

KEK Internal 2011-7  
December 2011  
R/D

# Lecture Notes of Practices on How to Write Source Routine

H. Hirayama and Y. Namito



High Energy Accelerator Reserach Organization

**©High Energy Accelerator Research Organization (KEK), 2011**

KEK Reports are available from

High Energy Accelerator Research Organization (KEK)  
1-1 Oho, Tsukuba-sh  
Ibaraki-ken, 305-0801  
JAPAN

Phone: +81-29-864-5137  
Fax: +81-29-864-4604  
E-mail: [irdpub@mail.kek.jp](mailto:irdpub@mail.kek.jp)  
Internet: <http://www.kek.jp>

# Lecture Notes of Practices on How to Write Source Routine

H. Hirayama and Y. Namito



*High Energy Accelerator Research Organization*

# Contents

<b>Japanese Parts</b>	<b>1</b>
1 基にするユーザーコード <code>ucsource.f</code> の概要	2
2 実習課題 1 : 線源エネルギー	2
2.1 Co-60 の $\gamma$ -線源	2
2.1.1 <code>if</code> 文の使用する方法	2
2.1.2 <code>data</code> 文の使用する方法	3
2.1.3 <code>data</code> ファイルを使用する方法	5
2.2 Ir-192 の $\gamma$ 線源	6
2.3 Sr-90- $\beta$ 線源	8
2.3.1 ICRU Report 56 のデータを使用する場合	8
2.3.2 RADAR - The Decay Data のデータを使用する場合	12
3 実習課題 2 : 線源位置	15
3.1 直接サンプリング	16
3.2 Rejection 法	17
4 実習課題 4 : 線源方向 ( $2\pi$ )	18
4.1 直接サンプリング	18
4.2 Rejection 法	20
<b>English Parts</b>	<b>21</b>
1 Outlines of user code <code>ucsource.f</code>	22
2 Practice 1: Source energy	22
2.1 $\gamma$ -rays from Co-60	22
2.1.1 A way to use <code>if</code> statement	22
2.1.2 A way to use <code>data</code> statement	23
2.1.3 A way to use <code>data</code> file.	25
2.2 $\gamma$ -rays from Ir-192	26
2.3 $\beta$ -rays from Sr-90	28
2.3.1 Using ICRU Report 56 data	28
2.3.2 Using RADAR - The Decay Data	32
3 Practice 2: Source Position	35
3.1 Direct sampling	36
3.2 Rejection method	37
4 Practice 4: Source Direction	38
4.1 Direct sampling	38
4.2 Rejection method	40
<b>Appendix: Full listings of <code>ucsource.f</code></b>	<b>42</b>

線源の作り方実習  
(Japanese Parts)

## 1 基にするユーザーコード `ucsource.f` の概要

形状としては、Fig. 1 に示すように `cg` を用いた円筒形状である。各種の線源のテストを行うことを目的にしているので、物質は全て真空 (0) に設定している。単一エネルギー (1.253MeV) の光子が、Z-軸上-5cm の位置からビーム状に入力する様に設定されている。

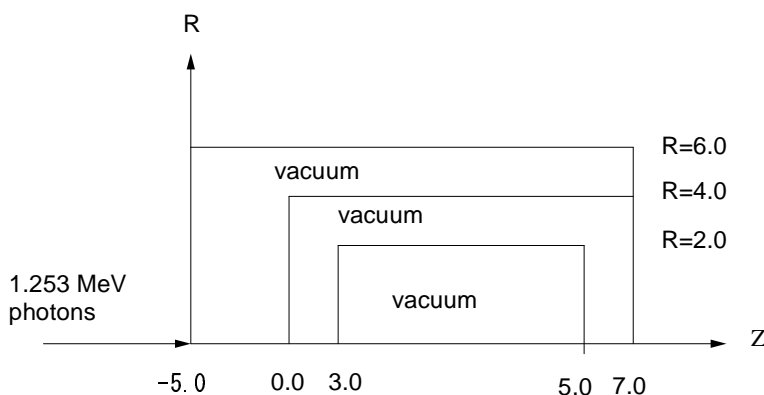


Figure 1: Geometry of `ucsource.f`

## 2 実習課題 1 : 線源エネルギー

### 2.1 Co-60 の $\gamma$ -線源

線源を 1.173MeV と 1.333MeV の線が同じ確率で発生する Co-60 の  $\gamma$ -線源に変更する。if 文を使用する方法、data 文を使用する方法とデータをファイルから読み込む方法がある。

#### 2.1.1 if 文の使用する方法

1. `cp ucsource.f ucsource1_0.f`

2. `ucsource1_0.f` の変更

- 線源エネルギーに関する配列を増やす。

```
* esbin(1),spg(1),spe(1)
```

を

```
* esbin(2),spg(2),spe(2)
```

に変更。

```
nsebin=1
```

を

```
nsebin=2
```

に変更。

- 運動エネルギーの最大値を変更する。

```
ekein=1.253      ! Kinetic energy
```

を

```
ekein=1.333      ! Kinetic energy
```

に変更。

```
esbin(1)=ekein
```

を

```
esbin(1)=1.173  
esbin(2)=1.333
```

に変更。

- エネルギーのサンプリング部分を変更する。

```
ekein = ekein  
spg(1)=spg(1)+1.0
```

を

```
call randomset(rnnow)  
if(rnnow.le.0.5) then  
  ekein = 1.173  
  spg(1)=spg(1)+1.0  
else  
  ekein = 1.333  
  spg(2)=spg(2)+1.0  
end if
```

に変更。

3. `ucsource1_0.f` を `egs5run` で実行する。

- Linux 又は Cygwin の場合

ユーザーコード名として、`ucsource1_0` を、ユニット 4 のファイル名として `ucsource` を入力し、ユニット 25 には”return”を入力する。

”Does this user code read from the terminal?” に対して 1 を入力する。

- DOS の場合

```
egs5run ucsource1_0 ucsource ucsource
```

- `ucsource1_0.f` 等が、`egs5run.bat` を実行しているディレクトリーと別なディレクトリーにある場合は、ディレクトリー名を記載する。DOS の場合、ディレクトリーの識別子は、/ ではなく ¥ であるので、間違わないように注意する。

4. `egs5job.out` の光子の線源スペクトルが、1 対 1 に近い比率になっていることを確認する。

```
Sampled source spectrum  
Upper energy      particles/source  
1.1730    MeV--      Gamma      Electron      pdf  
1.3330    MeV--      0.49710    0.0000  
1.3330    MeV--      0.50290    0.0000
```

### 2.1.2 data 文の使用する方法

1. `cp ucsource1_0.f ucsource1_1.f`

2. `ucsource1_1.f` の変更

- `real*8` 宣言に `espdf(2)`, `escdf(2)` を追加する。

```
real*8  
* availke,tnum,wtin,wtsum,xi0,yi0,zi0,  
* esbin(2),spg(2),spe(2)
```

! Local variables

を

```
real*8                                ! Local variables
* availke,tnum,wtin,wtsum,xi0,yi0,zi0,
* esbin(2),spg(2),spe(2),espdf(2),escdf(2)
```

に変更。

- integer の宣言後に data 文を定義する。

```
integer
* i,icases,idin,ie,ifti,ifto,ii,j,k,n,nd,ner,nsebin
```

を

```
integer
* i,icases,idin,ie,ifti,ifto,ii,j,k,n,nd,ner,nsebin

data esbin/1.173,1.333/
data espdf/0.5,0.5/
```

に変更。

- nsebin=2 の後に、cdf を計算する文を追加する。

```
nsebin=2
```

を

```
nsebin=2
!-----
! Calculate cdf from pdf
!-----
tnum=0.D0
do ie=1,nsebin
  tnum=tnum+espdf(ie)
end do

escdf(1)=espdf(1)/tnum
do ie=2,nsebin
  escdf(ie)=escdf(ie-1)+espdf(ie)/tnum
end do
```

に変更。

- 運動エネルギーの最大値の表現を変更する。

```
ekein=1.333      ! Kinetic energy
```

を

```
ekein=esbin(nsebin) ! Maximum kinetic energy
```

に変更。

- 不要な文を削除する。

```
esbin(1)=1.173
esbin(2)=1.333
```

を削除する。

- 線源エネルギーのサンプリング部を変更する。

```
!-----
! Determine source energy
!-----
call randomset(rnnow)
if(rnnow.le.0.5) then
  ekein = 1.173
  spg(1)=spg(1)+1.0
else
  ekein = 1.333
  spg(2)=spg(2)+1.0
end if
```



を

```
! -----
! Determine source energy
! -----
call randomset(rnnow)
do ie=1,nsebin
  if(rnnow.le.escdf(ie)) go to 1000
end do
1000 ekein=esbin(ie)
  if(iqin.eq.0) then
    spg(ie)=spg(ie)+1.0
  else
    spe(ie)=spe(ie)+1.0
  end if
```

に変更。

3. `ucsource1.1.f` を `egs5run` で実行する。

- Linux の場合  
ユーザーコード名として、`ucsource1.1` を、ユニット 4 のファイル名として `ucsource` を入力し、ユニット 25 には”return”を入力する。  
”Does this user code read from the terminal?” に対して 1 を入力する。
- DOS の場合  
`egs5run ucsource1.1 ucsource ucsource`

4. `egs5job.out` の光子の線源スペクトルが、1 対 1 に近い比率になっていることを確認する。

```
Sampled source spectrum
Upper energy      particles/source      Electron      pdf
1.1730 MeV--      0.49710          0.0000
1.3330 MeV--      0.50290          0.0000
```

### 2.1.3 data ファイルを使用する方法

1. `cp ucsource1.1.f ucsource1.2.f`
2. `ucsource1.2.f` の変更

- local variable を変更する。

```
real*8                                     ! Local variables
* availke,tnum,wtin,wtsum,xi0,yi0,zi0,
* esbin(2),spg(2),spe(2),espdf(2),escdf(2)
```

を

```
real*8                                     ! Local variables
* availke,tnum,wtin,wtsum,xi0,yi0,zi0,esbin(MXEBIN),
* spg(MXEBIN),spe(MXEBIN),espdf(MXEBIN),escdf(MXEBIN)
```

に変更する。

- data 文

```
data esbin/1.173,1.333/
data espdf/0.5,0.5/
```

を削除する。

- open 文を追加する。

```
open(6,FILE='egs5job.out',STATUS='unknown')
```

を

```
open(6,FILE='egs5job.out',STATUS='unknown')
open(2,file='co60.inp',status='unknown')
```

に変更。

- co60.inp は、線源のエネルギーとその確率密度関数で以下の内容のファイルであり、配布ファイルに含まれている。

```
1.173,1.333
0.5,0.5
```

- nsebin=2 の後に、以下を挿入する。

```
read(2,*) (esbin(i),i=1,nsebin)
read(2,*) (espdf(i),i=1,nsebin)
```

- 結果の出力部を変更する。

```
write(6,170) esbin(ie),spg(ie),spe(ie)
170 FORMAT(G10.5,' MeV--',8X,G12.5,8X,G12.5)
```

を

```
write(6,170) esbin(ie),spg(ie),spe(ie),espdf(ie)/tnum
170 FORMAT(G10.5,' MeV--',8X,G12.5,8X,G12.5,8X,G12.5)
```

に変更する。

### 3. ucsouce1\_2.f を egs5run で実行する。

- Linux の場合

ユーザーコード名として、ucsource1\_2 を、ユニット 4 のファイル名として ucsource を入力し、ユニット 25 には”return”を入力する。

”Does this user code read from the terminal?”に対して 1 を入力する。

- DOS の場合

```
egs5run ucsouce1_2 ucsource ucsource
```

### 4. egs5job.out の光子の線源スペクトルが、1対1に近い比率になっていることを確認する。

```
Sampled source spectrum
Upper energy      particles/source
1.1730 MeV--      Gamma           Electron        pdf
1.3330 MeV--      0.49710         0.0000         0.50000
                  0.50290         0.0000         0.50000
```

## 2.2 Ir-192 の $\gamma$ 線源

Ir-192 から放出される  $\gamma$  線のエネルギーと崩壊当たりの放出率は、以下の通りである。(アイソトープ手帳第 10 版)

Energy	Emission Rate
0.296	28.7
0.308	30.0
0.317	82.7
0.468	47.8
0.589	4.5
0.604	8.2
0.612	5.3

1. `cp ucsource1_2.f ucsource2.f`

2. `ucsource2.f` の変更

- 線源データファイルに関する `open` 文を変更する。

```
open(2,file='co60.inp',status='unknown')
```

を

```
open(2,file='ir192.inp',status='unknown')
```

に変更。

- `ir192.inp` は、線源のエネルギーとその確率密度関数で以下の内容のファイルで、配布ファイルに含まれている。

```
0.296,0.308,0.317,0.468,0.589,0.604,0.612
0.287,0.300,0.827,0.478,0.045,0.082,0.053
```

- 線源のデータ数を変更する。<sup>1</sup>

```
nsebin=2
```

を

```
nsebin=7
```

に変更。

3. `ucsource2.f` を `egs5run` で実行する。

- Linux の場合

ユーザーコード名として、`ucsource2` を、ユニット 4 のファイル名として `ucsource` を入力し、ユニット 25 には”return”を入力する。

”Does this user code read from the terminal?”に対して 1 を入力する。

- DOS の場合

```
egs5run ucsource2 ucsource ucsource
```

---

<sup>1</sup>この問題のように、配列の引数となる変数の値を変更する場合には、まずデバッガー機能を含めてコンパイル、実行を行い、配列範囲外アクセスが起きないことを確認するべきである。方法としては、UNIX 又は Cygwin の場合は、”egs5run”と入力するところで”egs5run db”と入力する。これにより、デバッガー機能を含めたコンパイルが行われる。つぎに”egs5job.exe”と入力して、計算を実行する。DOS の場合は、”egs5run\_db ucsource2 ucsource”を実行する。配列範囲外アクセスが起きなければ計算は通常通り終了する。(追加的なメッセージはなにも表示されない) 配列範囲外アクセスが起きた場合には、ソースのどの行で、どの配列の何番目の要素に不正なアクセスが行われたかが表示されるので、ソースの当該部分を修正する。なお、デバッガーを含めてコンパイルした場合実行速度が低下するので、デバッガーの使用はプログラム変更の場合のみとする方がよい。

4. egs5job.out の光子の線源スペクトルが、設定した比率に近いものになっていることを確認する。

Sampled source spectrum		particles/source		
Upper energy		Gamma	Electron	pdf
.29600	MeV--	0.14040	0.0000	0.13851
.30800	MeV--	0.14210	0.0000	0.14479
.31700	MeV--	0.40050	0.0000	0.39913
.46800	MeV--	0.22930	0.0000	0.23069
.58900	MeV--	0.22000E-01	0.0000	0.21718E-01
.60400	MeV--	0.39300E-01	0.0000	0.39575E-01
.61200	MeV--	0.26400E-01	0.0000	0.25579E-01

## 2.3 Sr-90- $\beta$ 線源

$\beta$  線源は、 $\gamma$  線源と異なり、スペクトルは連続である。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も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し、乱数により対応するエネルギー区間をサンプリングし、エネルギー区間内では、一様分布として直線内挿によりエネルギーを決定する方法である。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。

この方法を理解するために、Sr-90 の  $\beta$  線を例にしてサンプリングルーチンを作成する。

### 2.3.1 ICRU Report 56 のデータを使用する場合

ICRU Report 56 には、Sr-90 の  $\beta$  線スペクトルが、(エネルギー / 最大エネルギー) を 41 等分した各区分当たりの崩壊当たりの  $\beta$  線数で与えられている。(次図及び表) このデータを使用して  $\beta$  線のエネルギーを決定するルーチンを作成する。

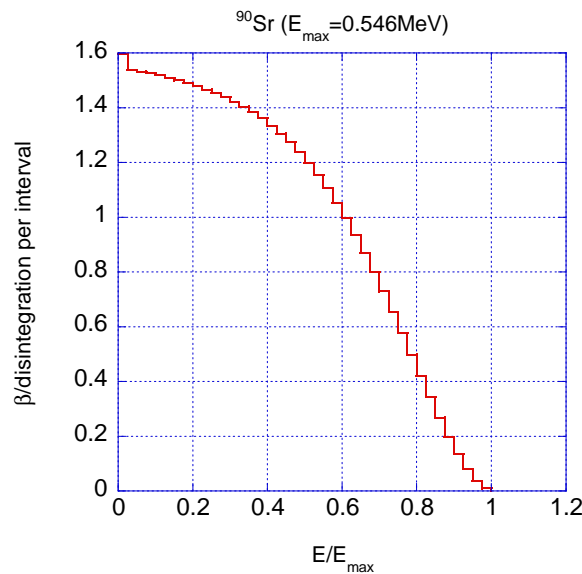


Figure 2:  $\beta$ -ray spectrum from Sr-90(ICRU Report 56).

Table 1  $\beta$ -ray spectrum from Sr-90(ICRU Report 56).

$E/E_{max}$	$\beta$ per dis. per bin	$E/E_{max}$	$\beta$ per dis per bin
0.00	1.597	0.025	1.538
0.05	1.532	0.075	1.526
0.10	1.518	0.125	1.509
0.15	1.500	0.175	1.490
0.20	1.479	0.225	1.466
0.25	1.453	0.275	1.439
0.30	1.422	0.325	1.404
0.35	1.384	0.375	1.361
0.40	1.335	0.425	1.306
0.45	1.274	0.475	1.238
0.50	1.198	0.525	1.154
0.55	1.106	0.575	1.053
0.60	0.997	0.625	0.935
0.65	0.870	0.675	0.801
0.70	0.729	0.725	0.654
0.75	0.577	0.775	0.498
0.80	0.420	0.825	0.343
0.85	0.268	0.875	0.198
0.90	0.135	0.925	0.081
0.95	0.038	0.975	0.010
1.00	0.000		

1. cp ucsource2.f ucsource3.f

2. ucsource3.f の変更

- local variable に, deltaes, emax を追加する。

```

real*8                                     ! Local variables
* availke, tnum, wtin, wtsum, xi0, yi0, zi0, esbin(MXEBIN),
* spg(MXEBIN), spe(MXEBIN), espdf(MXEBIN), escdf(MXEBIN)

```

を

```

real*8                                     ! Local variables
* availke, tnum, wtin, wtsum, xi0, yi0, zi0, esbin(MXEBIN),
* spg(MXEBIN), spe(MXEBIN), espdf(MXEBIN), escdf(MXEBIN),
* deltaes, emax

```

に変更。

- 線源に関する open 文を変更する。

```
open(2, file='ir192.inp', status='unknown')
```

を

```
open(2, file='sr90beta.inp', status='unknown')
```

に変更。

- sr90beta.inp は、上記の崩壊当たり各区分エネルギー当たりの  $\beta$  線の放出率であり、以下の内容のファイルで、配布ファイルに含まれている。放出率から累積分布関数 (cdf) を求めてサンプリングに使用する。

```

0.546
41
0.025
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

```

0.546 は  $\beta$  線の最大エネルギー ( $E_{max}$ , MeV)、41 は分点数、0.025 は、 $E/E_{max}$  の区分幅である。

- 線源データファイルから nsebin を読み込み部を変更する。

```

nsebin=7                ! Number of source energy bins
read(2,*) (esbin(i),i=1,nsebin)
read(2,*) (espdf(i),i=1,nsebin)

```

を

```

read(2,*) emax          ! Maximum beta-ray energy
read(2,*) nsebin       ! Number of source energy bins
read(2,*) deltaes      ! Source energy bin width in MeV
read(2,*) (espdf(i),i=1,nsebin)

```

に変更。

- エネルギービンの値を計算する文を追加する。

```

do ie=1,nsebin
  tnum=tnum+espdf(ie)
end do

```

を

```

do ie=1,nsebin
  esbin(ie)=(ie-1)*deltaes*emax
  tnum=tnum+espdf(ie)
end do

```

に変更。

- cdf作成の部分を変更する。

```

escdf(1)=espdf(1)/tnum
do ie=2,nsebin
  escdf(ie)=escdf(ie-1)+espdf(ie)/tnum
end do

```

を

```

escdf(1)=0.0
do ie=2,nsebin
  escdf(ie)=escdf(ie-1)+espdf(ie-1)/tnum
end do

```

に変更。

- 線源粒子の種類を変更する。

```

iqin=0                ! Incident charge - photons

```

を

```

iqin=-1               ! Incident charge - electrons

```

に変更。

- ヒストリー数を増やす。

```

ncases=10000

```

を

```
ncases=100000
```

に変更。

- 線源エネルギーのサンプリング部を変更する。

```
do ie=1,nsebin
  if(rnnow.le.escdf(ie)) go to 1000
end do
1000 ekein=esbin(ie)
```

を。

```
do ie=2,nsebin
  if(rnnow.le.escdf(ie)) go to 1000
end do
1000 ekein=esbin(ie-1)+(rnnow-escdf(ie-1))*(esbin(ie)-esbin(ie-1))
*      /(escdf(ie)-escdf(ie-1))
```

に修正。

- 結果の出力部分を変更する。

```
do ie=1,nsebin
! -----
! Gamma spectrum per source
! -----
!   spg(ie)=spg(ie)/ncount
! -----
! Electron spectrum per source
! -----
!   spe(ie)=spe(ie)/ncount
!
!   write(6,170) esbin(ie),spg(ie),spe(ie),espdf(ie)/tnum
170   FORMAT(G10.5,' MeV--',8X,G12.5,8X,G12.5,8X,G12.5)
end do
```

を

```
do ie=2,nsebin
! -----
! Gamma spectrum per source
! -----
!   spg(ie)=spg(ie)/ncount
! -----
! Electron spectrum per source
! -----
!   spe(ie)=spe(ie)/ncount
!
!   write(6,170) esbin(ie),spg(ie),spe(ie),espdf(ie-1)/tnum
170   FORMAT(G10.5,' MeV--',8X,G12.5,8X,G12.5,8X,G12.5)
end do
```

に修正。

### 3. ucsource3.f を egs5run で実行する。

- Linux の場合  
ユーザーコード名として、ucsource3 を、ユニット4のファイル名として ucsource を  
入力し、ユニット25には”return”を入力する。  
”Does this user code read from the terminal?”に対して1を入力する。
- DOS の場合  
egs5run ucsource3 ucsource

4. egs5job.out の電子の線源スペクトルが、設定した比率に近いものになっていることを確認する。

Sampled source spectrum		particles/source		
Upper energy		Gamma	Electron	pdf
.13650E-01	MeV--	0.0000	0.38520E-01	0.39108E-01
.27300E-01	MeV--	0.0000	0.37330E-01	0.37663E-01
.40950E-01	MeV--	0.0000	0.36750E-01	0.37516E-01
.54600E-01	MeV--	0.0000	0.37710E-01	0.37369E-01
.68250E-01	MeV--	0.0000	0.37060E-01	0.37173E-01
.81900E-01	MeV--	0.0000	0.36010E-01	0.36953E-01
.95550E-01	MeV--	0.0000	0.36990E-01	0.36732E-01
.10920	MeV--	0.0000	0.36450E-01	0.36487E-01
.12285	MeV--	0.0000	0.37110E-01	0.36218E-01
.13650	MeV--	0.0000	0.35230E-01	0.35900E-01
.15015	MeV--	0.0000	0.35310E-01	0.35581E-01
.16380	MeV--	0.0000	0.34350E-01	0.35239E-01
.17745	MeV--	0.0000	0.35240E-01	0.34822E-01
.19110	MeV--	0.0000	0.34100E-01	0.34381E-01
.20475	MeV--	0.0000	0.33260E-01	0.33892E-01
.21840	MeV--	0.0000	0.34070E-01	0.33328E-01
.23205	MeV--	0.0000	0.32810E-01	0.32692E-01
.24570	MeV--	0.0000	0.32090E-01	0.31982E-01
.25935	MeV--	0.0000	0.31810E-01	0.31198E-01
.27300	MeV--	0.0000	0.30760E-01	0.30316E-01
.28665	MeV--	0.0000	0.29310E-01	0.29337E-01
.30030	MeV--	0.0000	0.28920E-01	0.28259E-01
.31395	MeV--	0.0000	0.27620E-01	0.27084E-01
.32760	MeV--	0.0000	0.25490E-01	0.25786E-01
.34125	MeV--	0.0000	0.23880E-01	0.24415E-01
.35490	MeV--	0.0000	0.23650E-01	0.22896E-01
.36855	MeV--	0.0000	0.21360E-01	0.21305E-01
.38220	MeV--	0.0000	0.19210E-01	0.19615E-01
.39585	MeV--	0.0000	0.18200E-01	0.17852E-01
.40950	MeV--	0.0000	0.15610E-01	0.16015E-01
.42315	MeV--	0.0000	0.14610E-01	0.14130E-01
.43680	MeV--	0.0000	0.12170E-01	0.12195E-01
.45045	MeV--	0.0000	0.10670E-01	0.10285E-01
.46410	MeV--	0.0000	0.84100E-02	0.83995E-02
.47775	MeV--	0.0000	0.63300E-02	0.65628E-02
.49140	MeV--	0.0000	0.50100E-02	0.48487E-02
.50505	MeV--	0.0000	0.33400E-02	0.33059E-02
.51870	MeV--	0.0000	0.21200E-02	0.19835E-02
.53235	MeV--	0.0000	0.88000E-03	0.93055E-03
.54600	MeV--	0.0000	0.25000E-03	0.24488E-03

### 2.3.2 RADAR - The Decay Data のデータを使用する場合

RADAR - The Decay Data (<http://www.doseinfo-radar.com/RADARDecay.html>) には、多くの  $\beta$  線スペクトルデータが示されている。(以下、「RADAR データ」という。) RADAR データでは、20 等分したエネルギー区分に対する確率密度 (PDF:Probability distribution function) が与えられている。このデータを使用して Sr-90 の  $\beta$  線のエネルギーを決定するルーチンを作成する。



Table 2  $\beta$ -ray spectrum from RADAR.

$E_{min}$	$E_{max}$	PDF
0.0000	0.0273	7.79E-02
0.0273	0.0546	7.60E-02
0.0546	0.0819	7.50E-02
0.0819	0.1092	7.40E-02
0.1092	0.1365	7.30E-02
0.1365	0.1638	7.17E-02
0.1638	0.1911	7.01E-02
0.1911	0.2184	6.80E-02
0.2184	0.2457	6.53E-02
0.2457	0.2730	6.19E-02
0.2730	0.3003	5.78E-02
0.3003	0.3276	5.27E-02
0.3276	0.3549	4.68E-02
0.3549	0.3822	4.01E-02
0.3822	0.4095	3.27E-02
0.4095	0.4368	2.48E-02
0.4368	0.4641	1.71E-02
0.4641	0.4914	9.75E-03
0.4914	0.5187	4.28E-03
0.5187	0.5460	1.01E-03

1. cp ucsource3.f ucsource3\_1.f

2. ucsource3\_1.f の変更

- local variable の, deltaes, emax を beint に変更する。beint は、崩壊当たりに放出される  $\beta$  線数である。

```
* deltaes, emax
```

を

```
* beint
```

に変更。

- local variable の integer に iebeta を追加する。

```
* i, icas, idin, ie, ifti, ifto, ii, j, k, n, nd, ner, nsebin
```

を

```
* i, icas, idin, ie, ifti, ifto, ii, j, k, n, nd, ner, nsebin, ibeta
```

に変更。

- 線源情報に関する character 文を追加する。

```
character*24 medarr(MXMED)
```

を

```
character*24 medarr(MXMED)
character*20 soinf
```

に変更。

- 線源に関する open 文を変更する。

```
open(2,file='sr90beta.inp',status='unknown')
```

を

```
open(2,file='Sr-90.dat',status='unknown')
```

に変更。

- Sr-90.dat は、以下の内容のファイルで、配布ファイルに含まれている。

```
Sr-90 Beta-
-1,      1.0000
0.0273,7.79E-02
0.0546,7.60E-02
0.0819,7.50E-02
0.1092,7.40E-02
0.1365,7.30E-02
0.1638,7.17E-02
0.1911,7.01E-02
0.2184,6.80E-02
0.2457,6.53E-02
0.2730,6.19E-02
0.3003,5.78E-02
0.3276,5.27E-02
0.3549,4.68E-02
0.3822,4.01E-02
0.4095,3.27E-02
0.4368,2.48E-02
0.4641,1.71E-02
0.4914,9.75E-03
0.5187,4.28E-03
0.5460,1.01E-03
```

-1 は、崩壊に伴い放出される  $\beta$  線が電子であることを示している。陽電子の場合は、1 となる。

- 線源データファイルから nsebin を読み込み部とその処理に関する部分を変更する。

```
read(2,*) emax           ! Maximum beta-ray energy  RADAR
read(2,*) nsebin         ! Number of source energy bins
read(2,*) deltaes       ! Source energy bin width in MeV
read(2,*) (espdf(i),i=1,nsebin)
```

```
!-----
! Calculate cdf from pdf
!-----
```

```
tnum=0.D0
do ie=1,nsebin
  esbin(ie)=(ie-1)*deltaes*emax
  tnum=tnum+espdf(ie)
end do
```

```
escdf(1)=0.0
do ie=2,nsebin
  escdf(ie)=escdf(ie-1)+espdf(ie-1)/tnum
end do
```

```
iqin=-1           ! Incident charge - electrons
ekein=esbin(nsebin) ! Maximum kinetic energy
```

を

```
nsebin=21           ! Number of bin for beta-ray spectrum
esbin(1)=0.d0
espdf(1)=0.d0
read(2,'(A20)') soinf ! Source information
read(2,*) ibeta, beint ! Charge od beta-ray, emitted beta per decay
do ie=2,nsebin
  read(2,*) esbin(ie),espdf(ie) ! Upper bin and pdf
end do
```

```

!-----
! Calculate cdf from pdf
!-----
do ie=2,nsebin
  escdf(ie)=escdf(ie-1)+espdf(ie)/tnum
end do

iqin=ibeta          ! Incident charge - electrons
ekein=esbin(nsebin) ! Maximum kinetic energy

```

に変更。

- 結果の出力部分を変更する。

```
write(6,170) esbin(ie),spg(ie),spe(ie),espdf(ie-1)/tnum
```

を

```
write(6,170) esbin(ie),spg(ie),spe(ie),espdf(ie)
```

に変更。

### 3. ucsouce3\_1.f を egs5run で実行する。

- Linux の場合

ユーザーコード名として、ucsource3\_1 を、ユニット 4 のファイル名として ucsource を入力し、ユニット 25 には”return”を入力する。

”Does this user code read from the terminal?”に対して 1 を入力する。

- DOS の場合

```
egs5run ucsource3_1 ucsource
```

### 4. egs5job.out の電子の線源スペクトルが、設定した比率に近いものになっていることを確認する。

Sampled source spectrum		particles/source		
Upper energy		Gamma	Electron	pdf
.27300E-01	MeV--	0.0000	0.77120E-01	0.77905E-01
.54600E-01	MeV--	0.0000	0.75250E-01	0.76005E-01
.81900E-01	MeV--	0.0000	0.74090E-01	0.75005E-01
.10920	MeV--	0.0000	0.74380E-01	0.74004E-01
.13650	MeV--	0.0000	0.73020E-01	0.73004E-01
.16380	MeV--	0.0000	0.70870E-01	0.71704E-01
.19110	MeV--	0.0000	0.69590E-01	0.70104E-01
.21840	MeV--	0.0000	0.68480E-01	0.68004E-01
.24570	MeV--	0.0000	0.65610E-01	0.65304E-01
.27300	MeV--	0.0000	0.62860E-01	0.61904E-01
.30030	MeV--	0.0000	0.58070E-01	0.57803E-01
.32760	MeV--	0.0000	0.53090E-01	0.52703E-01
.35490	MeV--	0.0000	0.47070E-01	0.46803E-01
.38220	MeV--	0.0000	0.39880E-01	0.40102E-01
.40950	MeV--	0.0000	0.33350E-01	0.32702E-01
.43680	MeV--	0.0000	0.24820E-01	0.24801E-01
.46410	MeV--	0.0000	0.17320E-01	0.17101E-01
.49140	MeV--	0.0000	0.97500E-02	0.97506E-02
.51870	MeV--	0.0000	0.43600E-02	0.42803E-02
.54600	MeV--	0.0000	0.10200E-02	0.10101E-02

## 3 実習課題 2 : 線源位置

半径 1.5cm から 4cm の領域に一様に分布している面線源の場合に、線源位置をサンプリングするルーチンを作成する。

### 3.1 直接サンプリング

半径  $R_0$  から  $R_1$  の領域に一様に分布している面線源の場合の半径の分布に関する確率密度関数 (pdf) は、次のようになる。

$$f(r)dr = c \times 2\pi r dr$$
$$\int_{R_0}^{R_1} f(\xi)d\xi = c\pi[\xi^2]_{R_0}^{R_1} = c\pi[R_1^2 - R_0^2] = 1$$
$$c = \frac{1}{\pi(R_1^2 - R_0^2)} \rightarrow f(r)dr = \frac{2rdr}{R_1^2 - R_0^2}$$

線源位置の半径 ( $r$ ) は、以下の式を解くことにより決定する。

$$\eta = \int_{R_0}^r f(\xi)d\xi = \frac{r^2 - R_0^2}{R_1^2 - R_0^2}$$
$$r = \sqrt{R_0^2 + \eta(R_1^2 - R_0^2)}$$

$x$  と  $y$  の位置は、 $\phi$  を、 $0$  から  $2\pi$  の一様分布から決定し、

$$x = r \cos \phi, \quad y = r \sin \phi$$

により決定する。具体的なプログラムは、以下のようにする。

1. cp ucsource.f ucsource4.f

2. ucsource4.f の変更

- local variable に r02,r12,phai を追加する。

```
* esbin(1),spg(1),spe(1)
```

を

```
* esbin(1),spg(1),spe(1),r02,r12,phai,rr0
```

に変更する。

- 線源の weight 設定後に、r01,r12 の設定文を挿入する。

```
wtin=1.0          ! Weight = 1 since no variance reduction used
```

を

```
wtin=1.0          ! Weight = 1 since no variance reduction used
r02=1.5*1.5
r12=4.0*4.0
```

に変更。

- 線源位置のサンプリングを挿入する。

```
! -----
! Determine source position
! -----
```

を

```
! -----
! Determine source position
! -----
call randomset(rnnow)
rr0=sqrt(r02+rnnow*(r12-r02))
call randomset(rnnow)
phai=PI*(2.0*rnnow-1.0)
xin=rr0*cos(phai)
yin=rr0*sin(phai)
```

に変更。

3. ucsource4.f を egs5run で実行する。

- Linux の場合  
ユーザーコード名として、ucsource4 を、ユニット 4 のファイル名として ucsource を入力し、ユニット 25 には”return”を入力する。  
”Does this user code read from the terminal?” に対して 1 を入力する。
- DOS の場合  
egs5run ucsource4 ucsource ucsource

4. CGView で、座標軸を X-Y にし、軸を若干傾け、半径 1.5-4.0 の領域から光子が出ていることを確認する。

### 3.2 Rejection 法

Rejection 法では、 $x$  及び  $y$  をそれぞれ -1 から 1 の範囲の正方形内で一様にサンプリングし、 $R_0/R_1 \leq r = \sqrt{x^2 + y^2} \leq 1.0$  の場合には  $x * R_1$  及び  $y * R_1$  を線源位置とし、それ以外の場合は、サンプリングをやり直す (新たな乱数を用いてサンプリングする) ことにより、位置を決定する。具体的なプログラムは、以下のようにする。

1. cp ucsource4.f ucsource5.f

2. ucsource5.f の変更

- local variable を変更する。  
\* esbin(1), spg(1), spe(1), r02, r12, phai, rr0

を

```
* esbin(1), spg(1), spe(1), r0, r1, rr0
```

に変更。

- r02, r12 の設定を r0, r1 の設定に変更する。

```
r02=1.5*1.5  
r12=4.0*4.0
```

を

```
r0=1.5  
r1=4.0
```

に変更。

- 線源位置のサンプリング方法を変更する。

```
! -----  
! Determine source position  
! -----  
call randomset(rnnow)  
rr0=sqrt(r02+rnnow*(r12-r02))  
call RANDOMSET(rnnow)  
phai=PI*(2.0*rnnow-1.0)  
xin=rr0*cos(phai)  
yin=rr0*sin(phai)
```

を

```

!-----
! Determine source position
!-----
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

```

3. ucsouce5.f を egs5run で実行する。

- Linux の場合  
ユーザーコード名として、ucsource5 を、ユニット 4 のファイル名として ucsource を入力し、ユニット 25 には”return”を入力する。  
”Does this user code read from the terminal?” に対して 1 を入力する。
- DOS の場合  
egs5run ucsource5 ucsource ucsource

4. CGView で、直接サンプリングと同じように半径 1.5-4cm の領域から光子が出ていることを確認する。

## 4 実習課題 4 : 線源方向 ( $2\pi$ )

### 4.1 直接サンプリング

等方線源の場合、Z 方向の方向余弦である  $w$  の確率密度関数は、以下の様になる。

$$f(\theta)d\theta = c \times 2\pi \sin \theta d\theta \quad (0 \leq \theta \leq \pi)$$

$$w = \cos \theta$$

とすると

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

$$\int_1^{-1} g(w)dw = -c \times 2\pi \times (-2) = 1$$

から

$$c = \frac{1}{4\pi} \rightarrow g(w)dw = -\frac{1}{2}dw$$

となる。w は、以下の式を解くことにより決定することができる。

$$\eta = \int_1^w g(w)dw = \frac{1}{2}(1-w) \rightarrow w = 1 - 2\eta$$

$1 - 2\eta$  と  $2\eta - 1$  は、等価なので、どちらを使用しても良い。この問題のように  $\cos \theta$  が正の領域のみに限られる等方線源の場合は、

$$\int_1^0 g(w)dw = -c \times 2\pi \times (-1) = 1$$

から

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

となるので、

$$\eta = \int_1^w g(w)dw = w \rightarrow w = 1 - \eta$$

$1 - \eta$  と  $\eta$  は、等価なので、 $w = \eta$  とする。実際のプログラムは、以下の様にする。

1. cp ucsource.f ucsource6.f

2. ucsource6.f の変更

- local variable に,phai,rr0 を追加する。

```
real*8                                     ! Local variables
* availke,tnum,wtin,wtsum,xi0,yi0,zi0,
* esbin(1),spg(1),spe(1)
```

を

```
real*8                                     ! Local variables
* availke,tnum,wtin,wtsum,xi0,yi0,zi0,
* esbin(1),spg(1),spe(1),phai,rr0
```

に変更。

- 線源の方向をサンプリングする文を挿入する。

```
! -----
! Determine source direction
! -----
```

を

```
! -----
! Determine source direction
! -----
call randomset(rnnow)
win=rnnow
call randomset(rnnow)
phai=PI*(2.0*rnnow-1.0)
uin=dsqrt(1.0-win*win)*cos(phai)
vin=dsqrt(1.0-win*win)*sin(phai)
```

- 方向余弦の規格化

- サンプリングした方向余弦は、規格化されていなければならない。

( $xin*xin+yin*yin+win*win=1.0$ )

この例のやり方の場合は、規格化がされているが、ユーザーが自分のやり方で線源の方向を決定する場合には、規格化されているかどうかを確認する必要がある。

call shower の前の

```
! -----
! Verify the normarization of source direction cosines
! -----
if(abs(uin*uin+vin*vin+win*win-1.0).gt.1.e-6) then
  write(6,fmt="( ' Following source direction cosines are not',
1    ' normarized.' ,3e12.5)")uin,vin,win
  stop
end if
```

が、このためのルーティンである。

3. ucsource6.f を egs5run で実行する。

- Linux の場合

ユーザーコード名として、ucsource6 を、ユニット 4 のファイル名として ucsource を入力し、ユニット 25 には”return”を入力する。

”Does this user code read from the terminal?”に対して 1 を入力する。

- DOS の場合

egs5run ucsoouce6 ucsource

4. CGView で、光子が  $2\pi$  方向に等方的に発生していることを確認する。

## 4.2 Rejection 法

Rejection 法では、 $x, y$  及び  $z$  をそれぞれ  $-1$  から  $1$  の立方体中で一様にサンプリングし、サンプリングされた位置が半径  $1$  の球の内側の場合は、原点からサンプリングされた点に向かう方向を方向余弦とする。球の外側の場合は、サンプリングをやり直す。実際のプログラムは、以下のようにする。

1. `cp ucsource6.f ucsource7.f`

2. `ucsource7.f` の変更

- 線源の方向をサンプリングする部分を修正する。

```
! -----  
! Determine source direction  
! -----  
call randomset(rnnow)  
win=rnnow  
call randomset(rnnow)  
phai=PI*(2.0*rnnow-1.0)  
uin=dsqrt(1.0-win*win)*cos(phai)  
vin=dsqrt(1.0-win*win)*sin(phai)
```

を

```
! -----  
! Determine source direction  
! -----  
1300 call randomset(rnnow)  
      zi0=rnnow  
      call randomset(rnnow)  
      xi0=2.0*rnnow-1.0  
      call randomset(rnnow)  
      yi0=2.0*rnnow-1.0  
      rr0=dsqrt(xi0*xi0+yi0*yi0+zi0*zi0)  
      if(rr0.gt.1.0) go to 1300  
      win = zi0/rr0  
      uin = xi0/rr0  
      vin = yi0/rr0
```

3. `ucsource7.f` を `egs5run` で実行する。

- Linux の場合  
ユーザーコード名として、`ucsource7` を、ユニット 4 のファイル名として `ucsource` を入力し、ユニット 25 には "return" を入力する。  
"Does this user code read from the terminal?" に対して 1 を入力する。
- DOS の場合  
`egs5run ucsouce7 ucsource`

4. CGView で、光子が直接サンプリングの場合と同じように  $2\pi$  方向に等方的に発生していることを確認する。



**Practices of  
How to Write Source Routine  
(English Parts)**

## 1 Outlines of user code `ucsource.f`

Geometry used in `ucsource.f` is a cylindrical geometry with cg as shown in Fig.1. All regions are set to vacuum (0) to test source. 1.253 MeV photon pencil beam enters on the cylinder from  $Z=-5\text{cm}$  along the  $Z$ -axis.

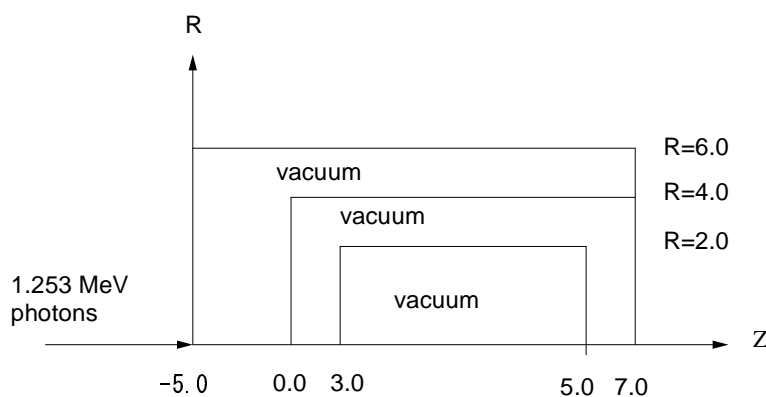


Figure 1: Geometry of `ucsource.f`

## 2 Practice 1: Source energy

### 2.1 $\gamma$ -rays from Co-60

1.173MeV and 1.333MeV photons are emitted with equal probability. There are 3 ways. (1) a way to use `if` statement, (2) a way to use `data` statement and (3) a way to use data file.

#### 2.1.1 A way to use `if` statement

1. `cp ucsource.f ucsource1_0.f`

You must use `copy ucsource.f ucsource1_0.f` or copy function of Windows in the case of DOS.

2. Modify `ucsource1_0.f` as follows:

- Increase the number of source energy bin.

Change

```
* esbin(1),spg(1),spe(1)
```

to

```
* esbin(2),spg(2),spe(2)
```

Change

```
nsebin=1
```

to

```
nsebin=2
```

- Modify the maximum electron kinetic energy used.

Change

```

ekein=1.253      ! Kinetic energy
to
ekein=1.333      ! Kinetic energy

```

- Modify photon energy for each energy bin.  
Change

```

esbin(1)=ekein
to
esbin(1)=1.173
esbin(2)=1.333

```

- Modify statements for source energy sampling.  
Change

```

ekein = ekein
spg(1)=spg(1)+1.0
to
call randomset(rnnow)
if(rnnow.le.0.5) then
  ekein = 1.173
  spg(1)=spg(1)+1.0
else
  ekein = 1.333
  spg(2)=spg(2)+1.0
end if

```

### 3. Run `ucsource1_0.f` by `egs5run`.

- In the case of Linux or Cygwin  
Enter `ucsource1_0` as the user code.  
Enter `ucsource` as the file name for unit 4 and simply enter "return" for unit 25.  
Enter 1 for "Does this user code read from the terminal?".

- In the case of DOS  
`egs5run ucsource1_0 ucsource`

### 4. Check the sampled source photons spectrum in `egs5job.out`.

Example of the result.

Sampled source spectrum			particles/source		
Upper energy		Gamma	Electron	pdf	
1.1730	MeV--	0.49710	0.0000		
1.3330	MeV--	0.50290	0.0000		

#### 2.1.2 A way to use data statement

##### 1. `cp ucsource1_0.f ucsource1_1.f`

##### 2. Modify `ucsource1_1.f` as follows:

- Add `espdf(2)`, `escdf(2)` as local variable.

Change

```

real*8
* availke,tinum,wtin,wtsum,xi0,yi0,zi0,
* esbin(2),spg(2),spe(2)
! Local variables

```

to

```
      real*8                                ! Local variables
      * availke,tnum,wtin,wtsum,xi0,yi0,zi0,
      * esbin(2),spg(2),spe(2),espdf(2),escdf(2)
```

- Insert `data` statements for source  $\gamma$ -ray energies and their emission probabilities.

Change

```
      integer
      * i,icases,idin,ie,ifti,ifto,ii,j,k,n,nd,ner,nsebin
```

to

```
      integer
      * i,icases,idin,ie,ifti,ifto,ii,j,k,n,nd,ner,nsebin

      data esbin/1.173,1.333/
      data espdf/0.5,0.5/
```

- Add statements to get cdf after `nsebin=2`.

Change

```
      nsebin=2
```

to

```
      nsebin=2
!-----
! Calculate cdf from pdf
!-----
      tnum=0.D0
      do ie=1,nsebin
         tnum=tnum+espdf(ie)
      end do

      escdf(1)=espdf(1)/tnum
      do ie=2,nsebin
         escdf(ie)=escdf(ie-1)+espdf(ie)/tnum
      end do
```

- Modify maximum electron kinetic energy used.

Change

```
      ekein=1.333          ! Kinetic energy
```

to

```
      ekein=esbin(nsebin) ! Maximum kinetic energy
```

- Delete following 2 lines.

```
      esbin(1)=1.173
      esbin(2)=1.333
```

- Modify the source energy sampling routine.

Change

```
!-----
! Determine source energy
!-----
      call randomset(rnnow)
      if(rnnow.le.0.5) then
         ekein = 1.173
         spg(1)=spg(1)+1.0
      else
         ekein = 1.333
         spg(2)=spg(2)+1.0
      end if
```

```

to
! -----
! Determine source energy
! -----
call randomset(rnnow)
do ie=1,nsebin
  if(rnnow.le.escdf(ie)) go to 1000
end do
1000 ekein=esbin(ie)
      if(iqin.eq.0) then
          spg(ie)=spg(ie)+1.0
      else
          spe(ie)=spe(ie)+1.0
      end if

```

3. Run `ucsource1_1.f` by `egs5run`.

- In the case of Linux or Cygwin  
Enter `ucsource1_1` as the user code.  
Enter `ucsource` as the file name for unit 4 and simply enter "return" for unit 25.  
Enter 1 for "Does this user code read from the terminal?".

- In the case of DOS  
`egs5run ucsource1_1 ucsource`

4. Check the sampled source photons spectrum in `egs5job.out`.  
Example of the result.

Sampled source spectrum		particles/source		
Upper energy		Gamma	Electron	pdf
1.1730	MeV--	0.49710	0.0000	
1.3330	MeV--	0.50290	0.0000	

### 2.1.3 A way to use data file.

1. `cp ucsource1_1.f ucsource1_2.f`

2. Modify `ucsource1_2.f` as follows:

- Modify local variable

```

      real*8                                     ! Local variables
      * availke,tnum,wtin,wtsum,xi0,yi0,zi0,
      * esbin(2),spg(2),spe(2),espdf(2),escdf(2)

```

to

```

      real*8                                     ! Local variables
      * availke,tnum,wtin,wtsum,xi0,yi0,zi0,esbin(MXEBIN),
      * spg(MXEBIN),spe(MXEBIN),espdf(MXEBIN),escdf(MXEBIN)

```

- Delete following data statements.

```

data esbin/1.173,1.333/
data espdf/0.5,0.5/

```

- Add an open statement for source data file.

Change

```

open(6,FILE='egs5job.out',STATUS='unknown')

```

to

```
open(6,FILE='egs5job.out',STATUS='unknown')
open(2,file='co60.inp',status='unknown')
```

- `co60.inp` is the data file including source photon energies and their probability distribution functions (pdf) constituting by the following data. `co60.inp` is included in the distribution files

```
1.173,1.333
0.5,0.5
```

- Insert following statements after `nsebin=2`.

```
read(2,*) (esbin(i),i=1,nsebin)
read(2,*) (espdf(i),i=1,nsebin)
```

- Modify output statement for obtained results.

Change

```
170 write(6,170) esbin(ie),spg(ie),spe(ie)
FORMAT(G10.5,' MeV--',8X,G12.5,8X,G12.5)
```

to

```
170 write(6,170) esbin(ie),spg(ie),spe(ie),espdf(ie)/tnum
FORMAT(G10.5,' MeV--',8X,G12.5,8X,G12.5,8X,G12.5)
```

### 3. Run `ucsource1_2.f` by `egs5run`.

- In the case of Linux or Cygwin  
Enter `ucsource1_2` as the user code.  
Enter `ucsource` as the file name for unit 4 and simply enter "return" for unit 25.  
Enter 1 for "Does this user code read from the terminal?".
- In the case of DOS  
`egs5run ucsource1_2 ucsource`

### 4. Check the sampled source photons spectrum in `egs5job.out`.

Example of the result.

```
Sampled source spectrum
Upper energy      particles/source
1.1730 MeV--      Gamma      Electron      pdf
1.3330 MeV--      0.49710    0.0000        0.50000
                  0.50290    0.0000        0.50000
```

## 2.2 $\gamma$ -rays from Ir-192

$\gamma$ -ray energies and their emission probabilities per decay from Ir-192 are as follows.

Energy	Emission Rate
0.296	28.7
0.308	30.0
0.317	82.7
0.468	47.8
0.589	4.5
0.604	8.2
0.612	5.3

1. `cp ucsource1_2.f ucsource2.f`

2. Modify `ucsource2.f` as follows:

- Modify the `open` statement for the source data file.

Change

```
open(2,file='co60.inp',status='unknown')
```

to

```
open(2,file='ir192.inp',status='unknown')
```

- `ir192.inp` is the data file including source photon energies and their pdf constituting by the following data and is included in the distribution files.

```
0.296,0.308,0.317,0.468,0.589,0.604,0.612
0.287,0.300,0.827,0.478,0.045,0.082,0.053
```

- Modify the number of  $\gamma$ -ray energies. <sup>1</sup>

Change

```
nsebin=2
```

to

```
nsebin=7
```

3. Run `ucsource2.f` by `egs5run`.

- In the case of Linux or Cygwin  
Enter `ucsource2` as the user code.  
Enter `ucsource` as the file name for unit 4 and simply enter "return" for unit 25.  
Enter 1 for "Does this user code read from the terminal?".
- In the case of DOS  
`egs5run ucsource2 ucsource`

4. Check the sampled source photons spectrum in `egs5job.out`. Example of the results.

```
Sampled source spectrum
                                particles/source
Upper energy                    Gamma            Electron            pdf
.29600    MeV--                  0.14040          0.0000          0.13851
.30800    MeV--                  0.14210          0.0000          0.14479
.31700    MeV--                  0.40050          0.0000          0.39913
.46800    MeV--                  0.22930          0.0000          0.23069
.58900    MeV--                  0.22000E-01     0.0000          0.21718E-01
.60400    MeV--                  0.39300E-01     0.0000          0.39575E-01
.61200    MeV--                  0.26400E-01     0.0000          0.25579E-01
```

---

<sup>1</sup>If it is necessary to change the value of the argument, you must check "out of bound" error by running `egs5` with debug option as follows. In the case of Unix or Cygwin, key in "`egs5run db`" instead of "`egs5run`". Next, execute programme by "`egs5job.exe`". In the case of DOS, execute "`egs5run_db ucsource2 ucsource`".

Modify `egs5run.bat` to use debug mode and save as `egs5run_db.bat`.

If "out of bound error" occurs, the line number and the argument caused error is displayed. Running `egs5` with debug option needs more CPU time than usual way and therefore is used only for this kind of check.

### 2.3 $\beta$ -rays from Sr-90

Electron spectrum of  $\beta$ -decay is continuous. This is different from the case of  $\gamma$ -decay. It is generally difficult to apply a direct method as the sampling of continuous probability. Approximate way but able to apply to any case if a shape of spectrum is given as follows:

- Divide energy of  $\beta$ -rays in equal n-th bins.
- Use fraction of the integration of each energy bin to the integration for whole energy region as pdf.
- Sample the bin using cdf created from pdf and random number.
- Determine energy supposing uniform distribution within each energy bin.

Followings are sampling routines to determine source *beta*-ray energy from Sr-90.

#### 2.3.1 Using ICRU Report 56 data

ICRU Report 56 gives  $\beta$ -ray spectrum from typical radioisotopes in the form of a number of  $\beta$ -rays per bin per disintegrations. As an example,  $\beta$ -ray spectrum from Sr-90 is shown in Fig.2 or Table 1.

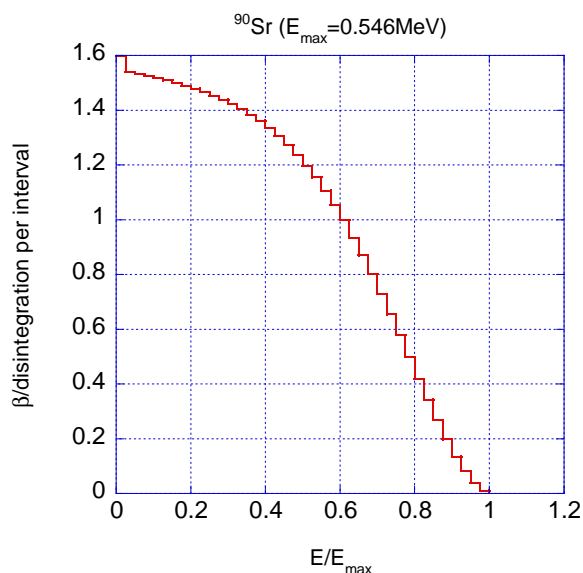


Figure 2:  $\beta$ -ray spectrum from Sr-90(ICRU Report 56).



Table 1  $\beta$ -ray spectrum from Sr-90(ICRU Report 56).

$E/E_{max}$	$\beta$ per dis. per bin	$E/E_{max}$	$\beta$ per dis per bin
0.00	1.597	0.025	1.538
0.05	1.532	0.075	1.526
0.10	1.518	0.125	1.509
0.15	1.500	0.175	1.490
0.20	1.479	0.225	1.466
0.25	1.453	0.275	1.439
0.30	1.422	0.325	1.404
0.35	1.384	0.375	1.361
0.40	1.335	0.425	1.306
0.45	1.274	0.475	1.238
0.50	1.198	0.525	1.154
0.55	1.106	0.575	1.053
0.60	0.997	0.625	0.935
0.65	0.870	0.675	0.801
0.70	0.729	0.725	0.654
0.75	0.577	0.775	0.498
0.80	0.420	0.825	0.343
0.85	0.268	0.875	0.198
0.90	0.135	0.925	0.081
0.95	0.038	0.975	0.010
1.00	0.000		

1. cp ucsource2.f ucsource3.f

2. Modify ucsource3.f as follows:

- Add ,deltaes,emax to local variable.

Change

```

real*8                                     ! Local variables
* availke,tnum,wtin,wtsum,xi0,yi0,zi0,esbin(MXEBIN),
* spg(MXEBIN),spe(MXEBIN),espdf(MXEBIN),escdf(MXEBIN)

```

to

```

real*8                                     ! Local variables
* availke,tnum,wtin,wtsum,xi0,yi0,zi0,esbin(MXEBIN),
* spg(MXEBIN),spe(MXEBIN),espdf(MXEBIN),escdf(MXEBIN),
* deltaes,emax

```

- Modify the open statement for the source data file.

Change

```

open(2,file='ir192.inp',status='unknown')

```

to

```

open(2,file='sr90beta.inp',status='unknown')

```

- sr90beta.inp is the data file including  $\beta$ -ray emission rate per energy bin and is included in the distribution files.

```

0.546
41
0.025
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

```

0.546 is the maximum kinetic energy of  $\beta$ -rays ( $E_{max}$ , MeV), 41 is the number of energy bin and 0.025 is the bin width given by  $E/E_{max}$ .

- Modify the statements to read data from the source data file.

Change

```

nsebin=7          ! Number of source energy bins
read(2,*) (esbin(i),i=1,nsebin)
read(2,*) (espdf(i),i=1,nsebin)

```

to

```

read(2,*) emax          ! Maximum beta-ray energy
read(2,*) nsebin       ! Number of source energy bins
read(2,*) deltaes      ! Source energy bin width in MeV
read(2,*) (espdf(i),i=1,nsebin)

```

- Insert statement to calculate energy corresponding each energy bin. Change

```

do ie=1,nsebin
  tnum=tnum+espdf(ie)
end do

```

to

```

do ie=1,nsebin
  esbin(ie)=(ie-1)*deltaes*emax
  tnum=tnum+espdf(ie)
end do

```

- Modify statements related to calculate cdf.

```

escdf(1)=espdf(1)/tnum
do ie=2,nsebin
  escdf(ie)=escdf(ie-1)+espdf(ie)/tnum
end do

```

to

```

escdf(1)=0.0
do ie=2,nsebin
  escdf(ie)=escdf(ie-1)+espdf(ie-1)/tnum
end do

```

- Modify the type of source particle.

Change

```

iqin=0          ! Incident charge - photons

```

to

```

iqin=-1        ! Incident charge - electrons

```

- Increase history number. Change

```

ncases=10000

```

to

```

ncases=100000

```

- Modify the source energy sampling routines as follows.

Change

```

do ie=1,nsebin
  if(rnnow.le.escdf(ie)) go to 1000
end do
1000 ekein=esbin(ie)

to

do ie=2,nsebin
  if(rnnow.le.escdf(ie)) go to 1000
end do
1000 ekein=esbin(ie-1)+(rnnow-escdf(ie-1))*(esbin(ie)-esbin(ie-1))
*      /(escdf(ie)-escdf(ie-1))

```

- Modify output statements of results as follows.

Change

```

do ie=1,nsebin
! -----
! Gamma spectrum per source
! -----
!   spg(ie)=spg(ie)/ncount
! -----
! Electron spectrum per source
! -----
!   spe(ie)=spe(ie)/ncount
!
170 write(6,170) esbin(ie),spg(ie),spe(ie),espdf(ie)/tnum
   FORMAT(G10.5,' MeV--',8X,G12.5,8X,G12.5,8X,G12.5)
end do

to

do ie=2,nsebin
! -----
! Gamma spectrum per source
! -----
!   spg(ie)=spg(ie)/ncount
! -----
! Electron spectrum per source
! -----
!   spe(ie)=spe(ie)/ncount
!
170 write(6,170) esbin(ie),spg(ie),spe(ie),espdf(ie-1)/tnum
   FORMAT(G10.5,' MeV--',8X,G12.5,8X,G12.5,8X,G12.5)
end do

```

### 3. Run `ucsource3.f` by `egs5run`.

- In the case of Linux or Cygwin
  - Enter `ucsource3` as the user code.
  - Enter `ucsource` as the file name for unit 4 and simply enter "return" for unit 25.
  - Enter 1 for "Does this user code read from the terminal?".
- In the case of DOS
  - `egs5run ucsource3 ucsource`

### 4. Check the sampled source electron spectrum in `egs5job.out`.

Example of the result.

Sampled source spectrum

Upper energy	particles/source		
	Gamma	Electron	pdf
.13650E-01 MeV--	0.0000	0.38520E-01	0.39108E-01
.27300E-01 MeV--	0.0000	0.37330E-01	0.37663E-01
.40950E-01 MeV--	0.0000	0.36750E-01	0.37516E-01
.54600E-01 MeV--	0.0000	0.37710E-01	0.37369E-01
.68250E-01 MeV--	0.0000	0.37060E-01	0.37173E-01
.81900E-01 MeV--	0.0000	0.36010E-01	0.36953E-01
.95550E-01 MeV--	0.0000	0.36990E-01	0.36732E-01
.10920 MeV--	0.0000	0.36450E-01	0.36487E-01
.12285 MeV--	0.0000	0.37110E-01	0.36218E-01
.13650 MeV--	0.0000	0.35230E-01	0.35900E-01
.15015 MeV--	0.0000	0.35310E-01	0.35581E-01
.16380 MeV--	0.0000	0.34350E-01	0.35239E-01
.17745 MeV--	0.0000	0.35240E-01	0.34822E-01
.19110 MeV--	0.0000	0.34100E-01	0.34381E-01
.20475 MeV--	0.0000	0.33260E-01	0.33892E-01
.21840 MeV--	0.0000	0.34070E-01	0.33328E-01
.23205 MeV--	0.0000	0.32810E-01	0.32692E-01
.24570 MeV--	0.0000	0.32090E-01	0.31982E-01
.25935 MeV--	0.0000	0.31810E-01	0.31198E-01
.27300 MeV--	0.0000	0.30760E-01	0.30316E-01
.28665 MeV--	0.0000	0.29310E-01	0.29337E-01
.30030 MeV--	0.0000	0.28920E-01	0.28259E-01
.31395 MeV--	0.0000	0.27620E-01	0.27084E-01
.32760 MeV--	0.0000	0.25490E-01	0.25786E-01
.34125 MeV--	0.0000	0.23880E-01	0.24415E-01
.35490 MeV--	0.0000	0.23650E-01	0.22896E-01
.36855 MeV--	0.0000	0.21360E-01	0.21305E-01
.38220 MeV--	0.0000	0.19210E-01	0.19615E-01
.39585 MeV--	0.0000	0.18200E-01	0.17852E-01
.40950 MeV--	0.0000	0.15610E-01	0.16015E-01
.42315 MeV--	0.0000	0.14610E-01	0.14130E-01
.43680 MeV--	0.0000	0.12170E-01	0.12195E-01
.45045 MeV--	0.0000	0.10670E-01	0.10285E-01
.46410 MeV--	0.0000	0.84100E-02	0.83995E-02
.47775 MeV--	0.0000	0.63300E-02	0.65628E-02
.49140 MeV--	0.0000	0.50100E-02	0.48487E-02
.50505 MeV--	0.0000	0.33400E-02	0.33059E-02
.51870 MeV--	0.0000	0.21200E-02	0.19835E-02
.53235 MeV--	0.0000	0.88000E-03	0.93055E-03
.54600 MeV--	0.0000	0.25000E-03	0.24488E-03

### 2.3.2 Using RADAR - The Decay Data

RADAR - The Decay Data (<http://www.doseinfo-radar.com/RADARDecay.html>) provide many  $\beta$ -ray spectra in the form of a probability density function (PDF) per energy interval. As an example,  $\beta$ -ray spectrum from Sr-90 is shown in Table 2.

Table 2  $\beta$ -ray spectrum from RADAR.

$E_{min}$	$E_{max}$	PDF
0.0000	0.0273	7.79E-02
0.0273	0.0546	7.60E-02
0.0546	0.0819	7.50E-02
0.0819	0.1092	7.40E-02
0.1092	0.1365	7.30E-02
0.1365	0.1638	7.17E-02
0.1638	0.1911	7.01E-02
0.1911	0.2184	6.80E-02
0.2184	0.2457	6.53E-02
0.2457	0.2730	6.19E-02
0.2730	0.3003	5.78E-02
0.3003	0.3276	5.27E-02
0.3276	0.3549	4.68E-02
0.3549	0.3822	4.01E-02
0.3822	0.4095	3.27E-02
0.4095	0.4368	2.48E-02
0.4368	0.4641	1.71E-02
0.4641	0.4914	9.75E-03
0.4914	0.5187	4.28E-03
0.5187	0.5460	1.01E-03

1. `cp ucsource3.f ucsource3_1.f`

2. Modify `ucsource3_1.f` as follows:

- Change local variable `,deltaes,emax` to `beint`.

Change

```
* deltaes,emax
```

to

```
* beint
```

- Add `iebeta` to local variable integer.

Change

```
* i,icases,idin,ie,ifti,ifto,ii,j,k,n,nd,ner,nsebin
```

to

```
* i,icases,idin,ie,ifti,ifto,ii,j,k,n,nd,ner,nsebin,ibeta
```

- Add character statement concerning source information.

Change

```
character*24 medarr(MXMED)
```

to

```
character*24 medarr(MXMED)
```

```
character*20 soinf
```

- Modify the `open` statement for the source data file.

Change

```
open(2,file='sr90beta.inp',status='unknown')
```

to

```
open(2,file='Sr-90.dat',status='unknown')
```

- `Sr-90beta.dat` is the data file including  $\beta$ -ray pdf per energy bin and is included in the distribution files.

```
Sr-90 Beta-
-1, 1.0000
0.0273,7.79E-02
0.0546,7.60E-02
0.0819,7.50E-02
0.1092,7.40E-02
0.1365,7.30E-02
0.1638,7.17E-02
0.1911,7.01E-02
0.2184,6.80E-02
0.2457,6.53E-02
0.2730,6.19E-02
0.3003,5.78E-02
0.3276,5.27E-02
0.3549,4.68E-02
0.3822,4.01E-02
0.4095,3.27E-02
0.4368,2.48E-02
0.4641,1.71E-02
0.4914,9.75E-03
0.5187,4.28E-03
0.5460,1.01E-03
```

-1 at the second line means electron decay. In the case of positron decay this number must be 1.

- Modify the statements to read data from the source data file and related treatments.

Change

```
read(2,*) emax ! Maximum beta-ray energy RADAR
read(2,*) nsebin ! Number of source energy bins
read(2,*) deltaes ! Source energy bin width in MeV
read(2,*) (espdf(i),i=1,nsebin)
!-----
! Calculate cdf from pdf
!-----
tnum=0.D0
do ie=1,nsebin
  esbin(ie)=(ie-1)*deltaes*emax
  tnum=tnum+espdf(ie)
end do

escdf(1)=0.0
do ie=2,nsebin
  escdf(ie)=escdf(ie-1)+espdf(ie-1)/tnum
end do
iqin=-1 ! Incident charge - electrons
ekein=esbin(nsebin) ! Maximum kinetic energy
```

to

```
nsebin=21 ! Number of bin for beta-ray spectrum
esbin(1)=0.d0
espdf(1)=0.d0
read(2,'(A20)') soinf ! Source information
read(2,*) ibeta, beint ! Charge of beta-ray, emitted beta per decay
do ie=2,nsebin
```

```

        read(2,*) esbin(ie),espdf(ie) ! Upper bin and pdf
    end do
!-----
! Calculate cdf from pdf
!-----
    do ie=2,nsebin
        escdf(ie)=escdf(ie-1)+espdf(ie)/tnum
    end do

    iqin=ibeta          ! Incident charge - electrons
    ekein=esbin(nsebin) ! Maximum kinetic energy

```

- Modify output statements of results as follows.

Change

```
write(6,170) esbin(ie),spg(ie),spe(ie),espdf(ie-1)/tnum
```

to

```
write(6,170) esbin(ie),spg(ie),spe(ie),espdf(ie)
```

3. Run `ucsource3_1.f` by `egs5run`.

- In the case of Linux or Cygwin  
Enter `ucsource3_1` as the user code.  
Enter `ucsource` as the file name for unit 4 and simply enter "return" for unit 25.  
Enter 1 for "Does this user code read from the terminal?".
- In the case of DOS  
`egs5run ucsource3_1 ucsource`

4. Check the sampled source electron spectrum in `egs5job.out`.

Example of the result.

Sampled source spectrum		particles/source		
Upper energy		Gamma	Electron	pdf
.27300E-01	MeV--	0.0000	0.77120E-01	0.77905E-01
.54600E-01	MeV--	0.0000	0.75250E-01	0.76005E-01
.81900E-01	MeV--	0.0000	0.74090E-01	0.75005E-01
.10920	MeV--	0.0000	0.74380E-01	0.74004E-01
.13650	MeV--	0.0000	0.73020E-01	0.73004E-01
.16380	MeV--	0.0000	0.70870E-01	0.71704E-01
.19110	MeV--	0.0000	0.69590E-01	0.70104E-01
.21840	MeV--	0.0000	0.68480E-01	0.68004E-01
.24570	MeV--	0.0000	0.65610E-01	0.65304E-01
.27300	MeV--	0.0000	0.62860E-01	0.61904E-01
.30030	MeV--	0.0000	0.58070E-01	0.57803E-01
.32760	MeV--	0.0000	0.53090E-01	0.52703E-01
.35490	MeV--	0.0000	0.47070E-01	0.46803E-01
.38220	MeV--	0.0000	0.39880E-01	0.40102E-01
.40950	MeV--	0.0000	0.33350E-01	0.32702E-01
.43680	MeV--	0.0000	0.24820E-01	0.24801E-01
.46410	MeV--	0.0000	0.17320E-01	0.17101E-01
.49140	MeV--	0.0000	0.97500E-02	0.97506E-02
.51870	MeV--	0.0000	0.43600E-02	0.42803E-02
.54600	MeV--	0.0000	0.10200E-02	0.10101E-02

### 3 Practice 2: Source Position

Suppose a source uniformly distributed inside an annular area between radius  $R_0(=1.5\text{cm})$  and  $R_1(=4.0\text{cm})$  in the X-Y plane.

### 3.1 Direct sampling

In this case, the PDF for the radius is

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

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

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

The radial position is obtained by solving

$$\eta = \int_{R_0}^r f(\xi)d\xi = \frac{r^2 - R_0^2}{R_1^2 - R_0^2}$$

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

Having thus located radius  $R$ , we note that  $f(\phi)d\phi = d\phi/2\pi$ , so that the next random number determines  $\phi$  by

$$\eta_2 = F(\phi) = \int_{-\pi}^{\phi} f(\phi)d\phi = (\phi + \pi)/2\pi \quad (1)$$

and  $\phi = \pi(2\eta_2 - 1)$ .

$x$  and  $y$  are calculated from

$$x = r \cos \phi, \quad (2)$$

$$y = r \sin \phi. \quad (3)$$

1. `cp ucsource.f ucsource4.f`

2. Modify `ucsource4.f` as follows:

- Add `r02,r12,phai` to local variable.

Change

```
* esbin(1),spg(1),spe(1)
```

to

```
* esbin(1),spg(1),spe(1),r02,r12,phai,rr0
```

- Insert statements to define `r02,r12`.

Change

```
wtin=1.0          ! Weight = 1 since no variance reduction used
```

to

```
wtin=1.0          ! Weight = 1 since no variance reduction used
r02=1.5*1.5
r12=4.0*4.0
```

- Insert statements to sample a source position.

Change

```
! -----
! Determine position
! -----
```



```

to
!
! -----
! Determine position
! -----
call randomset(rnnow)
rr0=sqrt(r02+rnnow*(r12-r02))
call RANDOMSET(rnnow)
phai=PI*(2.0*rnnow-1.0)
xin=rr0*cos(phai)
yin=rr0*sin(phai)

```

3. Run `ucsource4.f` by `egs5run`.

- In the case of Linux or Cygwin  
Enter `ucsource4` as the user code.  
Enter `ucsource` as the file name for unit 4 and simply enter "return" for unit 25.  
Enter 1 for "Does this user code read from the terminal?".
- In the case of DOS  
`egs5run ucsource4 ucsource`

4. Check the trajectories using CGview.

- Select X-Y display mode.
- Bend axis a little and confirm that photons are emitted from the annulus of radii between 1.5 and 4.0.

### 3.2 Rejection method

A source position  $(x, y)$  can be sampled as follows using the "rejection" method.

In this technique, a point is chosen randomly within the square  $-1.0 \leq x \leq 1.0$ ;  $-1.0 \leq y \leq 1.0$ . If this point satisfies  $R_0/R_1 \leq r = \sqrt{x^2 + y^2} \leq 1.0$ ,  $x * R_1$  and  $y * R_1$  is accepted as source position. If this point does not satisfy this condition, this point is rejected and new point is sampled again.

1. `cp ucsource4.f ucsource5.f`

2. Modify `ucsource5.f` as follows:

- Modify local variable.  
Change  
`* esbin(1),spg(1),spe(1),r02,r12,phai,rr0`

to

```
* esbin(1),spg(1),spe(1),r0,r1,rr0
```

- Modify statements related to radius of circles.

Change

```
r02=1.5*1.5
r12=4.0*4.0
```

to

```
r0=1.5
r1=4.0
```

- Modify the position sampling method.

Change

```

!           -----
!           Determine position
!           -----
           call randomset(rnnow)
           rr0=sqrt(r02+rnnow*(r12-r02))
           call RANDOMSET(rnnow)
           phai=PI*(2.0*rnnow-1.0)
           xin=rr0*cos(phai)
           yin=rr0*sin(phai)

to

!           -----
!           Determine position
!           -----
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

```

3. Run `ucsource5.f` by `egs5run`.

- In the case of Linux or Cygwin  
Enter `ucsource5` as the user code.  
Enter `ucsource` as the file name for unit 4 and simply enter "return" for unit 25.  
Enter 1 for "Does this user code read from the terminal?".
- In the case of DOS  
`egs5run ucsource5 ucsource`

4. Check the trajectories using CGview.

- Select X-Y display mode.
- Bend axis a little and confirm that photons are emitted from the annulus of radii between 1.5 and 4.0.

## 4 Practice 4: Source Direction

### 4.1 Direct sampling

Pdf of a directional cosine for Z-direction  $w$  is given by

$$f(\theta)d\theta = c \times 2\pi \sin \theta d\theta \quad (0 \leq \theta \leq \pi)$$

By setting

$$w = \cos \theta,$$

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

and

$$\int_1^{-1} g(w)dw = -c \times 2\pi \times (-2) = 1.$$

Therefore,

$$c = \frac{1}{4\pi} \rightarrow g(w)dw = -\frac{1}{2}dw.$$

w can be determined by solving the following equation.

$$\eta = \int_1^w g(w)dw = \frac{1}{2}(1-w) \rightarrow w = 1 - 2\eta.$$

When the isotropic source for  $2\pi$  region in this case, w can be determined by following way.

$$\int_1^0 g(w)dw = -c \times 2\pi \times (-1) = 1$$

$$c = \frac{1}{2\pi} \rightarrow g(w)dw = -dw.$$

$$\eta = \int_1^w g(w)dw = w \rightarrow w = 1 - \eta$$

$1 - \eta$  and  $\eta$  are equivalent and, therefore,  $w = \eta$  is usually used.

After sampling  $w$ ,  $\phi$  is sampled by

$$\phi = \pi(2\eta_2 - 1). \quad (4)$$

$u$  and  $v$  are given by

$$\sin \theta = \sqrt{1 - w^2}, \quad (5)$$

$$u = \sin \theta \cos \phi, \quad (6)$$

$$v = \sin \theta \sin \phi. \quad (7)$$

1. `cp ucsource.f ucsource6.f`

2. Modify `ucsource6.f`.

- Add `,phai,rr0` to local variable.

Change

```

      real*8
      * availke,tnum,wtin,wtsum,xi0,yi0,zi0,
      * esbin(1),spg(1),spe(1)

```

! Local variables

to

```

      real*8
      * availke,tnum,wtin,wtsum,xi0,yi0,zi0,
      * esbin(1),spg(1),spe(1),phai,rr0

```

! Local variables

- Insert the direction sampling routine.

Change

```

!      -----
!      Determine source direction
!      -----

```

to

```

! -----
! Determine source direction
! -----
call randomset(rnnow)
win=rnnow
call randomset(rnnow)
phai=PI*(2.0*rnnow-1.0)
uin=dsqrt(1.0-win*win)*cos(phai)
vin=dsqrt(1.0-win*win)*sin(phai)

```

- Normalization of directional cosines.
  - Sampled directional cosines must be normalized.  
( $xin*xin+yin*yin+win*win=1.0$ ).

In the sampled procedure above, the directional cosines are normalized properly. If user samples directional cosines by his/her own way, it is necessary to check whether they are normalized properly or not. Following statements must be inserted before `call shower`.

```

! -----
! Verify the normalization of source direction cosines
! -----
if(abs(uin*uin+vin*vin+win*win-1.0).gt.1.e-6) then
  write(6,fmt="( ' Following source direction cosines are not',
1    ' normalized. ',3e12.5)")uin,vin,win
  stop
end if

```

3. Run `ucsource6.f` by `egs5run`.

- In the case of Linux or Cygwin
  - Enter `ucsource6` as the user code.
  - Enter `ucsource` as the file name for unit 4 and simply enter "return" for unit 25.
  - Enter 1 for "Does this user code read from the terminal?".
- In the case of DOS
  - `egs5run ucsource6 ucsource`

4. Check the trajectories using CGview.

- Select X-Y display mode.
- Confirm that photons are produced isotropically within  $2\pi$  region.

## 4.2 Rejection method

A point  $(x_i, y_i, z_i)$  is chosen randomly within the box  $-1.0 \leq x \leq 1.0$ ;  $-1.0 \leq y \leq 1.0$ ;  $-1.0 \leq z \leq 1.0$ . If this point lies within a sphere with unit radius,

$$R = \sqrt{x_1^2 + y_1^2 + z_1^2} \leq 1, \quad (8)$$

the point is accepted and  $u, v, w$  are determined by

$$u = x_1/R; \quad v = y_1/R; \quad w = z_1/R. \quad (9)$$

Otherwise, the point is rejected and sample a point again.

1. `cp ucsource6.f ucsource7.f`
2. Modify `ucsource7.f`.

- Modify the direction sampling routine.

Change

```
! -----
! Determine source direction
! -----
call randomset(rnnow)
win=rnnow
call randomset(rnnow)
phai=PI*(2.0*rnnow-1.0)
uin=dsqrt(1.0-win*win)*cos(phai)
vin=dsqrt(1.0-win*win)*sin(phai)
```

to

```
! -----
! Determine source direction
! -----
1300 call randomset(rnnow)
      zi0=rnnow
      call randomset(rnnow)
      xi0=2.0*rnnow-1.0
      call randomset(rnnow)
      yi0=2.0*rnnow-1.0
      rr0=dsqrt(xi0*xi0+yi0*yi0+zi0*zi0)
      if(rr0.gt.1.0) go to 1300
      win = zi0/rr0
      uin = xi0/rr0
      vin = yi0/rr0
```

3. Run `ucsource7.f` by `egs5run`.

- In the case of Linux or Cygwin
  - Enter `ucsource7` as the user code.
  - Enter `ucsource` as the file name for unit 4 and simply enter "return" for unit 25.
  - Enter 1 for "Does this user code read from the terminal?".
- In the case of DOS
  - `egs5run ucsource7 ucsource`

4. Check the trajectories using CGview and confirm that photons are produced isotropically within  $2\pi$  region.

## Appendix: Full listings of ucucsource.f

```

*****
***** KEK, High Energy Accelerator Research *
***** Organization *
*** u c s o u r c e ***** *
***** EGS5.0 USER CODE - 28 Jul 2012/1430 *
***** *
* This is a general User Code based on the cg geometry scheme. *
*****

```

```

PROGRAMMERS: H. Hirayama *
              Applied Research Laboratory *
              KEK, High Energy Accelerator Research Organization *
              1-1, Oho, Tsukuba, Ibaraki, 305-0801 *
              Japan *

```

```

E-mail:      hideo.hirayama@kek.jp *
Telephone:   +81-29-864-5451 *
Fax:         +81-29-864-4051 *

```

```

Y. Namito *
Radiation Science Center *
Applied Research Laboratory *
KEK, High Energy Accelerator Research Organization *
1-1, Oho, Tsukuba, Ibaraki, 305-0801 *
Japan *

```

```

E-mail:      yoshihito.namito@kek.jp *
Telephone:   +81-29-864-5489 *
Fax:         +81-29-864-1993 *

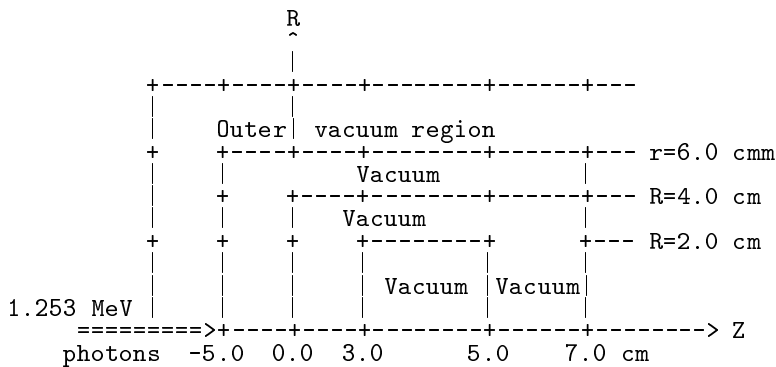
```

```

*****
***** The ucsorce.f User Code requires a cg-input file only *
***** (e.g., ucsorce.data). *
***** The following shows the geometry for ucsorce.data. *
***** Input data for CG geometry must be written at the top of data-input *
***** file together with material assignment to each region. Cg-data can *
***** be checked by CGview. *
***** This user code to understand source routine. *
***** Use Ranlux random number generator. *
*****

```

-----  
cg Geometry (ucsorce)  
-----



```

*****
23456789|123456789|123456789|123456789|123456789|123456789|123456789|12

```

-----  
main code  
-----

Step 1: Initialization

```
implicit none
```

```
-----  
EGS5 COMMONs  
-----
```

```
include 'include/egs5_h.f' ! Main EGS "header" file
```

```

include 'include/egs5_bounds.f'
include 'include/egs5_brempr.f'
include 'include/egs5_edge.f'
include 'include/egs5_media.f'
include 'include/egs5_misc.f'
include 'include/egs5_thresh.f'
include 'include/egs5_uphiot.f'
include 'include/egs5_useful.f'
include 'include/egs5_usersc.f'
include 'include/egs5_userxt.f'
include 'include/randomm.f'

!-----
! Auxiliary-code COMMONs
!-----
include 'auxcommons/aux_h.f' ! Auxiliary-code "header" file

include 'auxcommons/edata.f'
include 'auxcommons/etaly1.f'
include 'auxcommons/instuf.f'
include 'auxcommons/lines.f'
include 'auxcommons/nfac.f'
include 'auxcommons/watch.f'

!-----
! cg related COMMONs
!-----
include 'auxcommons/geom_common.f' ! geom-common file
integer irinn

common/totals/ ! Variables to score
* maxpict
integer maxpict

!**** real*8 ! Arguments
real*8 totke
real*8 rnnow,etot

real*8 ! Local variables
* availke,tnum,wtin,wtsum,xi0,yi0,zi0,
* esbin(1),spg(1),spe(1)

real
* tarray(2),tt,tt0,tt1,cputime,etime

integer
* i,icases,idin,ie,ifti,ifto,ii,j,k,n,nd,ner,nsebin

character*24 medarr(MXMED)

!-----
! Open files
!-----
Units 7-26 are used in pegs and closed. It is better not
to use as output file. If they are used, they must be opened
after call pegs5. Unit for pict must be 39.

open(6,FILE='egs5job.out',STATUS='unknown')
open(4,FILE='egs5job.inp',STATUS='old')
open(39,FILE='egs5job.pic',STATUS='unknown')

!=====
! call counters_out(0)
!=====

!-----
! Step 2: pegs5-call
!-----
!=====
! call block_set ! Initialize some general variables
!=====

!-----
! Define media before calling PEGS5
!-----

nmed=1
if(nmed.gt.MXMED) then

```



```

    write(6,'(A,I4,A,I4,A/A)')
*      ' nmed (' ,nmed,' ) larger than MXMED (' ,MXMED,' )',
*      ' MXMED in include/egs5_h.f must be increased.'
    stop
end if

medarr(1)='NAI'

do j=1,nmed
  do i=1,24
    media(i,j)=medarr(j)(i:i)
  end do
end do

chard(1) = 1.0d0      ! optional, but recommended to invoke
                    ! automatic step-size control

write(6,fmt="( 'chard = ',5e12.5)" (chard(j),j=1,nmed)

! -----
! Run KEK PEGS5 before calling HATCH
! -----
100 write(6,100)
   FORMAT('PEGS5-call comes next'/)

! =====
! call pegs5
! =====

!-----
! Step 3: Pre-hatch-call-initialization
!-----
!-----
! Initialize cg related parameter
!-----
npreci=3      ! PICT data mode for CGView in free format

ifti = 4      ! Input unit number for cg-data
ifto = 39     ! Output unit number for PICT

write(6,fmt="( ' CG data' )"
call geomgt(ifti,6) ! Read in CG data
write(6,fmt="( ' End of CG data',/ )"

if(npreci.eq.3) write(ifto,fmt="( 'CSTA-FREE-TIME' )"
if(npreci.eq.2) write(ifto,fmt="( 'CSTA-TIME' )"

rewind ifti
call geomgt(ifti,ifto)! Dummy call to write geom info for ifto
write(ifto,110)
110 FORMAT('CEND')

!-----
! Get nreg from cg input data
!-----
nreg=izonin

! Read material for each refion from egs5job.data
read(4,*) (med(i),i=1,nreg)

!-----
! Random number seeds. Must be defined before call hatch
! or defaults will be used. inseed (1- 2^31)
!-----
luxlev = 1
inseed=1
write(6,120) inseed
120 FORMAT(/,' inseed=',I12,5X,
*      '(seed for generating unique sequences of Ranlux)')

! =====
! call rluxinit ! Initialize the Ranlux random-number generator
! =====

!-----
! Step 4: Determination-of-incident-particle-parameters
!-----
! Define initial variables for incident particle normally incident
! on the slab
nsebin=1
iqin=0      ! Incident charge - photons
ekein=1.253 ! Kinetic energy

```

```

xin=0.0          ! Source position
yin=0.0
zin=-5.0
uin=0.0          ! Moving along z axis
vin=0.0
win=1.0
irin=0           ! Starting region (0: Automatic search in CG)
wtin=1.0         ! Weight = 1 since no variance reduction used

```

```

-----
! Step 5: hatch-call
-----

```

```

emaxe = 0.D0 ! dummy value to extract min(UE,UP+RM).

write(6,130)
130 format(/,' Call hatch to get cross-section data')

! -----
! Open files (before HATCH call)
! -----
open(UNIT=KMPI,FILE='pgs5job.pegs5dat',STATUS='old')
open(UNIT=KMPO,FILE='egs5job.dummy',STATUS='unknown')

write(6,140)
140 FORMAT(/,' HATCH-call comes next',/)

! =====
! call hatch
! =====

! -----
! Close files (after HATCH call)
! -----
close(UNIT=KMPI)
close(UNIT=KMPO)

write(39,fmt="( 'MSTA' )")
write(39,fmt="(i4)") nreg
write(39,fmt="(15i4)") (med(i),i=1,nreg)
write(39,fmt="( 'MEND' )")

```

```

-----
! Step 6: Initialization-for-howfar
-----

```

```

-----
! Step 7: Initialization-for-ausgab
-----

```

```

ncount = 0
ilines = 0
nwrite = 10
nlines = 10
idin = -1
totke = 0.
wtsum = 0.

! =====
! call ecnsv1(0,nreg,totke)
! call ntally(0,nreg)
! =====

esbin(1)=ekein
Zero the variables
do j=1,nsebin
  spg(j)=0.D0
  spe(j)=0.D0
end do

! Set histories and histories to write trajectories
ncases=10000
! Set maximum number for pict
maxpict=500

tt=etime(tarray)
tt0=tarray(1)

```

```

-----
! Step 8: Shower-call
-----

```

```

Write batch number
write(39,fmt="( '0 1' )")

```

```

do i=1,ncases
! -----
! Start of batch -loop
! -----

wtin = 1.0

wtsum = wtsum + wtin          ! Keep running sum of weights

! -----
! Determine source energy
! -----
ekein = ekein
spg(1)=spg(1)+1.0

etot = ekein + iabs(iqin)*RM          ! Incident total energy (MeV)
availke = etot + iqin*RM             ! Available K.E. (MeV) in system
totke = totke + availke              ! Keep running sum of KE

! -----
! Determine source direction
! -----

! -----
! Determine source position
! -----

! -----
! Get source region from cg input data
! -----

if(irin.le.0.or.irin.gt.nreg) then
  call srzone(xin,yin,zin,iqin+2,0,irinn)
  if(irinn.le.0.or.irinn.ge.nreg) then
    write(6,fmt="( ' Stopped in MAIN. irinn = ',i5)")irinn
    stop
  end if
  call rstnxt(iqin+2,0,irinn)
else
  irinn=irin
end if

! -----
! Compare maximum energy of material data and incident energy
! -----
if(etot+(1-iabs(iqin))*RM.gt.emaxe) then
  write(6,fmt="( ' Stopped in MAIN. ',
1    ' (Incident kinetic energy + RM) > min(UE,UP+RM). ')")
  stop
end if

! -----
! Verify the normalization of source direction cosines
! -----
if(abs(uin*uin+vin*vin+win*win-1.0).gt.1.e-6) then
  write(6,fmt="( ' Following source direction cosines are not',
1    ' normalized. ',3e12.5)")uin,vin,win
  stop
end if

! -----
! =====
! call shower (iqin,etot,xin,yin,zin,uin,vin,win,irinn,wtin)
! =====

ncount = ncount + 1          ! Count total number of actual cases

end do
! -----
! End of batch loop
! -----

call plotxyz(99,0,0,0.D0,0.D0,0.D0,0.D0,0,0,0.D0,0.D0)

write(39,fmt="( '9' )")          ! Set end of batch for CG View

tt=etime(tarray)
tt1=tarray(1)
cputime=tt1-tt0
write(6,150) cputime
150  format(' Elapsed Time (sec)=',G15.5)

! -----
! Step 9: Output-of-results

```

```

!-----
!
! Source spectrum. Incident particle spectrum to detector.
!-----
write(6,160)
160  FORMAT(/' Sampled source spectrum'/
*      30X,'particles/source'/
*      ' Upper energy',11X,' Gamma',14X,' Electron',
*      11X,' pdf')

do ie=1,nsebin

!-----
! Gamma spectrum per source
!-----
spg(ie)=spg(ie)/ncount
!-----
! Electron spectrum per source
!-----
spe(ie)=spe(ie)/ncount

write(6,170) esbin(ie),spg(ie),spe(ie)
170  FORMAT(G10.5,' MeV--',8X,G12.5,8X,G12.5)
end do

!
!=====
call counters_out(1)
!=====

stop
end

!-----last line of main code-----
!-----ausgab.f-----
! Version: 030831-1300
! Reference: SLAC-265 (p.19-20, Appendix 2)
!-----
!23456789|123456789|123456789|123456789|123456789|123456789|123456789|12
!-----
! Required subroutine for use with the EGS5 Code System
! A AUSGAB to: produce trajectory data for imode=0
!-----

subroutine ausgab(iarg)
implicit none
include 'include/egs5_h.f'           ! Main EGS "header" file
include 'include/egs5_epcont.f'     ! COMMONs required by EGS5 code
include 'include/egs5_misc.f'
include 'include/egs5_stack.f'
include 'include/egs5_useful.f'

include 'auxcommons/aux_h.f'       ! Auxiliary-code "header" file
include 'auxcommons/lines.f'       ! Auxiliary-code COMMONs

common/totals/                     ! Variables to score
* maxpict
integer maxpict

integer                             ! Arguments
* iarg

real*8                               ! Local variables
* edepwt

integer
* ie,iql,irl

!-----
! Set some local variables
!-----
irl = ir(np)

```

```

    iql = iq(np)
    edepwt = edep*wt(np)
!-----
! Output particle information for plot
!-----
    if (ncount.le.maxpict) then
        call plotxyz(iarg,np,iq(np),x(np),y(np),z(np),e(np),ir(np),
*         wt(np),time(np))
    end if

    return

end

!-----last line of ausgab.f-----
!-----howfar.f-----
! Version: 070627-1600
! Reference: T. Torii and T. Sugita, "Development of PRESTA-CG
! Incorporating Combinatorial Geometry in EGS4/PRESTA", JNC TN1410 2002-201,
! Japan Nuclear Cycle Development Institute (2002).
! Improved version is provided by T. Sugita. 7/27/2004
!-----
|23456789|123456789|123456789|123456789|123456789|123456789|123456789|12
!-----
! Required (geometry) subroutine for use with the EGS5 Code System
!-----
! This is a CG-HOWFAR.
!-----

subroutine howfar
implicit none
c
include 'include/egs5_h.f' ! Main EGS "header" file
include 'include/egs5_epcont.f' ! COMMONs required by EGS5 code
include 'include/egs5_stack.f'
include 'auxcommons/geom_common.f' ! geom-common file
c
c
integer i,j,jjj,ir_np,nozone,jty,kno
integer irnear,irnext,irlold,irlfg,itvlf,ihitcg
double precision xidd,yidd,zidd,x_np,y_np,z_np,u_np,v_np,w_np
double precision tval,tval0,tval00,tval10,tvalmn,delhow
double precision atvalttmp
integer iq_np
c
ir_np = ir(np)
iq_np = iq(np) + 2
c
if(ir_np.le.0) then
    write(6,*) 'Stopped in howfar with ir(np) <=0'
    stop
end if
c
if(ir_np.gt.izonin) then
    write(6,*) 'Stopped in howfar with ir(np) > izonin'
    stop
end if
c
if(ir_np.EQ.izonin) then
    idisc=1
    return
end if
c
tval=1.d+30
itvalm=0
c
body check
u_np=u(np)
v_np=v(np)
w_np=w(np)
x_np=x(np)
y_np=y(np)
z_np=z(np)
c
do i=1,nbody(ir_np)
    nozone=ABS(nbzone(i,ir_np))
    jty=itblty(nozone)

```

```

      kno=itblno(nozone)
c   rpp check
      if(jty.eq.ityknd(1)) then
        if(kno.le.0.or.kno.gt.irppin) go to 190
        call rppcg1(kno,x_np,y_np,z_np,u_np,v_np,w_np)
c   sph check
      elseif(jty.eq.ityknd(2)) then
        if(kno.le.0.or.kno.gt.isphin) go to 190
        call sphcg1(kno,x_np,y_np,z_np,u_np,v_np,w_np)
c   rcc check
      elseif(jty.eq.ityknd(3)) then
        if(kno.le.0.or.kno.gt.irccin) go to 190
        call rcccg1(kno,x_np,y_np,z_np,u_np,v_np,w_np)
c   trc check
      elseif(jty.eq.ityknd(4)) then
        if(kno.le.0.or.kno.gt.itrcin) go to 190
        call trccg1(kno,x_np,y_np,z_np,u_np,v_np,w_np)
c   tor check
      elseif(jty.eq.ityknd(5)) then
        if(kno.le.0.or.kno.gt.itorin) go to 190
        call torcg1(kno,x_np,y_np,z_np,u_np,v_np,w_np)
c   rec check
      elseif(jty.eq.ityknd(6)) then
        if(kno.le.0.or.kno.gt.irecin) go to 190
        call reccg1(kno,x_np,y_np,z_np,u_np,v_np,w_np)
c   ell check
      elseif(jty.eq.ityknd(7)) then
        if(kno.le.0.or.kno.gt.iellin) go to 190
        call ellcg1(kno,x_np,y_np,z_np,u_np,v_np,w_np)
c   wed check
      elseif(jty.eq.ityknd(8)) then
        if(kno.le.0.or.kno.gt.iwedin) go to 190
        call wedcg1(kno,x_np,y_np,z_np,u_np,v_np,w_np)
c   box check
      elseif(jty.eq.ityknd(9)) then
        if(kno.le.0.or.kno.gt.iboxin) go to 190
        call boxcg1(kno,x_np,y_np,z_np,u_np,v_np,w_np)
c   arb check
      elseif(jty.eq.ityknd(10)) then
        if(kno.le.0.or.kno.gt.iarbin) go to 190
        call arbcg1(kno,x_np,y_np,z_np,u_np,v_np,w_np)
c   hex check
      elseif(jty.eq.ityknd(11)) then
        if(kno.le.0.or.kno.gt.ihexin) go to 190
        call hexcg1(kno,x_np,y_np,z_np,u_np,v_np,w_np)
c   haf check
      elseif(jty.eq.ityknd(12)) then
        if(kno.le.0.or.kno.gt.ihafin) go to 190
        call hafcg1(kno,x_np,y_np,z_np,u_np,v_np,w_np)
c   tec check
      elseif(jty.eq.ityknd(13)) then
        if(kno.le.0.or.kno.gt.itecin) go to 190
        call teccg1(kno,x_np,y_np,z_np,u_np,v_np,w_np)
c   gel check
      elseif(jty.eq.ityknd(14)) then
        if(kno.le.0.or.kno.gt.igelin) go to 190
        call gelcg1(kno,x_np,y_np,z_np,u_np,v_np,w_np)
c
c**** add new geometry in here
c
      end if
190  continue
      end do
c
      irnear=ir_np
      if(itvalm.eq.0) then
        tval0=cgeps1
        xidd=x_np+tval0*u_np
        yidd=y_np+tval0*v_np
        zidd=z_np+tval0*w_np
310  continue
        if(x_np.ne.xidd.or.y_np.ne.yidd.or.z_np.ne.zidd) goto 320
        tval0=tval0*10.d0
        xidd=x_np+tval0*u_np
        yidd=y_np+tval0*v_np
        zidd=z_np+tval0*w_np

```

```

    go to 310
    continue
320  write(*,*) 'srzone:1'
    call srzone(xidd,yidd,zidd,iq_np,ir_np,irnext)
c
    if(irnext.ne.ir_np) then
        tval=0.0d0
        irnear=irnext
    else
        tval00=0.0d0
        tval10=10.0d0*tval0
        irlold=ir_np
        irlfg=0
330  continue
        if(irlfg.eq.1) go to 340
            tval00=tval00+tval10
            if(tval00.gt.1.0d+06) then
                write(6,9000) iq(np),ir(np),x(np),y(np),z(np),
&                               u(np),v(np),w(np),tval00
9000 format(' TVAL00 ERROR : iq,ir,x,y,z,u,v,w,tval=',
&           2I3,1P7E12.5)
                stop
            end if
            xidd=x_np+tval00*u_np
            yidd=y_np+tval00*v_np
            zidd=z_np+tval00*w_np
            call srzold(xidd,yidd,zidd,irlold,irlfg)
            go to 330
340  continue
c
        tval=tval00
        do j=1,10
            xidd=x_np+tval00*u_np
            yidd=y_np+tval00*v_np
            zidd=z_np+tval00*w_np
c
            write(*,*) 'srzone:2'
            call srzone(xidd,yidd,zidd,iq_np,irlold,irnext)
            if(irnext.ne.irlold) then
                tval=tval00
                irnear=irnext
            end if
            tval00=tval00-tval0
        end do
        if(ir_np.eq.irnear) then
            write(0,*) 'ir(np),tval=',ir_np,tval
        end if
    end if
else
    do j=1,itvalm-1
        do i=j+1,itvalm
            if(atval(i).lt.atval(j)) then
                atvaltmp=atval(i)
                atval(i)=atval(j)
                atval(j)=atvaltmp
            endif
        enddo
    enddo
    itvlf=0
    tvalmn=tval
    do jjj=1,itvalm
        if(tvalmn.gt.atval(jjj)) then
            tvalmn=atval(jjj)
        end if
        delhow=cgeps2
        tval0=atval(jjj)+delhow
        xidd=x_np+tval0*u_np
        yidd=y_np+tval0*v_np
        zidd=z_np+tval0*w_np
410  continue
        if(x_np.ne.xidd.or.y_np.ne.yidd.or.z_np.ne.zidd) go to 420
            delhow=delhow*10.d0
            tval0=atval(jjj)+delhow
            xidd=x_np+tval0*u_np
            yidd=y_np+tval0*v_np
            zidd=z_np+tval0*w_np
        go to 410
420  continue
c
        write(*,*) 'srzone:3'
        call srzone(xidd,yidd,zidd,iq_np,ir_np,irnext)

```

```

        if((irnext.ne.ir_np.or.atval(jjj).ge.1.).and.
&         tval.gt.atval(jjj)) THEN
            tval=atval(jjj)
            irnear=irnext
            itvlf=1
            goto 425
        end if
    end do
425  continue
    if(itvlf.eq.0) then
        tval0=cgmnst
        xidd=x_np+tval0*u_np
        yidd=y_np+tval0*v_np
        zidd=z_np+tval0*w_np
430  continue
        if(x_np.ne.xidd.or.y_np.ne.yidd.or.z_np.ne.zidd) go to 440
            tval0=tval0*10.d0
            xidd=x_np+tval0*u_np
            yidd=y_np+tval0*v_np
            zidd=z_np+tval0*w_np
            go to 430
440  continue
        if(tvalmn.gt.tval0) then
            tval=tvalmn
        else
            tval=tval0
        end if
    end if
    end if
    ihitcg=0
    if(tval.le.ustep) then
        ustep=tval
        ihitcg=1
    end if
    if(ihitcg.eq.1) THEN
        if(irnear.eq.0) THEN
            write(6,9200) iq(np),ir(np),x(np),y(np),z(np),
&                u(np),v(np),w(np),tval
9200 format(' TVAL ERROR : iq,ir,x,y,z,u,v,w,tval=',2I3,1P7E12.5)
            idisc=1
            itverr=itverr+1
            if(itverr.ge.100) then
                stop
            end if
            return
        end if
        irnew=irnear
        if(irnew.ne.ir_np) then
            call rstnxt(iq_np,ir_np,irnew)
        endif
    end if
    return
end
!-----last line of subroutine howfar-----

```