# 線源ルーチンの書き方

平山　英夫、波戸　芳仁
KEK, 高エネルギー加速器研究機構

# 線源ルーチン

- 線源ルーチン——粒子のパラメータを決めるルーチン
  - 粒子の種類
  - エネルギー
  - 位置
  - 方向
- もし、これらのパラメータが、ヒストリー毎に異なる場合には、線源ルーチンを、“Shower call loop”内のCALL SHOWER の前に書いておく必要がある。
  - ucnaicvg.f では、ヒストリー毎に変わらないモードと、方向を等方線源とするモードがある
  - ucphantomcgv.fで は、ヒストリー毎にエネルギー（100kVのX線）と方向を決定

# $^{192}$Irからのγ線(実習課題1)

- $^{192}$Ir は、以下のγ線を放出する。それぞれの累積分布関数(CDF), $F(Ei)$, は、放出率から求める。

| $i$ | Energy(MeV) | Emission rate(%) | $F(E_i)$ |
|---|---|---|---|
| 1 | 0.296 | 28.7 | 0.1385 |
| 2 | 0.308 | 30.0 | 0.2833 |
| 3 | 0.317 | 82.7 | 0.6824 |
| 4 | 0.468 | 47.8 | 0.9131 |
| 5 | 0.589 | 4.5 | 0.9348 |
| 6 | 0.604 | 8.2 | 0.9744 |
| 7 | 0.612 | 5.3 | 1.00 |

- この場合のγ-線のエネルギーは、**離散的確率変数**
- $E_i$ は、0と1の間の乱数 ηを使って決定する。

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

If, elseif, else 文を使用してサンプリングする方法と、data文や、入力データで、エネルギーと対応する累積分布関数を定義し、それを使ってサンプリングする方法がある。

$\gamma$ 線のエネルギーの数が多い場合には、後者の方が良い。

# 直接サンプリングルーチンのリスト

```
call randomset(rnnow)
 if (rnnow.lt.0.1385)  then
   ekin=0.296
 elseif (rnnow.lt.0.2833) then
   ekin=0.308
 elseif (rnnow.lt.0.6824) then
    ekin=0.317
 elseif (rnnow.lt.0.9131) then
    ekin=0.468
 elseif (rnnow.lt.0.9348) then
    ekin=0.589
 elseif (rnnow.lt.0.9744) then
    ekin=0.604
 else ekin=0.612;
 end if
```

# data 文でエネルギー及び累積分布関数を定義する場合

```
 real*8  esbin(7),escdf(7)
  data esbin/0.296,0.308,0.317,0.468,0.589,0.604,0.612/
  data escdf/0.1385,0.2833,0.6824, 0.9131, 0.9348, 0.9744,1.0/

  call randomset(rnnow)
  do ie=1,7
     if(rnnow.le.escdf(ie)) go to 1000
  end do
1000   ekin=esbin(ie)
```

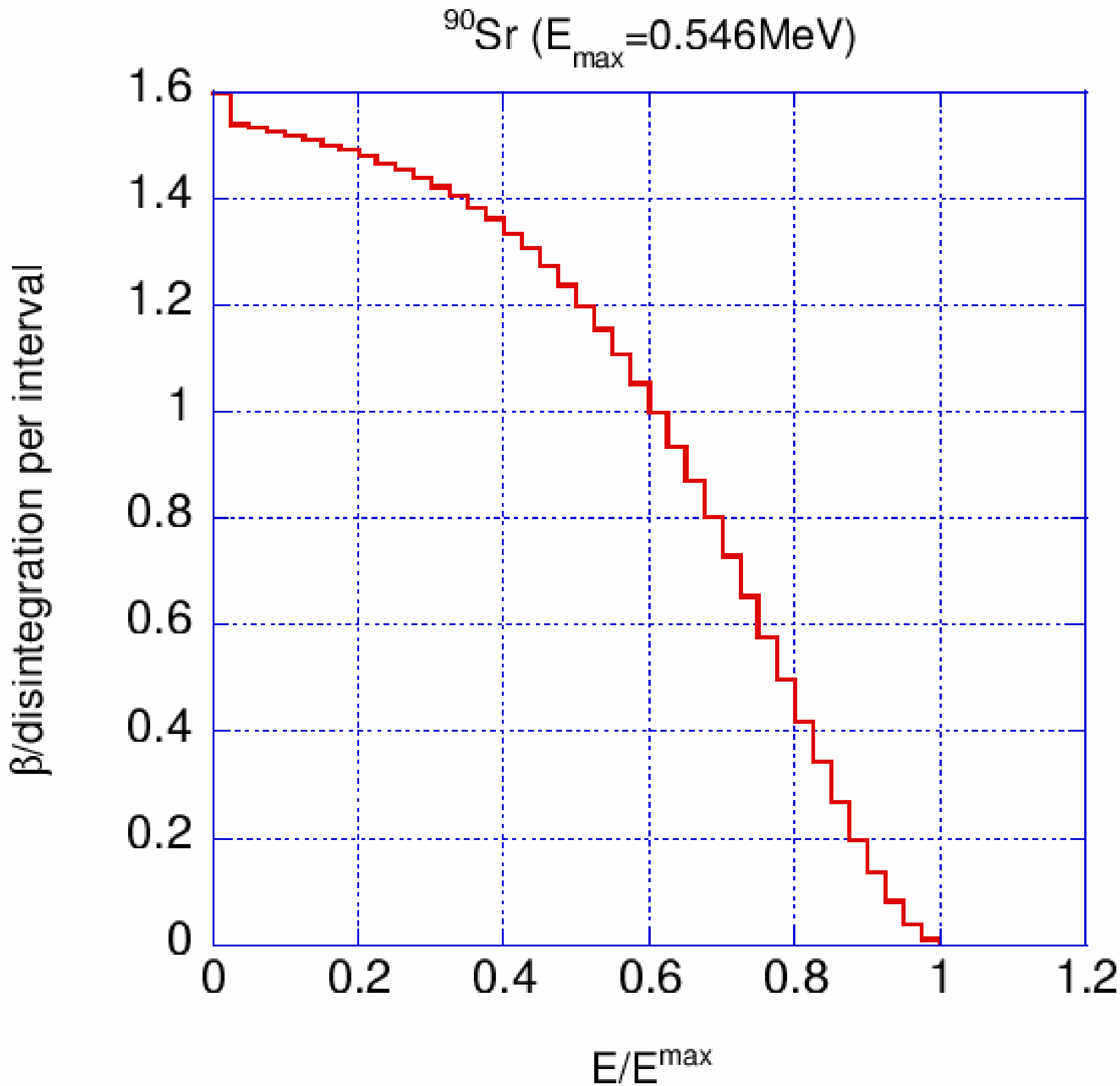
# $\beta$ 線(実習課題2)

- 連続型の過程のサンプリングでは、一般には直接サンプリングは難しい。
- 近似的な方法であるが、スペクトルの形が与えられている場合にどの様な場合にも適用できる方法は、横軸(この場合は、エネルギー)を等間隔に区分し、その区間の積分値の全領域の積分値に対する割合を確率密度関数とする方法である。
- 積分が困難な場合には、区間内の変化が直線であると仮定して台形公式を使用する。
- この場合、精度を上げるには、分点数を多くすると共に、対応する値を理論値等からできるだけ精度良く求める必要がある。

# $\beta$ 線(実習課題2)

- 等間隔に区分して確率分布関数を作成する例
- ICRU Report 56等の（エネルギー／最大エネルギー）各区分当たりの崩壊当たりの $\beta$ 線数データを使用
- ICRU Report 56では、41区間のデータが示されている $E_1/E_n < E_2/E_n < \cdots < E_n/E_n = 1$,
  - $E_n$ は、放出される $\beta$ 線のエネルギーの最大値
- 各区間の $\beta$ 線数から、各区間に対応する確率密度関数(pdf)と累積分布関数(cdf；$F(E_i)$)を計算（$F(E_0)=0, F(E_n)=1$）し、
  - $F(E_{i-1}) < \eta < F(E_i)$となる $i$ を求める。
  - $E_{i-1}$ と $E_i$ の間で直線内挿により、$E$ を求める。

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



90Sr (Emax=0.546MeV)
β/disintegration per interval
E/Emax
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6

# $^{90}Sr$の$\beta$線放出率

| T/Tmax | 放出率 | T/Tmax | 放出率 | T/Tmax | 放出率 | T/Tmax | 放出率 |
|---|---|---|---|---|---|---|---|
| 0 | 1.597 | 0.025 | 1.538 | 0.05 | 1.532 | 0.075 | 1.526 |
| 0.1 | 1.518 | 0.125 | 1.509 | 0.15 | 1.5 | 0.175 | 1.49 |
| 0.2 | 1.479 | 0.225 | 1.466 | 0.25 | 1.453 | 0.275 | 1.439 |
| 0.3 | 1.422 | 0.325 | 1.404 | 0.35 | 1.384 | 0.375 | 1.361 |
| 0.4 | 1.335 | 0.425 | 1.306 | 0.45 | 1.274 | 0.475 | 1.238 |
| 0.5 | 1.198 | 0.525 | 1.154 | 0.55 | 1.106 | 0.575 | 1.053 |
| 0.6 | 0.997 | 0.625 | 0.935 | 0.65 | 0.87 | 0.675 | 0.801 |
| 0.7 | 0.729 | 0.725 | 0.654 | 0.75 | 0.577 | 0.775 | 0.498 |
| 0.8 | 0.42 | 0.825 | 0.343 | 0.85 | 0.268 | 0.875 | 0.198 |
| 0.9 | 0.135 | 0.925 | 0.081 | 0.95 | 0.038 | 0.975 | 0.01 |
| 1 | 0 | | | | | | |

# スペクトル情報の定義

```
      real*8  esbin(41),espdf(41),escdf(41)
      data espdf/1.597,1.538 ,1.532,1.526 ,1.518,1.509 ,1.500,1.490 ,
     *           1.479,1.466 ,1.453,1.439 ,1.422,1.404 ,1.384,1.361 ,
     *           1.335,1.306 ,1.274,1.238 ,1.198,1.154 ,1.106,1.053 ,
     *           0.997,0.935 ,0.870,0.801 ,0.729,0.654 ,0.577,0.498 ,
     *           0.420,0.343 ,0.268,0.198 ,0.135,0.081 ,0.038,0.010 ,
     *           0.000/
```

```
      emax=0.546
      deltaes=0.025
      do i=1,41
        esbin(i)=(i-1)*deltaes*emax
      end do
```

# cdfの作成

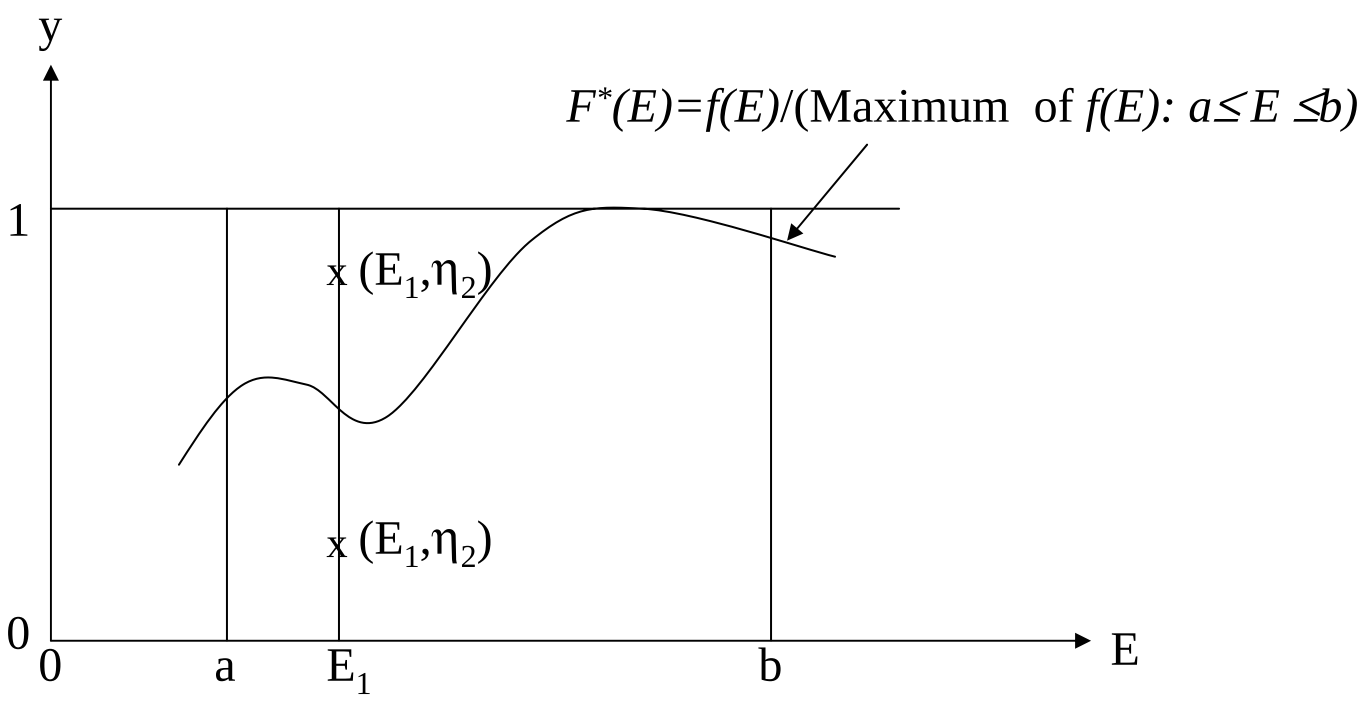
```
!---------------------------
!     Calculate CDF from pdf
!---------------------------
      tnum=0.D0
      do ie=1,41
        tnum=tnum+espdf(ie)
      end do

      do ie=1,41
        if(ie.eq.1) then
          escdf(ie)=0.0
        else
          escdf(ie)=escdf(ie-1)+espdf(ie-1)/tnum
        end if
      end do
```

# サンプリングルーチンのリスト

```
      call randomset(rnnow)
       do ie=2,41
         if (rnnow.le.escdf(ie))  go to 1000
        end do
1000    ekin=esbin(ie-1)+(rnnow-escdf(ie-1))*(esbin (ie)-esbin (ie-1))/
     * (escdf(ie)-escdf(ie-1))
         etot = ekin + iabs(iqin)*RM
```

# Rejection 法:Von Neumann's method

- スペクトル *f(E)* が式で与えられているが、その積分が難しい場合、von Neumann's method は、*E* を決定するのに便利である。



• $E_1$ を a と b の区間で一様分布としてサンプリングする:

$$f(E)dE = c \bullet dE$$

$$\int_a^b f(\xi)d\xi = c[\xi]_a^b = c(b-a) = 1$$

なので

$$f(E) = \frac{dE}{b-a}$$

$$\eta_1 = \int_a^E \frac{d\xi}{b-a} = \frac{E-a}{b-a}$$

$$E = a + \eta_1 \times (b-a)$$

- $E_1$の時の $y$ を計算する, $y=F^*(E_1)$.
- 次の乱数 $\eta_2$ を求め、以下の場合には、$E_1$ をエネルギー とする。

$$\eta_2 < y$$

- 上記の条件に当てはまらない場合は、サンプリングをやりなおす。

# 一様な線状線源の場合の位置の決定

- 線源が $a$ と $b$ の間で一様に分布しているとする。

  $a \le x < b.$

- この場合、確率分布関数(PDF)は、次のようになる。

$$f(x)dx = c\,dx;\ \int_a^b f(\xi)d\xi = c[\xi]_a^b = c(b-a) = 1$$

$$f(x)dx = \frac{dx}{b-a}$$

- 以下の式を解くことにより位置を決定することができる。

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

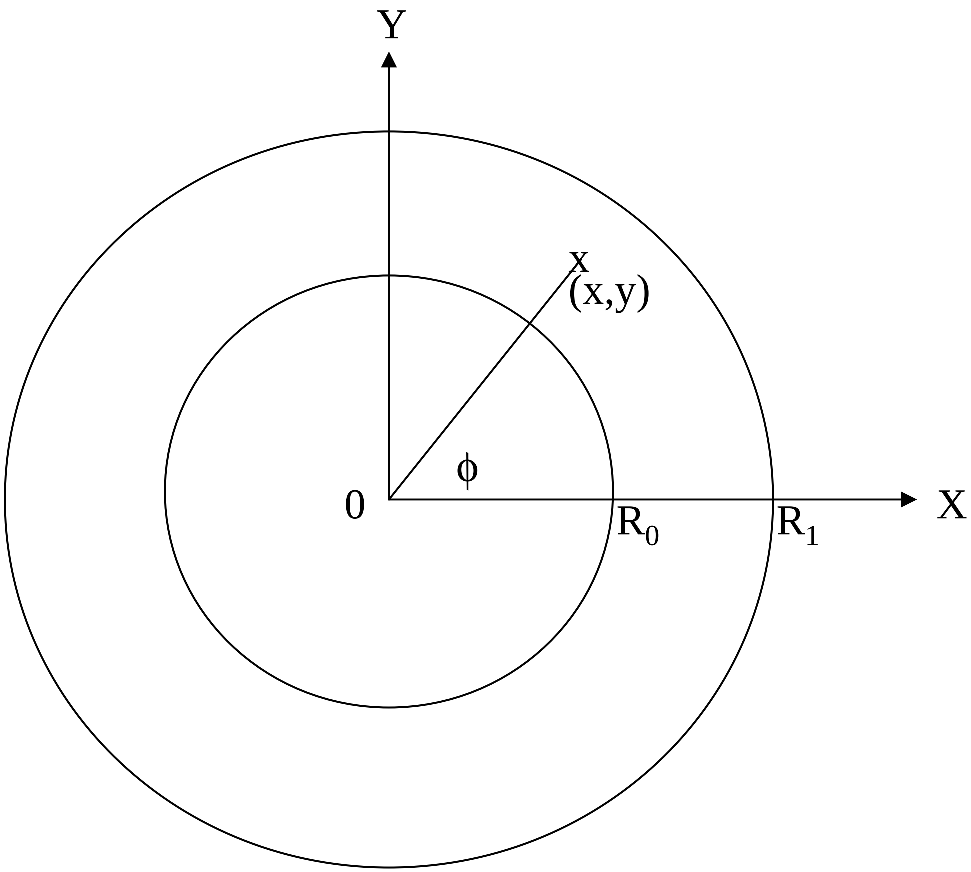
位置 $x$ は、$x=a+\eta(b-a)$ となる。

サンプリングルーチンのリスト

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

# $R_0<R<R_1$の円環に一様に分布した線源 (実習課題3)

- X-Y 平面で半径が$R_0$と$R_1$ の間の領域に一様に分布している線源を考える。



•この場合、半径に対する確率密度関数は、以下のようになる

$$f(r)dr = c \bullet 2\pi r dr$$

$$\int_{R_0}^{R_1} f(\xi)d\xi = c\pi\left[\xi^2\right]_{R_0}^{R_1} = c\pi(R_1^2 - R_0^2) = 1$$

$$c = \frac{1}{\pi(R_1^2 - R_0^2)}$$

$$f(r)dr = \frac{2rdr}{R_1^2 - R_0^2}$$

- 半径は、次式を解くことにより決める事ができる。

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

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

- $x$ と $y$ は次式から決定する。

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

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

# Φのサンプリング

Φは、0から2π(360度)まで一様であるので確率密度関数は以下の様になる

$$f(\phi)d\phi = cd\phi;\ \int_0^{2\pi} f(\xi)d\xi = c\left[\xi\right]_0^{2\pi} = c \bullet 2\pi = 1$$

$$f(\phi)d\phi = \frac{d\phi}{2\pi}$$

Φは、以下を解くことにより決定することが出来る

$$\eta = F(\phi) = \int_0^{\phi} f(\xi)d\xi = \frac{\phi}{2\pi}$$

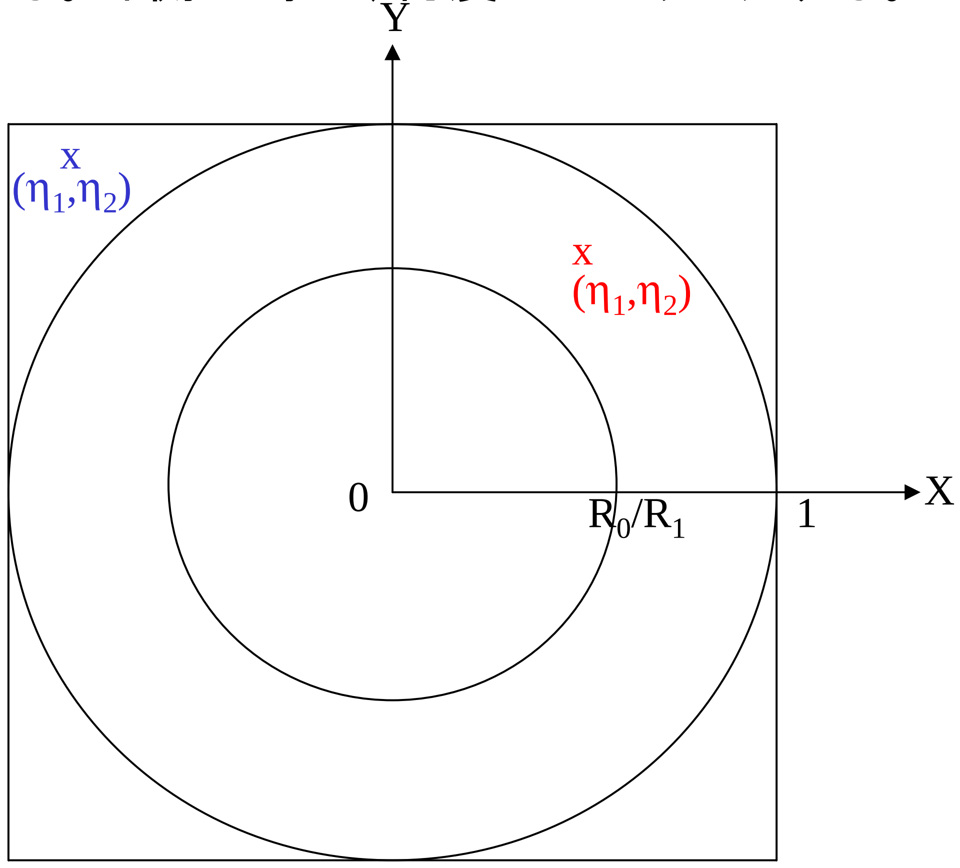
$$\phi = 2\pi\eta$$

## サンプリングルーチンのリスト

```
    call randomset(rnnow)
    rr0=sqrt(r02+rnnow*(r12-r02))
    call randomset(rnnow)
    phai=PI*(2.0*rnnow-1.0)
!   include  'include/egs5_misc.f' must be included to use PI.
   xin=rr0*cos(phai)
   yin=rr0*sin(phai)
```

**r02=r0*r0, r12=r12*r12で、do –loop の外側で定義**

- 位置 $(x,y)$ は、“rejection” method により、より簡単に決める事ができる。
- $-1 \leq x \leq 1;\ -1 \leq y \leq 1$ の区画内で、一様に分布しているとして$x$と$y$をサンプリングする。
- この点が、$R_0/R_1<R<1$ の範囲にある場合には、$x$ と $y$ を $R_1$倍する事により位置を決定する。外側の時は、再度サンプリングする。



# サンプリングルーチンのリスト

```
1100    call randomset(rnnow)
        xi0=2.0*rnnow-1.0
        call randomset(rnnow)
        yi0=2.0*rnnow-1.0
        rr0=sqrt(xi0*xi0+yi0*yi0)
        if (rr0.gt.1.0.or.rr0.lt.r0/r1)  go to 1100
        xin =r1*xi0
        yin=r1*yi0
```

# 点等方線源(実習課題4)

- Z方法の方向余弦wの確率密度関数は、

$$f(\theta)d\theta = c \bullet 2\pi \sin\theta d\theta$$

$w = \cos(\theta)$とすると

$$\frac{dw}{d\theta} = -\sin\theta \rightarrow g(w)dw = -c \bullet 2\pi dw$$

$$\int_1^{-1} g(\xi)d\xi = -c \bullet 2\pi \bullet (-2) = 1; \quad c = \frac{1}{4\pi}$$

$$g(w)dw = -\frac{1}{2}dw$$

wは、次式を解くことにより決める事ができる。

$$\eta = G(w) = \int_1^w g(\xi)d\xi = \frac{1}{2}(1-w); \quad w = 1-2\eta$$

1-2 $\eta$ と2 $\eta$ -1は等価なので、どちらを使用しても良い。

2 $\pi$ 方向の時は、以下の様になる

$$\int_1^0 g(\xi)d\xi = -c \bullet 2\pi \bullet (-1) = 1; \quad c = \frac{1}{2\pi}$$

$$g(w)dw = -dw$$

$$\eta = G(w) = \int_1^w g(\xi)d\xi = 1-w; \quad w = 1-\eta$$

1- $\eta$ は、$\eta$ と等価なので、w= $\eta$ とする。

## Φのサンプリング

Φは、0から2π(360度)まで一様であるので確率密度関数は以下の様になる

$$f(\phi)d\phi = c\,d\phi; \quad \int_0^{2\pi} f(\xi)d\xi = c\left[\xi\right]_0^{2\pi} = c \bullet 2\pi = 1$$

$$f(\phi)d\phi = \frac{d\phi}{2\pi}$$

Φは、以下を解くことにより決定することが出来る

$$\eta = F(\phi) = \int_0^{\phi} f(\xi)d\xi = \frac{\phi}{2\pi}$$

$$\phi = 2\pi\eta$$

u, vは、wとφから以下の様になる

$$u = \sqrt{1 - w^2} \bullet \cos(\phi)$$

$$v = \sqrt{1 - w^2} \bullet \sin(\phi)$$

# サンプリングルーチンのリスト(2π)

```
      call randomset(rnnow)
      wi=rnnow
      call randomset(rnnow)
      phai=PI*(2.0*rnnow-1.0)
!     include  'include/egs5_misc.f‘ must be included to use PI.
      ui=dsqrt(1.0-wi*wi)*cos(phai)
      vi=dsqrt(1.0-wi*wi)*sin(phai)
```

# 点等方線源(rejection法)

- この場合には、rejection 法が最も効率的である。
- 点 ($x_i$,$y_i$,$z_i$) が以下の直方体中に一様に分布しているとして$x_i$, $y_i$, $z_i$ をサンプリングする。
  $-1 \leq x \leq 1; -1 \leq y \leq 1; -1 \leq z \leq 1.$
- もし、この点が半径1の単位球内にある場合には、

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

u, v, w は、次式で決める事ができる。

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

- 外の場合は、位置のサンプリングからやり直す。

# サンプリングルーチンのリスト

```
1200    call randomset(rnnow)
        x0=2.0*rnnow-1.0
        call randomset(rnnow)
        y0=2.0*rnnow-1.0
        call randomset(rnnow)
        z0=2.0*rnnow-1.0
        rr0=sqrt(x0*x0+y0*y0+z0*z0)
        if (rr.gt.1.0) go to 1200
        uin = x0/rr0
        vin = y0/rr0
        win = z0/rr0
```

winが正の２π等方線源のばあいは、z0=rnnow