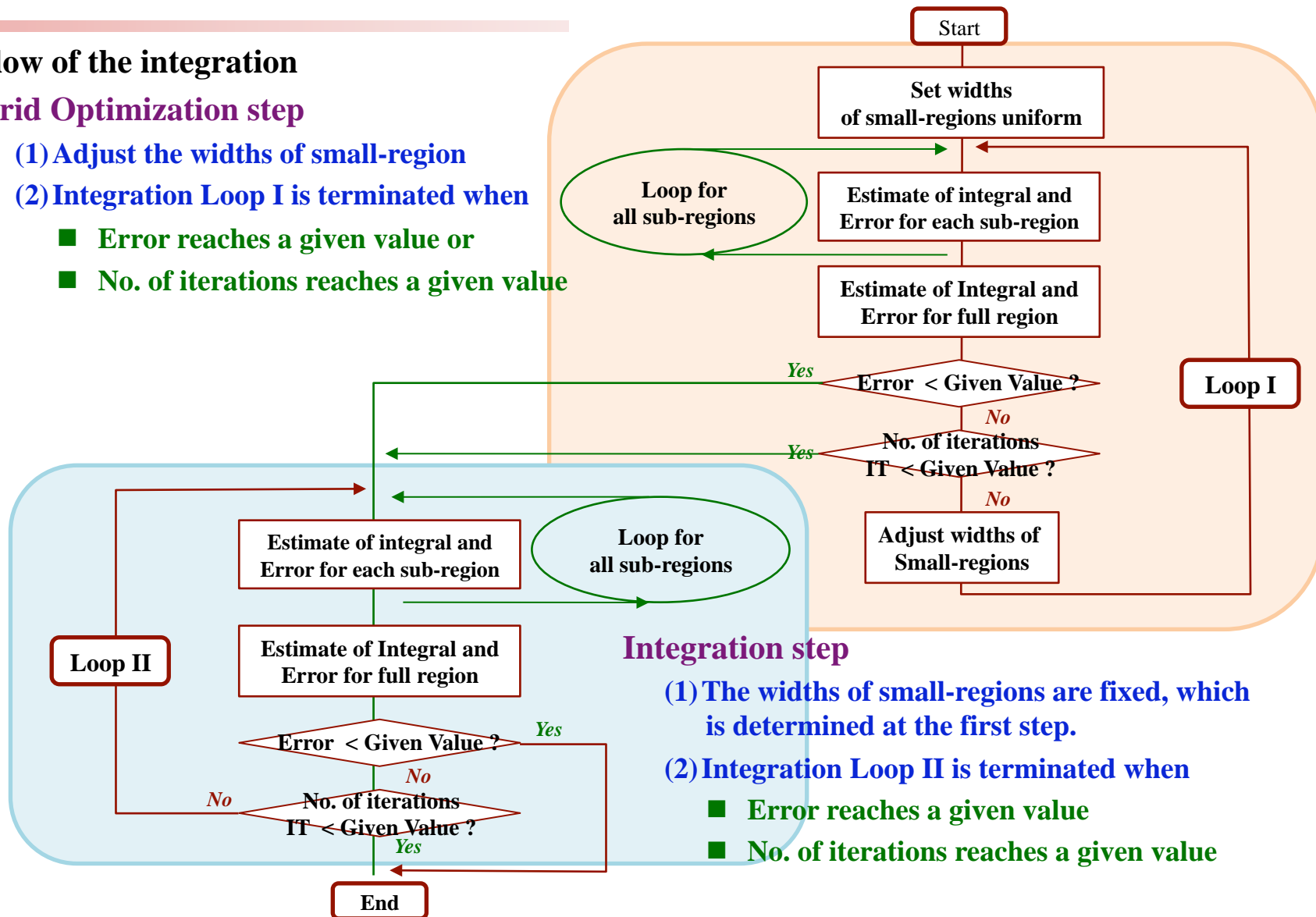
### Flow of the integration

#### Grid Optimization step

(1) Adjust the widths of small-region

(2) Integration Loop I is terminated when

- Error reaches a given value or
- No. of iterations reaches a given value





## 4. Event Generation

Generation of many sets of values of variables  $x_1, x_2, \dots, x_n$

which reproduce the distribution of function  $f(x_1, x_2, \dots, x_n)$ .

$$f(x_1, x_2, \dots, x_n) = \frac{d^n \sigma(x_1, x_2, \dots, x_n)}{dx_1 dx_2 \dots dx_n}$$

**Event generation = Generation of many sets of Random numbers**

**with a given specific distribution**

### ■ Direct Method

A function  $f(x)$  defined in  $[a, b]$  gives

the probability density  $p(x)$  :

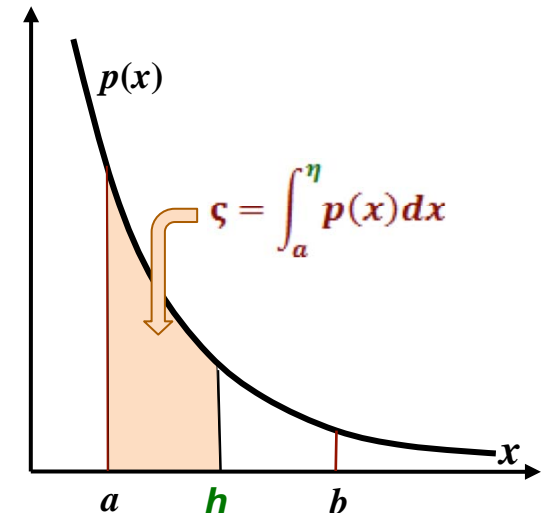
$$p(x) = \frac{1}{\int_a^b f(y) dy} f(x) , \quad \int_a^b p(x) dx = 1$$

If the variable  $z$  , defined by

$$\zeta = \int_a^\eta p(x) dx \quad \text{or} \quad \zeta = \frac{\int_a^\eta f(y) dy}{\int_a^b f(y) dy} ,$$

has a uniform distribution in the region  $[0,1]$ ,

the variable  $h$  has a distribution with the probability density  $p(x)$ .



- [NB]**
- The function  $f(x)$  must have the generating function  $F(x) = \int f(x) dx$
  - The generating function  $y = F(x)$  must have the inverse  $x = F^{-1}(y)$  .



## 4. Event Generation (2)

### ■ Example of Direct Method

$$1. f(x) = \frac{1}{x} \text{ in region } [a, b]: \frac{\int_a^\eta \frac{1}{x} dx}{\int_a^b \frac{1}{x} dx} = \zeta \quad \longrightarrow \quad \eta = a \left( \frac{b}{a} \right)^\zeta$$

$$2. f(x) = e^{-x} (y > 0): \int_0^\eta e^{-x} dx = \zeta \quad \longrightarrow \quad \eta = -\log \zeta$$

### ■ Rejection Method

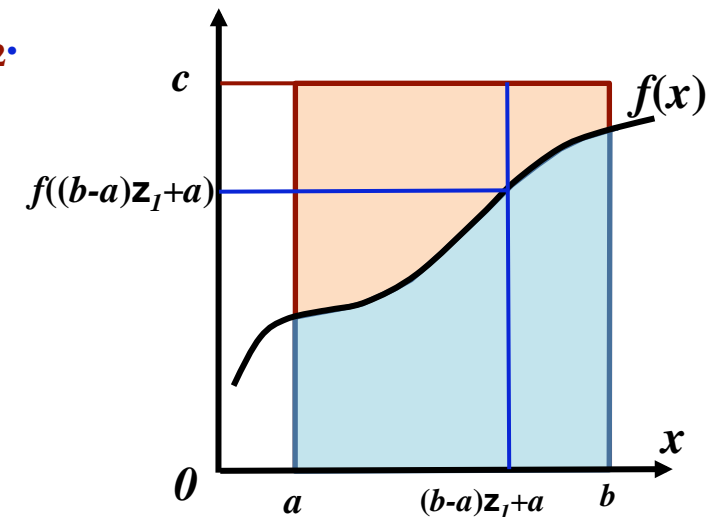
A function  $f(x)$  is given in  $[a, b]$

■ Take two uniform random numbers  $z_1$  and  $z_2$ .

■ If  $\zeta_2 \leq \frac{f((b-a)\zeta_1 + a)}{c}$ ,

the value  $x = (b-a)z_1 + a$  is accepted.

Generation efficiency :  $\frac{\text{Area of } \square}{\text{Area of } \square + \square}$





## 4. Event Generation (3)

### ■ Composition Method

#### Assumption:

- the probability  $f(x)$  is given by

$$f(x) = \int g(x,y)dH(y)$$

$H(y)$  : a cumulative distribution function

$g(x,y)$  : probability density of  $x$  with parameter  $y$

which can be modified as

$$f(x) = \int g(x,y)h(y)dy \quad \text{and} \quad H(y) = \int h(y)dy .$$

- Consider, for example, a particle is scattered during passing through a material

$h(\mathbf{e})$  : the scattering probability of the particle in the material depends on its energy  $\mathbf{e}$ .

$g(\mathbf{q},\mathbf{e})$  : the angular distribution depending on particle energy  $\mathbf{e}$  and the scattering angle  $\mathbf{q}$

#### Method:

- (1) Generate two uniform random numbers  $z_1$  and  $z_2$ .

- (2) Determine  $y$  by  $y = H^{-1}(z_1)$  (Direct Method)

- (3) Determine a random number  $h$  by  $G_y(x) = \int g(x,y)dx$  (Direct Method)

$$h = G_y^{-1}(z_2) ,$$



## *5. Event Generation Package SPRING*

---

**There are several possible methods**

**to generate random numbers with a required distribution**

**(1) Direct method**

**(2) Rejection method**

**(3) Composition method**

**(4) Composition-rejection method**

**(5).....**

**No method, however, is for general purpose.**

**⇒ We must find the most efficient way case by case.**

**The program package BASES/SPRING makes it possible.**





# 5. Event Generation Package SPRING (2)

## Generation algorithm of SPRING

(1) According to the algorithm of BASES, width of each small-region is adjusted so that

Each small-region has an equal contribution to the estimate of the integral

■ Generation method

- a) Integration by BASES
- b) Sample a small-region
- c) Sample a point in the small-region by the rejection method

However, this can be true only when

$$|f(x)| / p(x) = \text{const.}$$

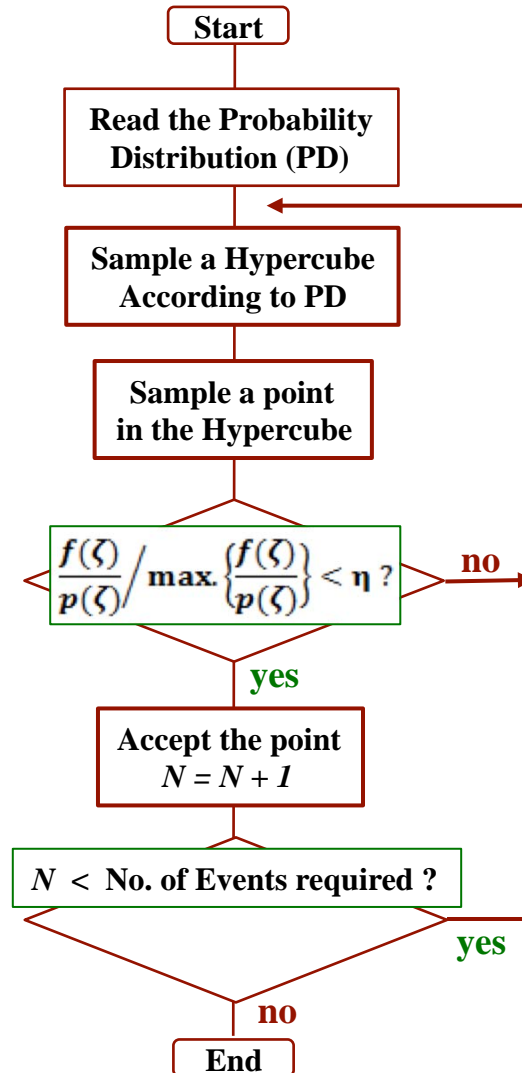
is exactly kept.

(2) In practice,

- a) Sample a sub-region according to its contribution to the estimate of integral
- b) Sample a point  $\mathbf{z}$  in a small-region in the sub-region and if  $\frac{f(\zeta)}{p(\zeta)} / \max. \left\{ \frac{f(\zeta)}{p(\zeta)} \right\} < \eta$  : uniform random number

then take the point as a generating point.

## Flow of Event Generation in SPRING





## 6. Advantage of BASES/SPRING system

### ■ BASES

- (1) Even a singular function can be integrated.
- (2) Easy to demand a geometrical boundary (cuts) of a detector in the estimate of integral.
- (3) Up to 50 dimensional integration is possible.

### ■ SPRING

- (1) Once integration has been done, it is easy to generate events (e.g. four vectors) with weight one.
- (2) By changing the initial seed of random number, a different set of events can be generated.
- (3) It is easy to demand a geometrical cuts of a detector for generating events without loss of CPU time.
- (4) By giving a differential cross section as the integrand, it is easy to estimate the expected number of observed events.

$\sigma$  : the cross section

$N_0$  : the number of generating events

$N_c$  : the number of those events, which pass through cuts of many selection criteria for a physics analysis

$L$  : the luminosity ( colliding beam) or the flux of beam ( fixed target)

The expected number of observed events

$$N_{expect} = \sigma L \frac{N_c}{N_0}$$



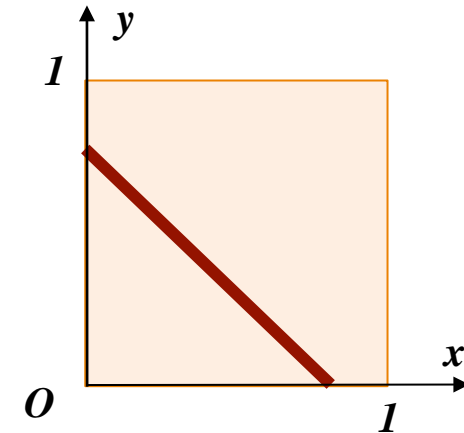
## 7. Vulnerability of BASES

To expose a vulnerability of BASES,

Calculate the following integral by BASES

$$I = \int_0^1 dx \int_0^1 dy \frac{1}{(x + y - a)^2 + \epsilon^2}$$

where  $a = 0.8$



Integration results of BASES

e	Expected	Estimate(error)	CPU time
$10^{-1}$	22.0287	22.0265(0.0196)	0.58
$10^{-2}$	248.1501	248.6445(0.5460)	19.16
$10^{-3}$	2,510.0960	2,508.27 (11.12)	117.48
$10^{-4}$	25,129.5631	24,617.0 (182.1)	585.24
$10^{-5}$	251,324.2342	239,900.9 (4,542.0)	1169.83



## 7. Vulnerability in BASES (2)

To reduce the vulnerability of BASES,  
change the integration variables from  $x, y$  to  $X, Y$

$$\begin{aligned} I &= \int_0^1 dx \int_0^1 dy \frac{1}{(x+y-a)^2 + \epsilon^2} \\ &= \int_{-1}^1 dX \int_0^2 dY \frac{0.5}{(Y-a)^2 + \epsilon^2} \end{aligned}$$

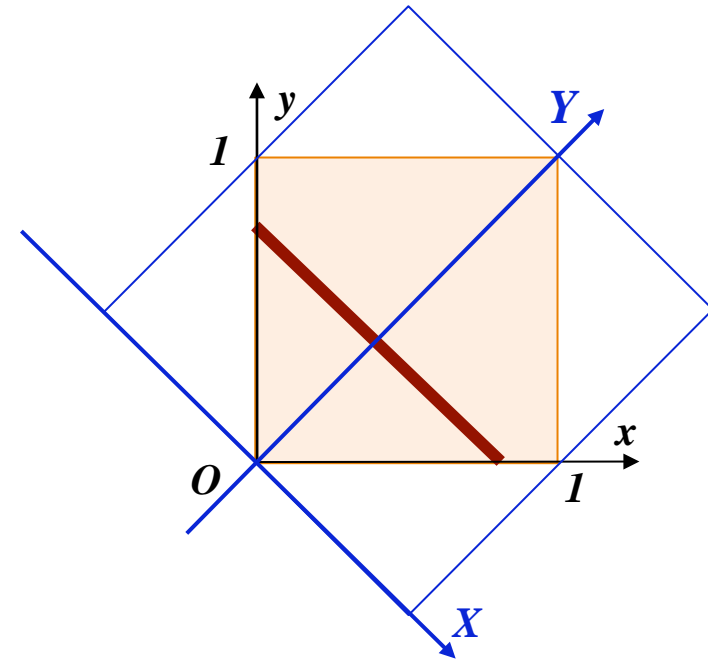
where only the shaded area

$$0 \leq x \leq 1 \quad \text{and} \quad 0 \leq y \leq 1$$

is taken into account.

### Integration results

e	Expected	Estimate(error)	CPU time
$10^{-1}$	22.0287	22.0270(0.0215)	1.70
$10^{-2}$	248.1501	248.042(0.1232)	9.47
$10^{-3}$	2,510.0960	2,511.476 (1.237 )	12.30
$10^{-4}$	25,129.5631	25,128.54 (12.15 )	17.26
$10^{-5}$	251,324.2342	251,475.9 (124.5 )	92.10





## 8. A Question from User

### ■ A question

Dear Setsuya

I am writing to you concerning your program package bases/spring which I am heavily using (and referring) in my Monte Carlo event generator programs CASCADE and RAPGAP (<http://www.desy.de/~xxxx>). I am very happy with the performance of BASES/Spring, which I use in the FORTRAN version.

Now, we have been trying to perform precision fits using the Monte Carlo event generator, and I observed some strange behaviour, which I cannot really interpret, and I just wanted to ask, if this behavior is known and how one can understand it....

We are using CASCADE to calculate a quantity F2 in deep inelastic scattering. To perform precision predictions we have been running the generator with  $10 \times 10^6$  events, to reduce the statistical error in certain bins.

Now to check the consistency, we were also running the generator with a different random number generator (here RANLUX) and with changing the seed.

We obtain always consistent results from the integration step, but when analysing the generated event, we find, that in quite a number of bins, the results from runs with 2 different random number seed do not agree within the statistical errors, they are different by up to 5 sigma.

I was wondering, whether this is something, which other people also have observed, and if yes, whether it is correct to calculate the error just as a  $\sqrt{N_{\text{event}}}$  ?

I would be very grateful to hear any comment on this

Thanks so much  
XXXX

2009/9/3

### ■ Reply to the question

Dear XXXXX

Thank you for email.

First of all the answer for your question is "yes".

It may happen when the accuracy of numerical integration is not enough.

If you want to generate  $10^6$  events, the accuracy of numerical integration should be less than  $10^{-4}$  percent.

If your probability distribution function has a quit singular behavior, the numerical integration should be done for an enough number of iterations.

(1) To see if the event generation were carried out correctly, you can check the number of miss-generations in the first page of SPRING output.

(2) If this is not the case. Please try more number of iterations in the numerical integration.

(3) This is a little bit radical way. You can check by modifying the program code as follows;

In subroutine SPRGEN around 55th line there is the code

```
FMAX = DXP(IC) .
```

After this line insert one line of code

```
FMAX = 1.5*FMAX .
```

By this change the generation efficiency will come down about 50 percent and the number of miss-generation will increase, but the generation might work safer and you may get more reasonable event set.

Best regards,

Setsuya



## 8. A Question from User (2)

### ■ A query to my reply came:

Dear Setsuya

thanks so much for your mail and your quick response.

I am a bit confused about the precision you recommend, is it really  $10^{(-4)}$  percent ?

And, is this precision needed in the integration step or in the grid optimization step ?

I attach the bases/spring output of an example run, with the setting I had. It seems the number of miss-generations is pretty small.

I would be very grateful for any further comment and advice

Thanks very much

best regards

XXXX

```

Date: 2009/ 8/24 09:00
*****
*
*     BBBBBBB   AAAA   SSSSSS  EEEEE  SSSSS  *
*      BB  BB  AA  AA  SS  SS  EE  SS  SS  *
*      BB  BB  AA  AA  SS  SS  EE  SS  *
*     BBBBBBB  AAAAAAA  SSSSSS  EEEEE  SSSSS  *
*      BB  BB  AA  AA  SS  SS  EE  SS  *
*      BB  BB  AA  AA  SS  SS  EE  SS  *
*      BBBB BB  AA  AA  SSSSSS  EEEEE  SSSSS  *
*
*      BASES Version 5.1
*      coded by S.Kawabata KEK, March 1994
*
*****

```

<< Parameters for BASES >>

```

(1) Dimensions of integration etc.
# of dimensions : Ndim =      8  ( 50 at max.)
# of Wilds      : Nwild =      8  ( 15 at max.)
# of sample points : Ncall = 45927(real) 50000(given)
# of subregions  : Ng =      48 / variable
# of regions     : Nregion =    3 / variable
# of Hypercubes  : Ncube =    6561

```

(2) About the integration variables

i	XL(i)	XU(i)	IG(i)	Wild
1	1.000000E-12	1.000000E+00	1	yes
2	1.000000E-12	1.000000E+00	1	yes
3	1.000000E-12	1.000000E+00	1	yes
4	1.000000E-12	1.000000E+00	1	yes
5	1.000000E-12	1.000000E+00	1	yes
6	1.000000E-12	1.000000E+00	1	yes
7	1.000000E-12	1.000000E+00	1	yes
8	1.000000E-12	1.000000E+00	1	yes

(3) Parameters for the grid optimization step

```

Max.# of iterations: ITMX1 =      30
Expected accuracy  : Acc1 =    1.0000 %

```

(4) Parameters for the integration step

```

Max.# of iterations: ITMX2 =     100
Expected accuracy  : Acc2 =    0.5000 %

```

Date: 2009/ 8/24 09:00

```

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 36 0.00 5.508E+02 4.835 5.507601(+0.266293)E 02 4.835 0: 3:33.00
2 69 0.00 5.797E+02 0.904 5.786128(+0.051402)E 02 0.888 0: 7:56.00
-----

```

Date: 2009/ 8/24 09:00

```

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 81 0.00 5.768E+02 0.608 5.767997(+0.035090)E 02 0.608 0:12:39.00
2 81 0.00 5.738E+02 0.621 5.753377(+0.025001)E 02 0.435 0:17:23.00
-----

```

\*\*\*\*\* END OF BASES \*\*\*\*\*

<< Computing Time Information >>

```

(1) For BASES          H: M: Sec
Overhead              : 0: 8:43.00
Grid Optim. Step     : 0: 7:56.00
Integration Step     : 0: 9:27.00
Go time for all      : 0:26: 6.00

```

(2) Expected event generation time

```

Expected time for 1000 events :      8.82 Sec

```





## 8. A Question from User (4)

### ■ Reply to XXXX

Dear XXXX

Yes, Such a high precision is required in the integration step to generate  $10^6$  event.

The reason is as follows:

The event generator SPRING uses an N dimensional histogram with variable width of the distribution function ( integrand ), which is prepared in the integration step, where N is the number of independent variables.

To generate an event

(1) SPRING samples a point x in the phase volume using a set of random numbers according to the N dimensional distribution function.

Because of this a high precision of the estimate is required.

(2) calculate the numerical value of the distribution function  $f(x)$

at this point x.

(3) If the value  $f(x)/F_{\max}(i)$  is smaller than a random number, SPRING accepts this point as an event. Otherwise go back to step (1) for the next trial.

$F_{\max}(i)$  is the maximum value of the distribution function in those N dimensional histogram bin i which includes the selected phase point x.

The values of  $F_{\max}(i)$  for all histogram bins are kept in the integration step.

If this value is too small, most of sampled phase points will be accepted, but the generated events do not reproduce the behavior of the distribution function. The number of miss-generation is almost zero. If this value is too large, the generation efficiency will be very low

and the number of miss-generation will be large.

The accuracy of the grid optimization step is not so important but I recommend to run this step until the accuracy of each iteration becomes stable. I think 2 iterations for this step is too small. You cannot only see if the grids were optimized but also cannot get an enough grid optimization. This may cause a large number of miss-generation and a low generation efficiency.

I recommend at least 5 iterations.

Best regards,  
Setsuya

### ■ Reply from XXXX

Dear Setsuya

thanks very much for your mail and explanations.  
We will follow your advice and will check the outcome.

Thanks very much again

Best regards  
XXXX