

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。
 - ucrz_nai.f, uccg_nai.f は、ヒストリー毎に変わらない例
 - ucxyz_phantom.f, uccg_phantom.f は、ヒストリー毎にエネルギーと方向が異なる例

^{192}Ir からの γ 線のエネルギー決定

- ^{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), $F(E_i)$, は、放出率から求める。

| 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 |

- 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。

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

ucrz_nai.f, uccg_nai.f, ucxyz_phantom, uccg_phantom で、ユニット4から読み込む線源エネルギー条件、**isamp=1**は、この様なサンプリングを行うために設けられている。

Isampとして、1を入力し、

ebin(i) として、 E_i を

epdf(i)として、 $f(E_i)$ を

入力データすれば、この例のようなガンマ線エネルギーをサンプリングすることができる。

サンプリングルーチンのリスト

```
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
```

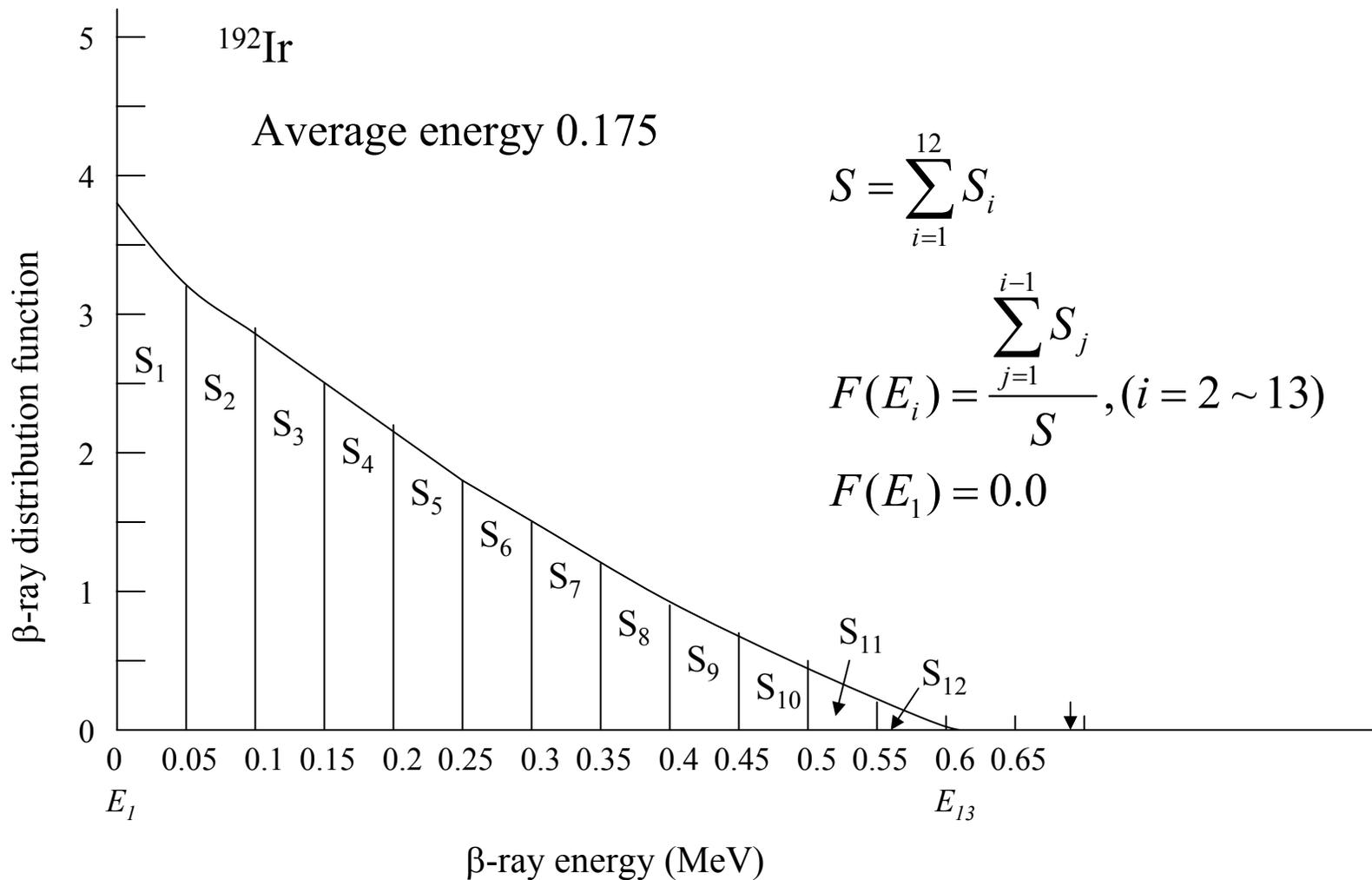
Getrz, getxyz, getcg 等の入力データとする場合

| | |
|---------------|--------------------|
| 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 |

^{192}Ir からの β 線エネルギーの決定

- β 線のスペクトルをエネルギー区間に分割する。
 - $E_1 < E_2 < \dots < E_n$,
 - E_n は、放出される β 線のエネルギーの最大値
- 各エネルギー区間に対応する累積分布関数(CDF)を、 $F(E_1), F(E_2), \dots, F(E_n)$, とすると E を、次の方法で決定する事ができる。
 - $F(E_i) < \eta < F(E_{i+1})$ となる i を求める。
 - E_i と E_{i+1} の間で直線内挿により、 E を求める。

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



- ¹⁹²Irからのβ線のエネルギースペクトルから、 E_i 及び $F(E_i)$ は次の様になる。

| i | $E_i(\text{MeV})$ | $F(E_i)$ | i | $E_i(\text{MeV})$ | $F(E_i)$ |
|-----|-------------------|----------|-----|-------------------|----------|
| 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 | | | |

サンプリングルーチンのリスト

```
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
```

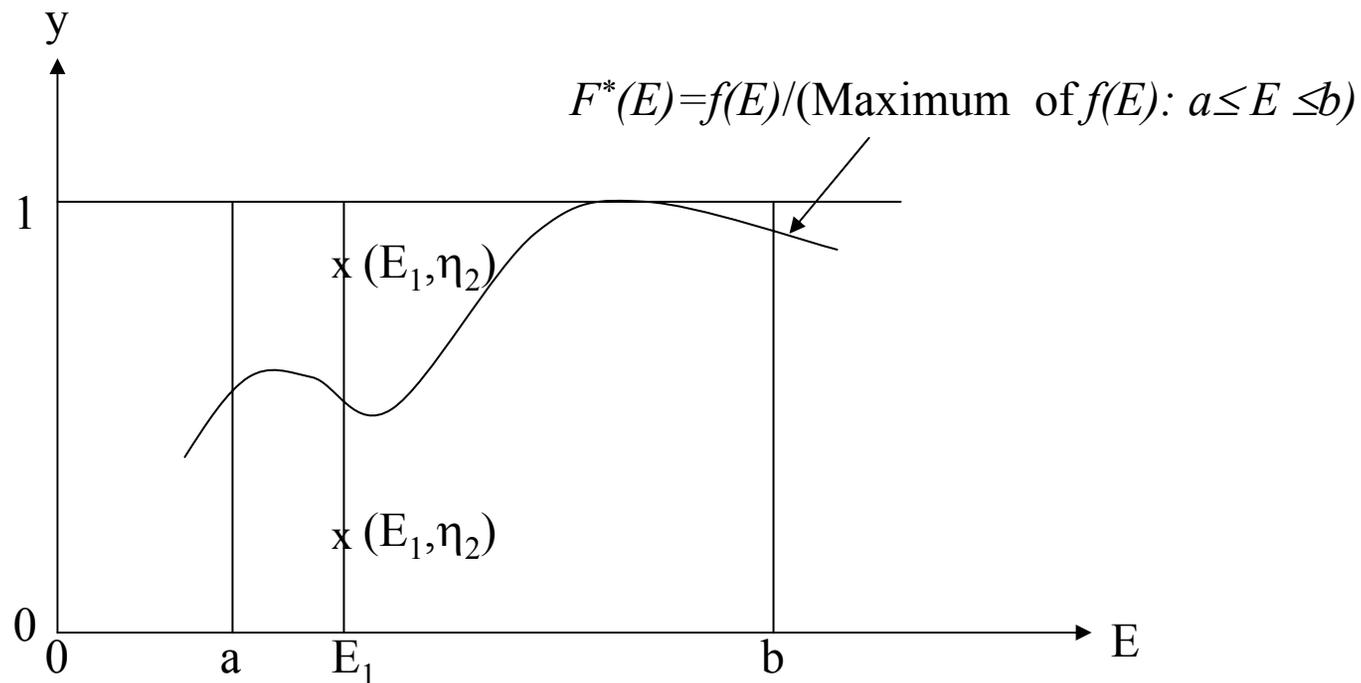
- $es(i)$ と $cdf(i)$ ($i=1,nemax$)は、事前に入力データ又は、ステートメントとし用意しておく必要がある。

Getrz, getxyz, getcg 等の入力データとする場合

| | |
|---------------|--------------------|
| 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 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:

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

- E_1 の時の y を計算する, $y = f^*(E_1)$.
- 次の乱数 η_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。

$$a \leq x < b.$$

- この場合、累積分布関数(CDF)は、次のようになる。

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

- 以下の式を解く。

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

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

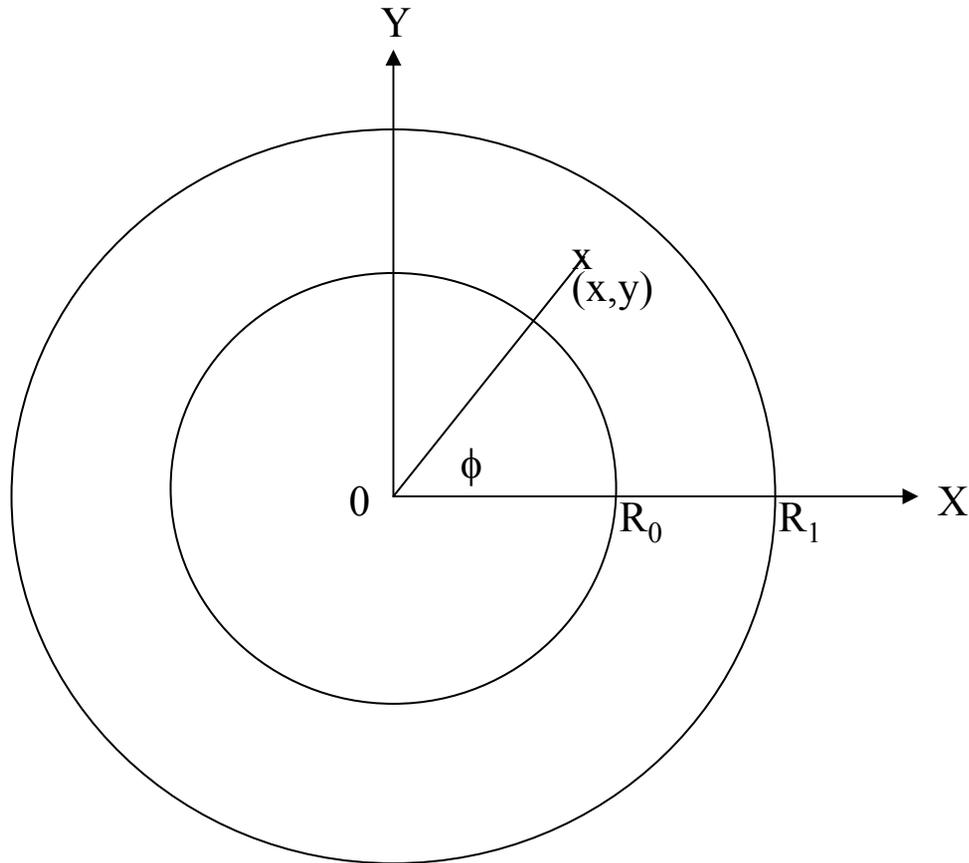
```
call RANDOMSET(RN0)
```

```
x1=xmin+RN0*(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

- 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、半径に対する累積分布関数は、

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

•半径は、次式を説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。

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

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

サンプリングルーチンのリスト

```
call RANDOMSET(RN1)
```

```
r02=r0*r0
```

```
r12=r1*r1;
```

```
rr=sqrt(r02+RN1*(r12-r02))
```

```
call RANDOMSET(RN2)
```

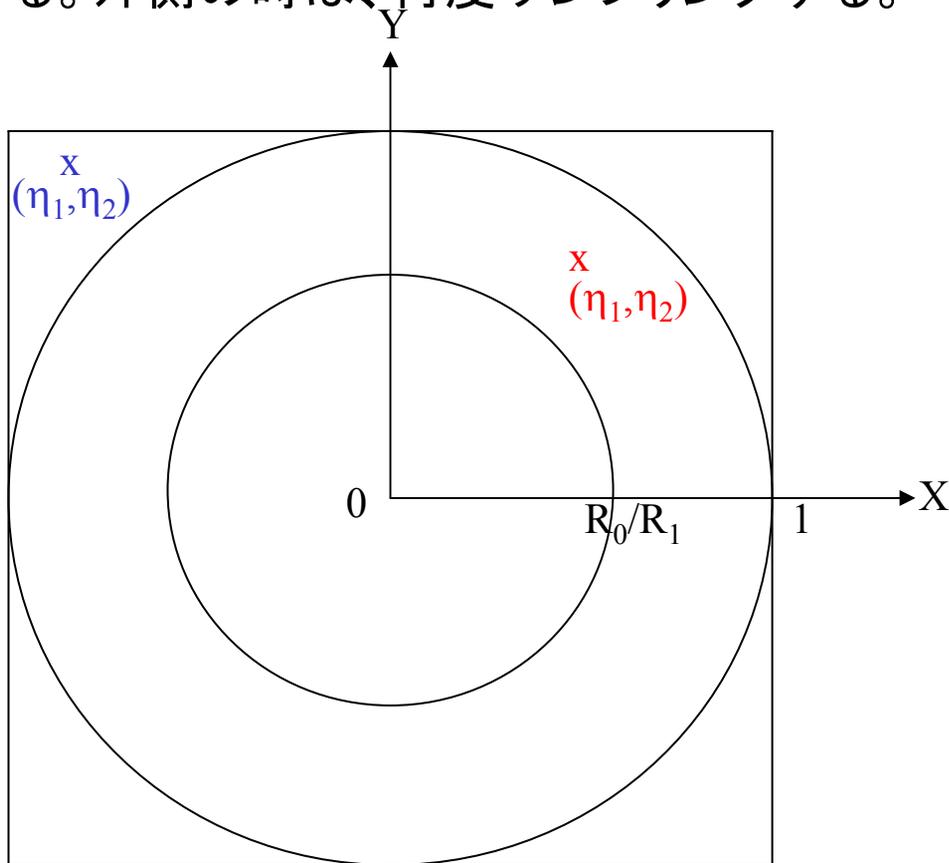
```
phai=PI*(2.0*RN2-1.0)
```

! include 'include/egs5_misc.f' must be included to use PI.

```
xin=rr*cos(phai)
```

```
yin=rr*sin(PHAI)
```

- 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

```
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
```

点等方線源の場合の方向の決定

- 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

```
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
```

ucrz_nai.f, uccg_nai.f 等のサンプルユーザーコードで正在している等方線源

1. ユニツト4から読み込む方向余弦のデータをuin=vin=win=0.0にする。
2. メインプログラムで、isot=1にセツトされる。

```
! -----  
! Set isotropic source flag if uin=vin=win=0  
! -----  
isot=0          ! monodirectional
```

- ```
if (uin+vin+win.eq.0.0) isot=1
```
3. Z-軸に対象な形状を仮定しているのて、win だけをサンプリングで決め、uin=0として方向余弦を求める。  
winとして、前方のみを考慮している場合には、次のように、決定される。

```
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
```