How to Write Source Routine

H. Hirayama and Y. Namito
KEK, High Energy Accelerator Research Organization

Source Routine

- Source routines determines an energy, a position and direction of a source particle.
- If these quantities are different for each particle, source routines must be placed before CALL SHOWER inside "Shower call loop".
  - ucrz_nai.f and uccg_nai.f are examples using fixed energy and direction.
  - Source energy and direction are sampled at each history in the case of ucxyz_pantom.f and uccg_phantom.f

Determination of energy of γ-ray from ^{192}Ir

- ^{192}Ir emits following γ-rays and their CDF, *F(Ei)* can be calculated from their emission rates.

i	Energy(MeV)	Emission rate(%)	$F(E_i)$
1	0.296	28.7	0.0999
2	0.308	30.0	0.2832
3	0.317	82.8	0.6826
4	0.468	47.8	0.9132
5	0.589	4.5	0.9252
6	0.604	8.2	0.9744
7	0.612	5.3	1.00

- γ-ray energy is a ***discrete random variable***.
- E_i can be determined by using random number, η, between 0 and 1.

$$F(E_{i-1}) = \sum_{j=1}^{i-1} p_j \le \eta < F(E_i) = \sum_{j=1}^{i} p_j$$

In ucrz_nai.f, uccg_nai.f, ucxyz_phantom and uccg_phantom, isamp=1, which is read from unit 4, is prepared to this discrete type sampling.

To use this option, prepare data of ebin and epdf after setting isamp=1.

E_i for ebin(i)

$f(E_i)$ for epdf(i)

Lists of a sampling routine

```
call RANDOMSET(e0)
if (e0.lt.0.0999)  ekin=0.296
elseif (e0.lt.0.2832)  ekin=0.308
elseif (e0.lt.0.6826) ekin=0.312
elseif (e0.lt.0.9132) ekin=0.468
elseif (e0.lt.0.9252) ekin=0.589
elseif (e0.lt.0.9744) ekin=0.604
else ekin=0.612;
end if
etot = ekin + iabs(iqin)*RM
```

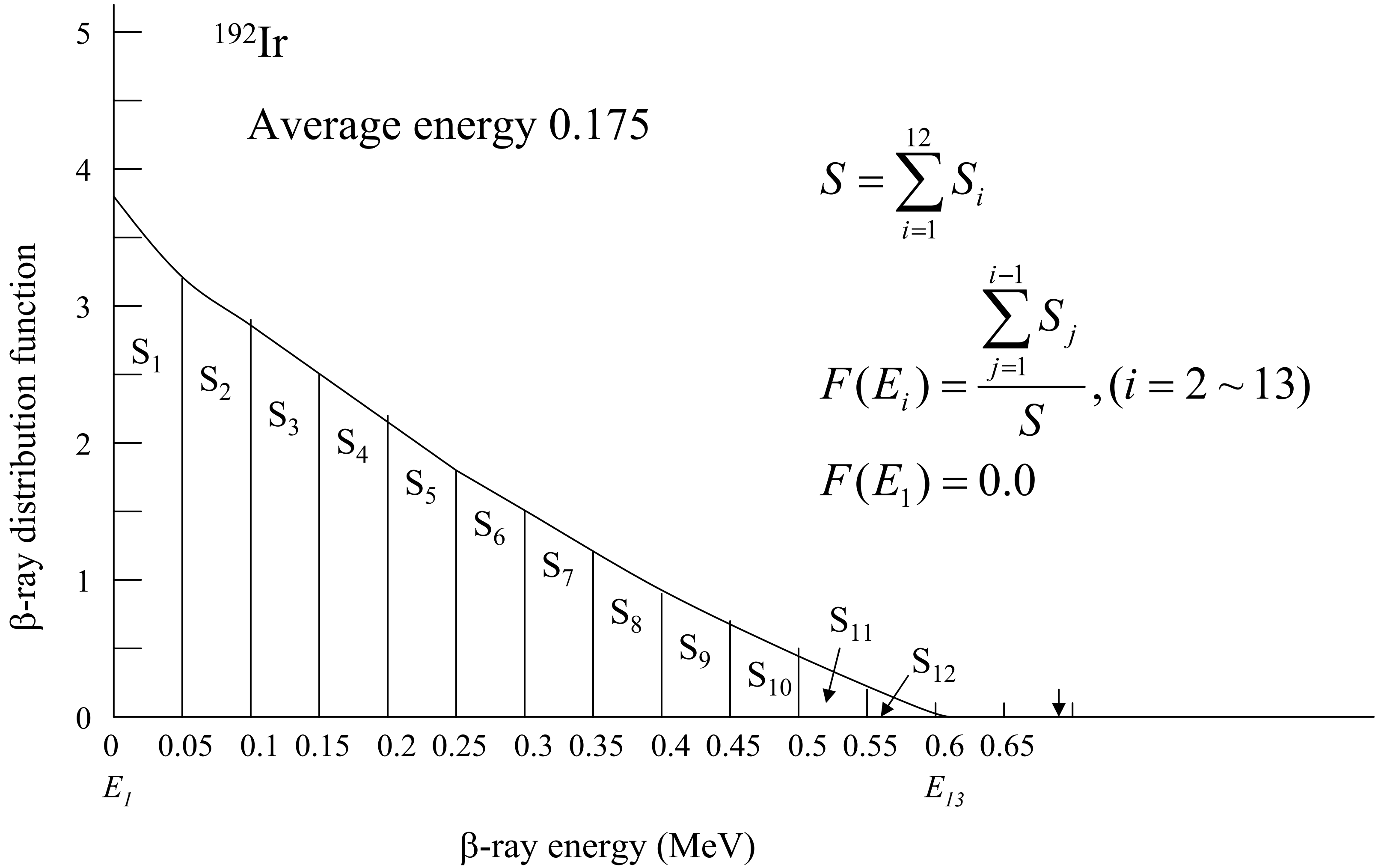
Input data for Getrz, getxyz, getcg

0.612, 0.0, 1	ekein, iqin, isamp
0.296, 28.7	ebin(1), epdf(1)
0.308, 30.0	ebin(2), epdf(2)
0.317, 82.8	ebin(3), epdf(3)
0.468, 47.8	ebin(4), epdf(4)
0.589, 4.5	ebin(5), epdf(5)
0.604, 8.2	ebin(6), epdf(6)
0.612, 5.3	ebin(7), epdf(7)
0.0, 0.0	end of data

Determination of β-ray from ^{192}Ir

- One usually decides upon a set of energy ranges with lower bounds $E_1 < E_2 < \cdots < E_n$, with the storage of the physical quantities for each these range, E_n being highest energy permitted to particles in the problem.
- If the corresponding CDFs are $F(E_1)$, $F(E_2)$, ..., $F(E_n)$, E is determined by the following procedure:
  - Determine i such that $F(E_i) < \eta < F(E_{i+1})$
  - Apply a linear interpolation between E_i and E_{i+1} to calculate E from η

$$E = E_i + \frac{(\eta - F(E_i)) \times (E_{i+1} - E_i)}{F(E_{i+1}) - F(E_i)}$$



^{192}Ir
Average energy 0.175
$$S = \sum_{i=1}^{12} S_i$$
$$F(E_i) = \frac{\sum_{j=1}^{i-1} S_j}{S}, (i = 2 \sim 13)$$
$$F(E_1) = 0.0$$
β-ray distribution function
0
1
2
3
4
5
S_1 S_2 S_3 S_4 S_5 S_6 S_7 S_8 S_9 S_{10} S_{11} S_{12}
0 0.05 0.1 0.15 0.2 0.25 0.3 0.35 0.4 0.45 0.5 0.55 0.6 0.65
E_1
E_{13}
β-ray energy (MeV)

•From a energy distribution of β-rays of ^{192}Ir, E_i and $F(E_i)$ can be calculated.

i	$E_i(MeV)$	$F(E_i)$	i	$E_i(MeV)$	$F(E_i)$
1	0.00	0.000	8	0.35	0.887
2	0.05	0.188	9	0.40	0.936
3	0.10	0.355	10	0.45	0.968
4	0.15	0.503	11	0.50	0.987
5	0.20	0.630	12	0.55	0.997
6	0.25	0.736	13	0.60	1.000
7	0.30	0.820			

Lists of a sampling routine

```
      call RANDOMSET(RN14)
       do ie=2,nemax
         if (RN14.le.cdf(ie))  go to 1000
        end do
1000  ekin=es(ie-1)+(RN14-cdf(ie-1))*(es(ie)-es(ie-1))/(cdf(ie)-cdf(ie-1))
         etot = ekin + iabs(iqin)*RM
```

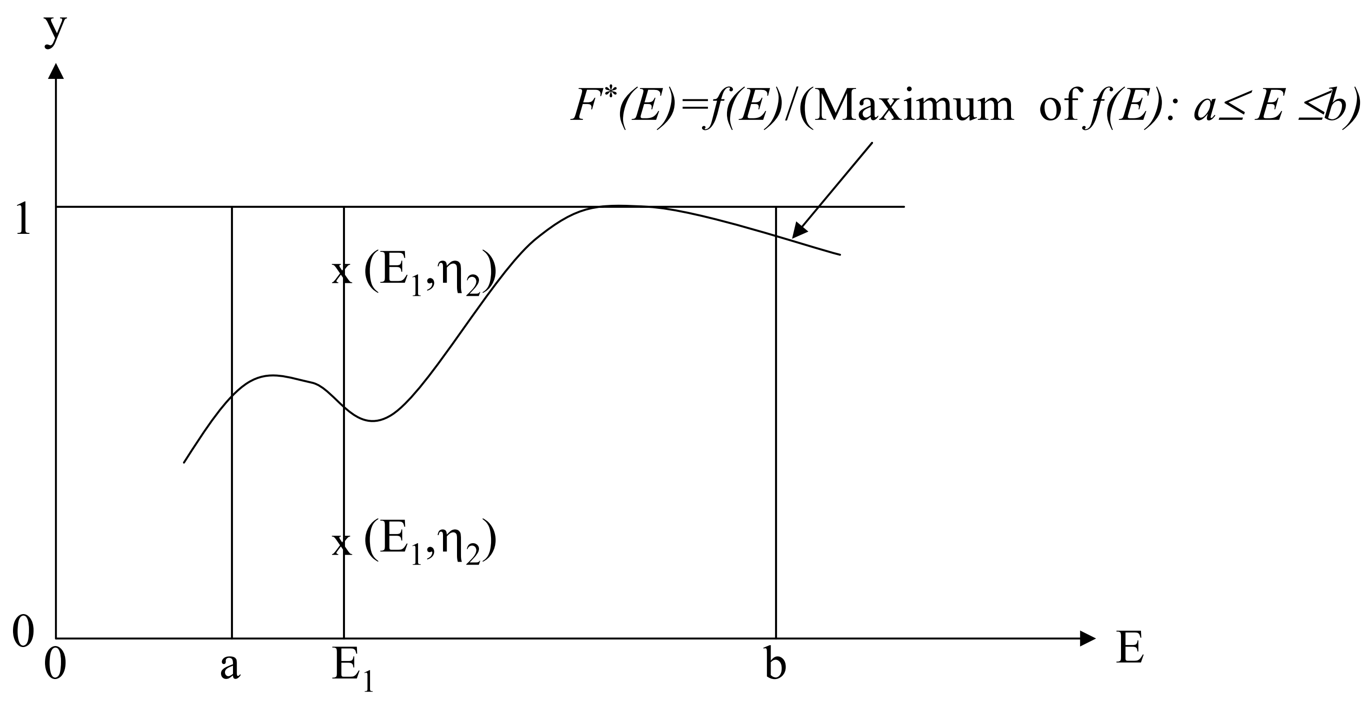
- Data of es(i) and cdf(i) (i=1,nemax) must be defined before this routine.

Input data for getrz, getxyz and getcg

0.600, 0.0, 2	ekein, iqin, isamp
0.0	ebinmin
0.05, 0.188	ebin(1), epdf(1)
0.10, 0.167	ebin(2), epdf(2)
0.15, 0.148	ebin(3), epdf(3)
0.20, 0.127	ebin(4), epdf(4)
0.25, 0.106	ebin(5), epdf(5)
0.30, 0.084	ebin(6), epdf(6)
0.35, 0.067	ebin(7), epdf(7)
0.40, 0.049	ebin(8), epdf(8)
0.45, 0.032	ebin(9), epdf(9)
0.50, 0.019	ebin(10), epdf(10)
0.55, 0.010	ebin(11), epdf(11)
0.60, 0.003	ebin(12), epdf(12)
0.0, 0.0	end of data

Rejection method: Von Neumann's method

- When a spectrum is given as equation, *f(E)*, but its integration is difficult, von Neumann's method is useful to determine *E*.



•Sample E1 from uniform distribution between a and b:

$$E_1 = a + \eta_1 (b - a); \eta_1 = \int_a^{E_1} d\xi / (b - a) = (E_1 - a)/(b - a).$$

•Calculate y for E_1, $y = f^*(E_1)$.

•Pick next random number, η_2. Terminate the algorithm and accept the value of E_1 if

$$\eta_2 < y.$$

•Otherwise, sample again.

Determination of position: Line source

- Suppose a source which is uniformly distributed between $a \leq x < b$.
- In this case, the PDF is

$$f(x)dx = dx/(b-a); \int_a^b f(\xi)d\xi$$

- By solving a following equation,

$$\eta = F(x) = \int_a^x f(\xi)d\xi = (x-a)/(b-a)$$

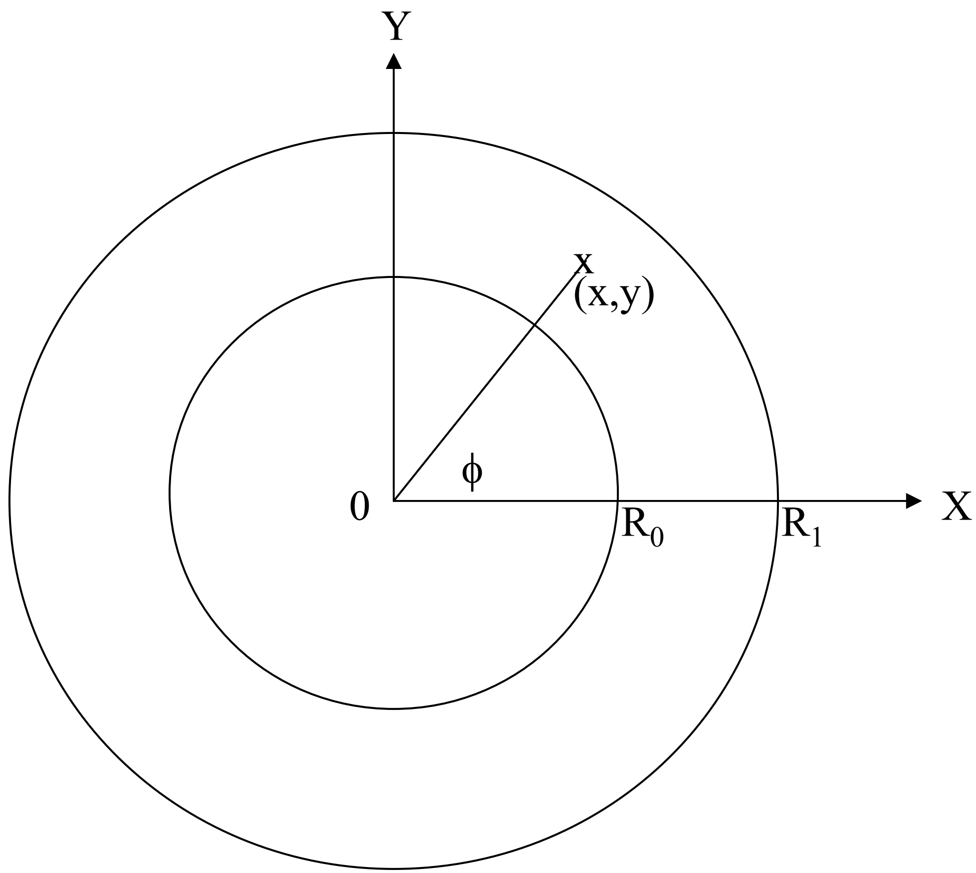
a position x is determined as $x=a+\eta(b-a)$.

Lists of sampling routine

```
    call RANDOMSET(RN0)
    x1=xmin+RN0*(xmax-xmin)
!   xmin and xmax are a and b, respectively.
```

Uniform source on an annulus of radii $R_0<R<R_1$

- Suppose a source which is uniformly distributed inside annual area between a radius R_0 and R_1 in the X-Y plane.



•In this case, the PDF for radius is

$$f(r)dr = 2\pi dr / \pi(R_1^2 - R_0^2) = 2rdr/(R_1^2 - R_0^2); \int_{R_0}^{R_1} f(\xi)d\xi = 1.$$

•The radial position is obtained by solving

$$\eta_1 = F(r) = \int_{R_0}^{r} f(\xi)d\xi = (r^2 - R_0^2)/(R_1^2 - R_0^2)$$

$$r = \sqrt{R_0^2 + \eta_1(R_1^2 - R_0^2)}$$

•x and y is determined from

$$x = r\cos(\phi)$$

$$y = r\sin(\phi)$$

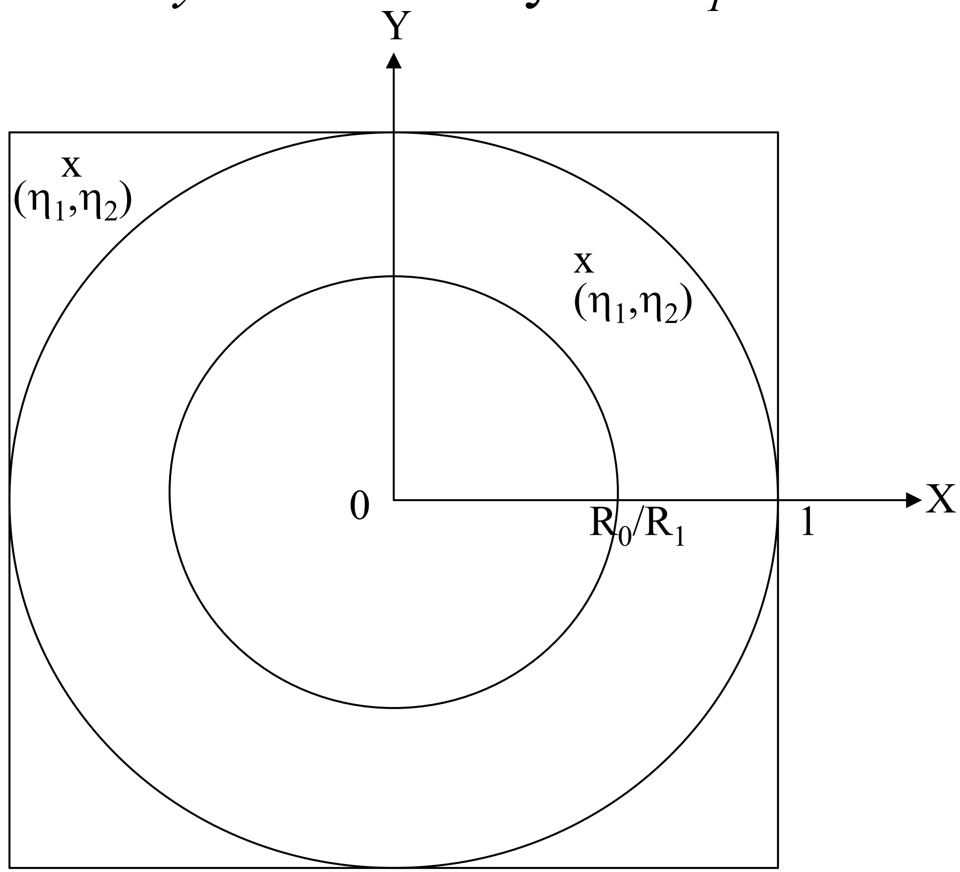
Lists of a sampling routine

```
      call RANDOMSET(RN1)
      r02=r0*r0
      r12=r1*r1;
      rr=sqrt(r02+RN1*(r12-r02))
      call RANDOMSET(RN2)
      phai=PI*(2.0*RN2-1.0)
!     include  'include/egs5_misc.f‘ must be included to use PI.
      xin=rr*cos(phai)
      yin=rr*sin(PHAI)
```

- A position (x,y) can be determined more easily using "rejection" method.

- A point is chosen randomly within the square $-1 \leq x \leq 1$; $-1 \leq y \leq 1$.

- If this point lies within an annulus of $R_0/R_1 < R < 1$ the point is accepted and x and y are scaled by the R_1.



List of sampling routine

```
1000   call RANDOMSET(RN3)
       call RANDOMSET(RN4)
       xi0=2.0*RN3-1.0
       yi0=2.0*RN4-1.0
       rr=sqrt(xi0*xi0+yi0*yi0)
       if (rr.gt.1.0.or.rr.lt.r0/r1)  go to 1000
       xin =r1*xi0
       yin=r1*yi0
```

Direction Co-ordinate for Point Isotropic Source

- Rejection method can be applied to this case more efficiently.
- A point (x_i,y_i,z_i) is chosen randomly within the box
  $-1 \le x \le 1$; $-1 \le y \le 1$; $-1 \le z \le 1$.
- If this point within a sphere with unit radius

$$R = \sqrt{x_1^2 + y_1^2 + z_1^2} \le 1,$$

  u, v, w are determined by

$$u = x_1 / R; v = y_1 / R; w = z_1 / R.$$

- Otherwise sample a point again.

サンプリングルーチンのリスト

```
1000    call RANDOMSET(RN9)
        call RANDOMSET(RN10)
        call RANDOMSET(RN11)
        x0=2.0*RN9-1.0
        y0=2.0*RN10-1.0
        z0=2.0*RN11-1.0
        rr=sqrt(x0*x0+y0*y0+z0*z0)
        if (rr.gt.1.0) go to 1000
        uin = x0/rr
        vin = y0/rr
        win = z0/rr
```

Isotropic source used in ucrz_nai.f, uccg_nai.f etc.

1. Set uin=vin=win=0.0 for directional cosine data read from unit 4.

2. At main program, isot is set to 1 by this input data.

```
! -----------------------------------------
! Set isotropic source flag if uin=vin=win=0
! -----------------------------------------
    isot=0              ! monodirectional
    if (uin+vin+win.eq.0.0) isot=1
```

3. In these user code, treat a symmetric geometry for Z-axis. Therefore, win only sampled and uin is treated always as 0.0.

Considering forward direction only, direction cosines are determined as follows.

```
    if (isot.eq.1.0) then        ! Sample isotropically (forward only).
      call randomset(rnnow)
      win = 1.D0 - rnnow
      vin = sqrt(1.D0 - win*win)
     end if
```