

egs5 サンプルプログラム (uccg_nai.f)
NaI 検出器の応答計算 (cg Version)
(July 28, 2004, Draft)

平山 英夫、波戸 芳仁

〒 305-0801 茨城県つくば市大穂 1 - 1
高エネルギー加速器研究機構

Contents

1. PRESTA-CG	1
1.1. Body の定義	1
1.2. リージョンの定義	1
1.3. リージョン定義の例	2
2. サンプルプログラム uccg_nai.f の概要	4
2.1. CG 入力データ	4
3. ユーザーコードの内容	5
3.1. メインプログラム	5
3.1.1. include 文及び型式宣言:	5
3.1.2. open 文:	6
3.1.3. Subroutine getcg の call:	6
3.1.4. 各種パラメータの設定とデータの初期化:	7
3.1.5. 輸送計算:	8
3.1.6. 統計誤差:	9
3.1.7. 計算結果の出力:	10
3.2. subroutine getcg	11
3.3. subroutine ausgab	12
3.4. subroutine howfar	13
4. ucrz_nai.f と uccg_nai.f の計算速度の比較	13
5. 実習課題	14
5.1. 実習課題 1 : NaI 検出器の計算	14
5.2. 実習課題 2 : Ge 検出器の計算	14
5.3. 実習課題 3 : 空気電離箱の計算	14
6. 実習課題の解答例	15
6.1. 実習課題 1	15
6.2. 実習課題 2	15
6.3. 実習課題 3	16

1. PRESTA-CG

1.1. Body の定義

PRESTA-CG*では、以下のような Body を使用する事ができる。

1. 直方体 (RPP)
x-, y- と z-方向の最小値及び最大値で、定義する。各面は、いずれかの軸と平行である。
2. 球 (SPH)
球の中心を示すベクトル V と半径で定義する。
3. 円筒 (RCC)
円筒の底面の中心を示すベクトル V と、中心からの高さベクトル H 及び円筒の半径で定義する。
4. 円錐台 (TRC)
円錐の底面の中心を示すベクトル V 、底面中心からの上面中心への高さベクトル H 、及び底面と上面のそれぞれの半径 $R1$ 及び $R2$ で定義する。
5. トーラス (TOR)
いずれかの軸に平行なトーラスの中心を示すベクトル V 、トーラス中心から、チューブの中心までの距離 $R1$ 、チューブの半径 $R2$ 及びトーラスの方向を示す番号、(n: x/y/z = 1/2/3) で定義する。更に、トーラスの始まりの角度 $\theta1$ と終わりの角度 $\theta2$ を指定する。トーラス全体を使用する場合には、 $\theta1=0$ 、及び $\theta2=2\pi$ とする。

Table 1 Data required to described each bosy type.

Body Type	Inp. #	Real Data defining Paticular Body					
RPP	#	Xmin	Xmax	Ymin	Ymax	Zmin	Zmax
SPH	#	Vx	Vy	Vz	R		
RCC	#	Vx	Vy	Vz	Hx	Hy	Hz
		R					
TRC	#	Vx	Vy	Vz	Hx	Hy	Hz
		R1	R2				
TOR	#	Vx	Vy	Vz	R1	R2	
		$\theta1$	$\theta2$	n			

1.2. リージョンの定義

各リージョンは、body の組み合わせにより定義する。組み合わせには、特別な記号、+、- 及び OR が使われる。

+ 記号の後に body 番号が書かれた場合には、body に含まれる全ての領域がリージョンとなる。一方、- 記号の後に body 番号が書かれた場合には、body の外側の全ての領域がリージョンとなる。Body 番号が OR 記号の後に書かれた場合は、OR 記号は結合記号として使用される。リージョンが、OR 記号で結合したサブリージョンの組み合わせで定義される場合もある。2つ以上の OR 記号が使われる場合、OR の機能は、OR 記号の間及び OR 記号からリージョン定義の行の最後までに含まれる全ての body 番号に、+ や - 記号に関係なく適用される。

*JNC TN1410 2002-001 by T. Trii and T. Sugita[1] の Appendix A を参照

1.3. リージョン定義の例

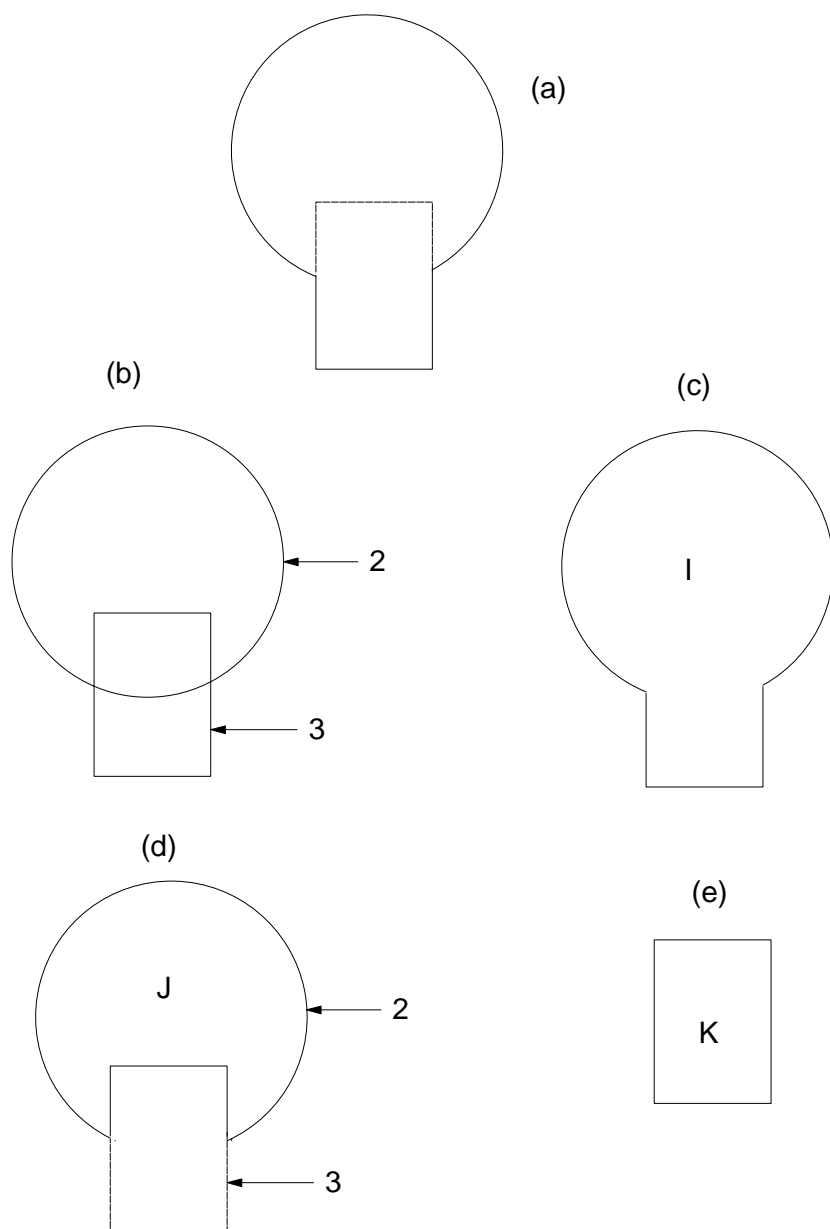


Figure 1: Examples of Combinatorial Geometry Method.

第1図に示すような、球 (body 2) に円筒 (body 3) が挿入し様な体系を考える。もし、球と円筒の物質が同じであれば、リージョン I (図 1c) の様に一つのリージョンとする事ができる。リージョン I は、

$$I = +2OR + 3.$$

と記述する。これは、リージョン I が、body 2 か body 3 のどちらかに属する領域であることを意味している。

球と円筒が異なった物質の場合、円筒部を除外した球には、円筒部のリージョン番号 (K) と異なったリージョン番号を付ける (例えば J)。

リージョン J (図 1d) は、

$$J = +2 - 3.$$

と記述する。これは、body 2 に属するが、body 3 に属しない領域を意味する。

リージョン K (図 2e) は、単に

$$K = +3.$$

と記述する。これは、body 3 の属する領域を意味する。

2 つ以上の body を組み合わせる場合には、+、- や OR 記号を含む長い記述となる。しかしながら、形状中の全ての点は、どれか一つのリージョンとして定義される様にしなければならない。

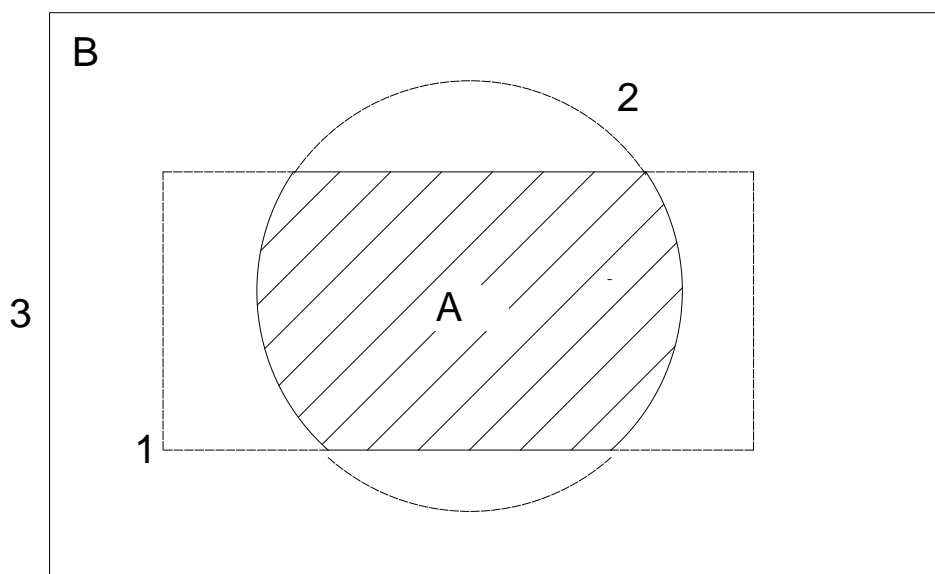


Figure 2: Use of OR operator.

OR 記号を使ったもっと複雑な例として、第 2 図の斜線部の領域 A と斜線を引いていない領域 B を考える。これらのリージョンは、2 つの直方体 (body 1 と 3) と、一つの円筒 (body 2) で記述される。それぞれのリージョンは、

$$A = +1 + 2$$

そして

$$B = +3 - 1 \text{OR} +3 - 2.$$

と記述する。OR 記号は、次に OR 記号が現れるまで、それに続く全ての body 番号に適用される事に注意する必要がある。

2. サンプルプログラム uccg_nai.fの概要

uccg_nai.fは、ucrz_nai.fと同じ形状をCGを使って記述するユーザコードである。CG入力データは、ユニット4のデータファイルの先頭に記載する。

2.1. CG 入力データ

ucrz_nai.fが、円筒と平板により形状を定義しているのと異なり、第3図に示すように円筒の組み合わせで体系を定義している。

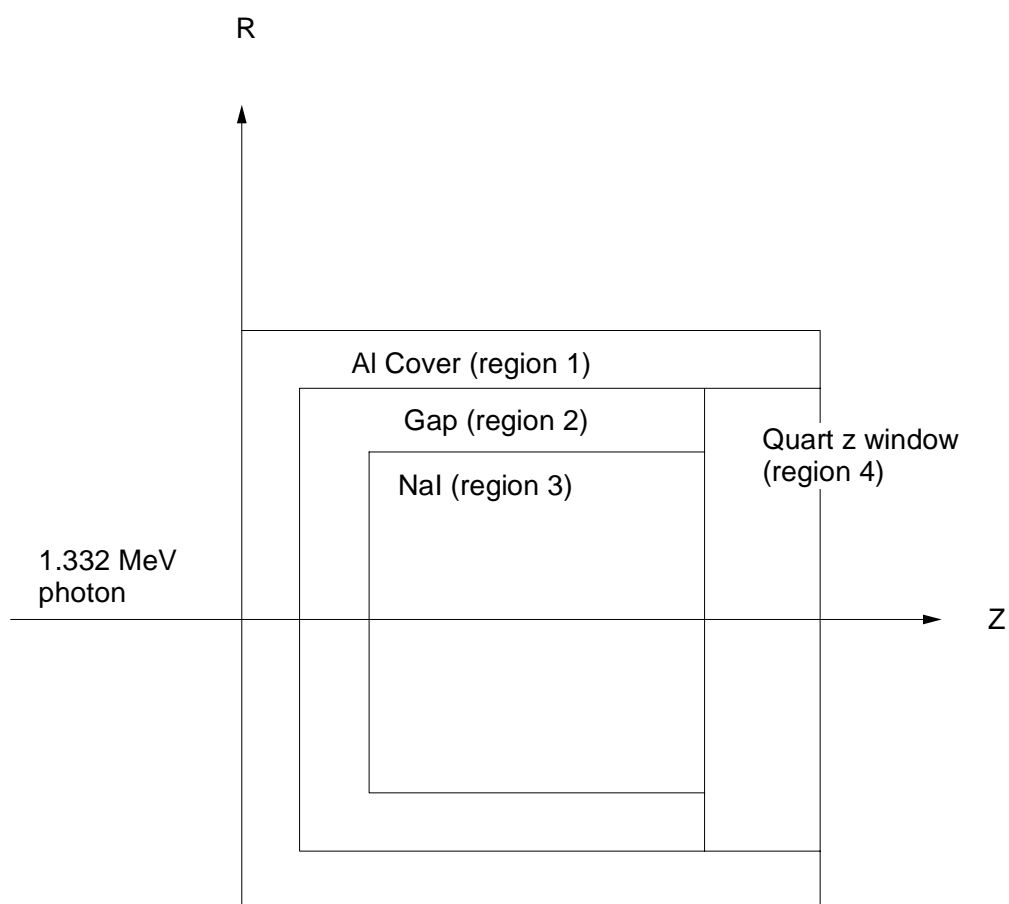


Figure 3: Geometry of uccg_nai.f

この形状の入力データは、PRESTA-CGでは、以下のように記述する。

```

RCC  1  0.00      0.0      0.0      0.00      0.0      8.72
      4.41
RCC  2  0.00      0.0      0.1      0.00      0.0      8.12
      4.31
RCC  3  0.00      0.0      0.6      0.00      0.0      7.62
      3.81
RCC  4  0.00      0.0      8.22     0.00      0.0      0.5
      4.31
RCC  5  0.00      0.0     -1.0     0.00      0.0     10.0
      5.00
END
Z1      +1  -2  -4
  
```

```
Z2          +2   -3
Z3          +3
Z4          +4
Z5          +5   -1
END
```

1. 線源条件

- 粒子のエネルギーは、ユニット 4 の入力データを用いる。
- 1.33 MeV のビーム状光子が、Z-軸に沿って、垂直に入射する。

2. 得られる情報

- 使用する物質に関するデータ
- 各リージョンに関する情報
- ピーク及び全検出効率
- レスポンス
- NaI 検出器領域の入ってくる、光子、電子、陽電子のスペクトル

3. ユーザーコードの内容

3.1. メインプログラム

3.1.1. include 文及び型式宣言: egs5 は、Fortran で書かれているので、egs5 やジオメトリーや、ユーザーコードで使われている変数の配列の大きさは、別のファイルに parameter 文で指定し、include 機能によりユーザーコードに取り入れている。common についても、同じく include 機能を用いている。

egs5 では、egs5 に直接関係する include 関係のファイルは、egs ディレクトリーの include ディレクトリと、ジオメトリー関係のサブルーティンを含め、ユーザー固有のユーザーコードに関連した include 関係ファイルは、ユーザーのワーキングディレクトリー（以下では、user_dir とする。）のサブディレクトリー user_auxcommon ディレクトリーとリンクすることにより、使用できるようにしている。[†]

この点が、Mortran のマクロ機能により、ユーザーコードで再設定できた EGS4 の場合と最も異なることである。従って、配列の大きさを変更する場合には、egs5 に直接関係する場合は、egs5.0/include/egs5.h.f 内の、その他の場合は、user_dir/user_auxcommons/aux.h.f の当該 parameter 文の値を変更することになる。

最初の設定は、egs に直接関連する include 文である。

```
implicit none
!
! -----
! EGS5 COMMONs
! -----
include 'include/egs5_h.f'           ! Main EGS "header" file

include 'include/egs5_edge.f'
include 'include/egs5_media.f'
include 'include/egs5_misc.f'
include 'include/egs5_switches.f'
include 'include/egs5_uphiot.f'
include 'include/egs5_useful.f'
include 'include/randomm.f'
```

include 'include/egs5_h.f' は、必ず必要であるが、それ以外の common に関連する include 文は、メインプログラムで、使用する可能性があるものだけで良い。[‡]

次の、設定は、ジオメトリー関係のサブルーティン及びユーザー固有のユーザーコードに関連する include 文である。

[†]これらの設定は、egs5run スクリプトで設定される。

[‡]EGS4 の COMIN マクロに対応する扱いである。

```

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

include 'user_auxcommons/edata.f'
include 'user_auxcommons/etaly1.f'
include 'user_auxcommons/instuf.f'
include 'user_auxcommons/lines.f'
include 'user_auxcommons/watch.f'

include 'auxcommons/etaly2.f'          ! Added SJW for energy balance

! -----
! cg related COMMONs
! -----
include 'user_auxcommons/cg/tvalcg.f'
include 'user_auxcommons/cg/zondta.f'
include 'user_auxcommons/cg/rppdta.f'
include 'user_auxcommons/cg/sphdta.f'
include 'user_auxcommons/cg/rccdta.f'
include 'user_auxcommons/cg/trcdta.f'
include 'user_auxcommons/cg/tordta.f'

```

etaly2.f は、準 egs5 的な扱いになっている common である。
最後の、7つの include 文が、CG に関連したものである。
特定のユーザーコード内で使用する common を次に定義する。

```

common/totals/                                ! Variables to score
* depe(20),faexp,fexps,imode,ndet,nreg
real*8 depe,faexp,fexps
integer imode,ndet,nreg

```

メインプログラムの先頭で、implicit none 宣言をしているので、メインプログラムで使用している全ての変数の型式宣言をする必要がある。

3.1.2. open 文: 実行文の先頭で、使用するユニットを open する。egs5 では、pegs をプログラムの一部として含む構造を標準としている。pegs の実行に伴い、ユニット 7-26 は、close されることから、メインプログラムで open していても、pegs 実行後に、再度 open することが必要となる。そのため、ユニット 7-26 の使用を避ける方が良い。

```

! -----
! Open files
! -----
open(UNIT= 4,FILE='egs5job.inp',STATUS='old')
open(UNIT= 6,FILE='egs5job.out6',STATUS='unknown')

```

3.1.3. Subroutine getcg の call: CG の入力データを読み込み、処理をサブルーティン geomgt を call する。次に、各種の counter パラメータをクリアサブルーティンと、getcg を call する。getcg は、egs5 で新たに必要なサブルーティンである。このサブルーティンをどの様に設定するかは、個々のユーザーコードにより異なるが、最低必要な機能は、pegs の実行と、subroutine hatch を call することである。このユーザーコードでは、物質の指定、カットオフエネルギー、オプション、入射粒子等の設定を全てこのサブルーティンにおいて、ユニット 4 から読み込むデータで設定する様にしている。

```

!-----
!      initialize cg related parameter
!-----
itbody=0
irppin=0
isphin=0
irccin=0

```



```

itorin=0
itrcin=0
izonin=0
izonad=0
itverr=0
igmmax=0
ifti = 4
ifto = 6
call geomgt(ifti,ifto,igmmax,itbody)

!-----
!   Get nreg from cg input data
!-----
nreg=izonin
if (nreg.gt.mxreg) then
  write(6,100) nreg,mxreg
100  FORMAT(' NREG(=',I12,') must be less than MXREG(=',I12,')' /' Yo
*u must chang MXREG in include/egs5_h.f.')
```

3.1.4. 各種パラメータの設定とデータの初期化: $u_{in}=v_{in}=w_{in}=0.0$ の時、等方線源とするためのフラグを設定する。

エネルギービンの幅を、入射エネルギーとビン数から計算する。このユーザーコードでは、後で述べるように、設定したヒストリーをバッチに分けて実行し、各バッチ毎の結果から、計算誤差を評価しているのので、バッチ数 (nbatch) を設定し、ヒストリー数 (ncases) をバッチ数で割って、バッチ当たりのヒストリー数を求める。その後、求めたい各データを初期化する。

```

ndet=1

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

!   Energy bin width
deltae=ekein / 50

!   Zero the variables
depe=0.D0
pef=0.D0
tef=0.D0
do j=1,50
  ph(J)=0.D0
  do nd=1,ndet
    spg(nd,j)=0.D0
    spe(nd,j)=0.D0
    spp(nd,j)=0.D0
  end do
end do

!   Set number of batch and histories per batch
nbatch = 50
ncaspb = ncases / nbatch
nofbat = 0
```

3.1.5. 輸送計算: 各バッチ毎に、ncaspb ヒストリーだけ subroutine shower を call する。この操作を、設定したバッチ回数 (nbatch) 繰り返す。線源のエネルギーは、ユニット 4 から読み込んだデータに基づき決定する。

各ヒストリー毎に、NaI 領域での吸収エネルギーの有無を調べ、吸収エネルギーがある場合には、全検出効率の数に、そのエネルギーが、入射粒子の 99.9% 以上の時は、ピーク検出効率の数に加える。また、吸収エネルギーの値により、波高分布のどの位置に属するかを調べる。

各バッチで、ncaspb ヒストリー終了毎に、バッチ毎の平均値を求める。

```

do nofbat=1,nbatch
do icasps=1,ncaspb
! -----
! Start of batch -loop
! Start of CALL SHOWER loop
! -----
! -----
! Select incident energy
! -----
eparte = 0.d0
epartd = 0.d0
! Initialize some energy-balance
! tallying parameters (SJW)

if (isamp .eq. 0) then
! Monoenergetic case
ekin = ekein
wtin = 1.0
else if (isamp .eq. 1) then
! Sample discrete energy from CDF
call randomset(rnnow)
i=0
110 continue
i = i + 1
if(ecdf(i) .le. rnnow) go to 110
ekin = ebin(i)
wtin = 1.0
else if (isamp .eq. 2) then
! Sample DIRECTLY from CDF
call edistr(ekin)
wtin = 1.0
else if (isamp .eq. 3) then
! Sample UNIFORMLY on energy
! interval and WEIGHT
call randomset(rnnow)
ekin = esam1 + rnnow*delsam
isam = 0
120 continue
isam = isam + 1
if (ekin .lt. ebin(isam)) go to 130
go to 120
130 continue
wtin = epdf(isam)
end if

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

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

! -----
! Print first NWRITE or NLINES, whichever comes first
! -----
if (ncount .le. nwrite .and. ilines .le. nlines) then
ilines = ilines + 1
write(6,140) etot,xin,yin,zin,uin,vin,win,iqin,irin,idin
140 FORMAT(4G15.7/3G15.7,3I5)
end if

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

! Added for energy balance tests (SJW)

```

```

        if(DABS(eparte + epartd - ekin)/ekin .gt. 1.d-10) then
          write(6,150) icases, eparte, epartd
150      FORMAT('Error on # ',I6,' Escape = ',F9.5,' Deposit = ',F9.5)
        endif

!      If some energy is deposited inside detector add pulse-height
!      and efficiency.

        if (depe .gt. 0.D0) then
          ie=depe/deltae + 1
          if (ie .gt. 50) ie = 50
          ph(ie)=ph(ie)+wtin
          tef=tef + wtin
          if(depe .ge. ekin*0.999) pef=pef +wtin
          depe = 0.D0
        end if

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

!      if (iwatch .gt. 0) call swatch(-1,iwatch)
!      =====

        end do                                ! -----
                                                ! End of CALL SHOWER loop
                                                ! -----

! Calculate average value for this BATCH
do ie=1,50
  phpb(ie,nofbat) = ph(ie) /ncaspb
  ph(ie)=0.D0
end do
pefpb(nofbat)=pef / ncaspb
tefpb(nofbat)=tef /ncaspb
pef=0.D0
tef=0.D0
do nd=1,ndet
  do ie=1,50
    spgpb(nd,ie,nofbat)=spg(nd,ie)/ncaspb !photon spectrum
    spepb(nd,ie,nofbat)=spe(nd,ie)/ncaspb !electron spectrum
    spppb(nd,ie,nofbat)=spp(nd,ie)/ncaspb !positron spectrum
    spg(nd,ie)=0.D0
    spe(nd,ie)=0.D0
    spp(nd,ie)=0.D0
  end do
end do

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

```

3.1.6. 統計誤差: x をモンテカルロ計算で計算したい量(スコアする量)とする。モンテカルロ計算の結果には、その統計誤差が必要である。ucrz_nai.fでは、次のようなMORSE-CGPで使用している方法を採用している。

- ヒストリー数を N とする。
- “ N ” ヒストリーをそれぞれ N/n ヒストリーの n バッチに分割する。各バッチで得られた値を x_i とする。
- x の平均値を

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (1)$$

より求める。

- 求め x_i の分散を以下の式で計算する。

$$s_x^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i^2 - \bar{x}^2) \quad (2)$$

- \bar{x} の統計誤差は、

$$s_{\bar{x}}^2 = \frac{s_x^2}{n} \quad (3)$$

となる。

- FSD(fractional standard deviation) として

$$\text{FSD} = s_{\bar{x}}/\bar{x} \quad (4)$$

を用いる。

3.1.7. 計算結果の出力: 得られた結果を処理して打ち出す処理を行う。ピーク検出効率、全検出効率及び波高分布について、バッチ毎の結果から、平均値とその誤差 (FSD) を求めて出力する。

```

! -----
! Calculate average and its deviation
! -----
!
! Peak efficiency
! -----
avpe = 0.D0
desci2 = 0.D0
do j = 1, nbatch
  avpe = avpe + pefpb(j)/nbatch
  desci2 = desci2 + pefpb(j)*pefpb(j)/nbatch
end do
sigpe = sqrt((desci2 - avpe*avpe)/(nbatch-1))
avpe = avpe*100.0
sigpe = sigpe*100.0
write(6,210) avpe,sigpe
210 FORMAT(' Peak efficiency =',G15.5,'+-',G15.5,' %')

! -----
! Total efficiency
! -----
avte = 0.D0
desci2 = 0.D0
do j = 1, nbatch
  avte = avte + tefpb(j)/nbatch
  desci2 = desci2 + tefpb(j)*tefpb(j)/nbatch
end do
sigte = sqrt((desci2 - avte*avte)/(nbatch-1))
avte = avte*100.0
sigte = sigte*100.0
write(6,220) avte,sigte
220 FORMAT(' Total efficiency =',G15.5,'+-',G15.5,' %')

! -----
! Pulse height distribution
! -----
write(6,230)
230 FORMAT('/' Pulse height distribution ')
do ie=1,50
  elow=deltae*(ie-1)
  eup=deltae*ie
  if (elow .gt. ekein ) go to 990

  avph = 0.D0
  desci2 = 0.D0

```

```

do j = 1, nbatch
  avph = avph + phpb(ie,j)/nbatch
  desc2 = desc2 + phpb(ie,j)*phpb(ie,j)/nbatch
end do
sigph = sqrt((desc2 - avph*avph)/(nbatch-1))
avph = avph/deltae
sigph = sigph/deltae
write(6,240) eup,avph,sigph
240  FORMAT(' E (upper-edge --',G10.4,' MeV )=',G15.5,'+-',G15.5,
*      ' counts/MeV/incident');
end do

990  continue

```

その後、同様にして、NaI 検出器に入射した粒子のスペクトルについて、平均と FSD を求めて出力する。

3.2. subroutine getcg

getcg は、CG 形状の問題について、使用する物質データ、各リージョンに設定する物質とオプション、ジオメトリ情報、入射粒子に関する情報、初期乱数の設定等をユニット 4 から読み込み、必要な設定を行い、pegs を実行し、その後に subroutine hatch を call するサブルーチンである。ユニット 4 から読み込むデータは以下の様になっている。

1. Record 1 : タイトル情報 (80 文字)
2. Record 2 : 使用する物質数 (nmed)
3. Record 3 : 物質名 : 24 文字で指定する。pegs 入力データの名前と対応が必要。
4. Record 4 : irlnl から irlinu のリージョンの物質、密度、ecut, pcut の指定
irlnl = 0 のデータは、指定モードの終了を意味する。
5. Record 4a : Record 4 で物質が真空でない ($medtmp \neq 0$) 場合の各種オプションの設定 (0=off, 1=on)
 - ipeangsw Switches for PE-angle sampling
 - iedgesw K & L-edge fluorescence
 - iraysw Rayleigh scattering
 - ipolarsw Linearly-polarized photon scattering
 - incohsw S /Z rejection
 - iprofrsw Doppler broadening
 - mpacrsw electron impact ionization
6. Record 5 : 入射粒子の位置 (xin, yin, zin)
7. Record 6 : 入射リージョン番号
8. Record 7 : 入射粒子の方向余弦 (uin, vin, win) $uin = vin = win = 0$ の時は、等方線源。
9. Record 8 : 最初の random number seed(ixx, jxx) を指定。ixx = 0, の時は、ixx= 123457 に、jxx = 0, の時は、jxx= 654321 に設定する。
10. Record 9 : ヒストリー数 (ncases) の指定。
11. Record 10 : 入射粒子の運動エネルギー (ekein:MeV), 電荷 (iqin) 及びサンプリング方法 (isamp) の指定。
 $isamp = 0$ の時は、 $isamp = 0$ の単一エネルギー、 $isamp = 1$ の時は、離散エネルギーからサンプリングを行い、 $isamp = 2$ の時は、スペクトルからサンプリングを行い、 $isamp = 3$ の時は、一様サンプリングを行いうエイトを使用する手法を用いることを意味する。 $isamp \neq 0$ の時は、以下のデータが必要である。
12. Record 10a : 最低エネルギーの指定。 ($isamp > 1$ の時)

13. Record 10b : 各エネルギービンの最大値 (ebin(i)) とビンに対応する確率 (epdf(i)) の指定。
 ブランク又は、0.0 は、指定の終了を意味する。
14. Record 11 : トラッキング状況を設定するフラグ (iwatch) の指定。
 iwatch= 0:トラッキングなし。iwatch = 1:反応毎のトラッキング、iwatch= 2:ステップ毎のトラッキング
15. Record 12 : 制動輻射 (ibrdst) 及び電子対生成 (iprdst) の際の角度分布オプションの設定及び
 スプリッティングパラメータ (ibrspl,nbrspl) の設定。
 ibrdst=0 制動輻射で、デフォルト値 ($\theta = m/E$) を使用
 ibrdst=1 制動輻射で、サンプリング使用 (recommended)
 iprdst=0 電子対生成で、デフォルト値 ($\theta = m/E$) を使用
 iprdst=1 電子対生成で、low-order distribution を使用
 iprdst=2 電子対生成で、推奨のサンプリングを使用
 ibrspl=0 スプリッティング使用せず
 ibrspl=1 nbrspl にスプリッティング
16. Record 13 : 電子輸送に使用するパラメータ (estepe,estepe2) の指定

3.3. subroutine ausgab

AUSGAB は、ユーザが求める情報をスコアするサブルーチンである。最初に、メインプログラムと同様に、include 文及びローカル変数の型式宣言を行う。

iwatch オプションの伴う処理、スタック番号が、最大値を超えていないことの確認後、途中結果の出力をする。

iarg < 5 の場合には、リージョン 1 とそれ以外のリージョンでの吸収エネルギーを計算する。

物質番号が 1(NaI) の時は、ステップ中での吸収エネルギーを、検出器の吸収エネルギーに加える。検出器中での更に、粒子が、外部から検出器部に入ってきた場合かどうかの判定を行い、外部から入ってきた粒子の場合は、粒子毎、エネルギー毎の情報に加える。

```

! -----
! Set some local variables
! -----
      irl = ir(np)
      iql = iq(np)
      edepwt = edep*wt(np)

! -----
! Keep track of energy deposition (for conservation purposes)
! -----
      if (iarg .lt. 5) then
         esum(iql+2,irl,iarg+1) = esum(iql+2,irl,iarg+1) + edepwt
         nsum(iql+2,irl,iarg+1) = nsum(iql+2,irl,iarg+1) + 1

! added SJW for particle by particle energy balance
         if(irl.eq.1) then
            eparte = eparte + edepwt
         else
            epartd = epartd + edepwt
         endif
      end if

! -----
! Score energy deposition inside NaI detector
! -----
      if (med(irl).eq. nreg) then
         depe = depe + edepwt

! -----
! Score particle information if it enters from outside
! -----
         if (irl .ne. iold .and. iarg .eq. 0) then
            if (iql .eq. 0) then                ! photon
               ie = e(np)/deltae + 1
            
```

```

        if(ie .gt. 50) ie = 50
        spg(1,ie) = spg(1,ie) + wt(np)
    elseif (iql .eq. -1) then          ! electron
        ie = (e(np) - RM)/deltae + 1
        if(ie .gt. 50) ie = 50
        spe(1,ie) = spe(1,ie) + wt(np)
    else                                ! positron
        ie = (e(np) - RM)/deltae + 1
        if(ie .gt. 50) ie = 50
        spp(1,ie) = spp(1,ie) + wt(np)
    end if
end if
end if
end if

! -----
! Print out stack information (for limited number cases and lines)
! -----
    if (ncount .le. nwrite .and. ilines .le. nlines) then
        ilines = ilines + 1
        write(6,101) e(np),x(np),y(np),z(np),u(np),v(np),w(np),
*           iql,irl,iarg
101  FORMAT(4G15.7/3G15.7,3I5)
    end if

! -----
! Print out particle transport information (if switch is turned on)
! -----
!
!           =====
! if (iwatch .gt. 0) call swatch(iarg,iwatch)
!           =====
!

return

end

```

3.4. subroutine howfar

howfar は、粒子の進行方向でのリージョン境界までの距離を計算し、反応点までの距離との比較をし、境界までの距離の方が短い場合には粒子の移動距離を境界までの距離に置き換え、リージョンが変わるという処理を行う。

その他に、howfarでは、ユーザが粒子の追跡を止める設定を行う。(idisc=1) 通常は、粒子が、検討している領域の外に出て追跡を終了する場合にこの設定を行う。

uccg_phantom.fでは、cgルーチンを使用している。

4. ucrz_nai.f と uccg_nai.f の計算速度の比較

複雑な形状の計算を行う場合には、cgは、相対的に容易であるが、反面、円筒平板形状のhowfarに比べ、計算時間が長いという問題がある。対象とする問題によって、違いは異なるが、ucrz_nai.fとuccg_nai.fで全く同じ条件の計算を行うと、ucrz_nai.fの方が2.1倍速いという結果が得られている。[§]

[§]CGの場合のスピードアップ(使用しているbody数が少ないこの場合には、約1.3倍)は、杉田氏の改良によってなされたものである。

5. 実習課題

5.1. 実習課題1 : NaI 検出器の計算

次のように変更して、ピーク検出効率及び全検出効率の変化を調べよ。

1. 線源を、Cs-137 の単一エネルギー光子 (0.662MeV) に変える。
2. 線源を、Co-60 に変え、1.117MeV と 1.332MeV 光子を同じ確率で発生させる。
3. Co-60 で、検出器の有感領域の厚さを 2 倍する。
4. Cs-137 線源について、一方向 (Z-方向) のみに放出している線源光子を、等方線源に変更する。

5.2. 実習課題2 : Ge 検出器の計算

検出器を、Ge に変更して、同じ大きさの NaI と、Cs-137 線源に対するピーク及び全検出効率と比較せよ。

5.3. 実習課題3 : 空気電離箱の計算

検出器を、 20° 、1 気圧の空気に変え、Cs-137 線源に対して、吸収エネルギーを求めよ。検出器の途中のギャップを除き、3 インチ直径で 3 インチ長さの空気の領域の周辺に厚さ、5mm の Al がある形状とする。

空気の W 値 (33.97eV/pair) を用いて、入射光子 1 個当たりのこの電離箱の出力 (Coulomb/source) を求めよ。電荷素量を、 $1.602 \times 10^{-19}\text{C/e}$ とする。

6. 実習課題の解答例

6.1. 実習課題 1

1. ¹³⁷ source

- uccg_nai.data の 36 行目の ekinin の値を 0.662 に変更する。
- uccg_nai.data を別な名前で保存し、egs5run の実行時にユニット 4 のファイルとして、保存したファイル名を指定する。

2. ⁶⁰Co source

- uccg_nai.data の 36 行目の ekinin の値を 1.332 に、isamp を 1 に変更する。
- 上記の後に、

```
1.117,    1.0          discrete energy 1
1.332,    1.0,        discrete energy 2
0.0,      0.0,        end of set energy
```

を挿入する。

- ucrz_nai.data を別な名前で保存し、egs5run の実行時にユニット 4 のファイルとして、保存したファイル名を指定する。

3. 長さ 2 倍の NaI

- 上記で作成したデータファイルの CG データ部分を以下の様に変更する。

```
RCC      1  0.00      0.0      0.0      0.00      0.0      16.44
          4.41
RCC      2  0.00      0.0      0.1      0.00      0.0      15.84
          4.31
RCC      3  0.00      0.0      0.6      0.00      0.0      15.34
          3.81
RCC      4  0.00      0.0     15.94      0.00      0.0       0.5
          4.31
RCC      5  0.00      0.0     -1.0      0.00      0.0      20.0
          5.00
END
Z1          +1  -2  -4
Z2          +2  -3
Z3          +3
Z4          +4
Z5          +5  -1
END
```

- ucrz_nai.data を別な名前で保存し、egs5run の実行時にユニット 4 のファイルとして、保存したファイル名を指定する。

4. Point source

- ucrz_nai.data の 33 行目の win を 0 に、35 行目の ekinin の値を 0.662 に変更する。
- uccg_nai.data を別な名前で保存し、egs5run の実行時にユニット 4 のファイルとして、保存したファイル名を指定する。

6.2. 実習課題 2

1. uccg_nai.data の 1 行目から 13 行目の NaI に関するデータを、次の Ge に関するデータに置き換え、別な名前で保存する。

```
ELEM
  &INP IAPRIM=1,EFRACH=0.05,EFRACL=0.20,
      IRAYL=1,IBOUND=0,INCOH=0,ICPROF=0,IMPACT=0 /END
GE-IAPRIM      GE
GE
ENER
  &INP AE=0.521,AP=0.0100,UE=2.511,UP=2.0 /END
TEST
```

```

&INP /END
PWLf
&INP /END
DECK
&INP /END

```

を挿入する。

- uccg_phantom.data を別な名前でも保存し、egs5run の実行時にユニット 4 のファイルとして、保存したファイル名を指定する。

6.3. 実習課題 3

- uccg_nai.f を修正。

- ローカル変数に、バッチ毎の、空気中での吸収エネルギー depepb(50) を加える。
- 円筒と平板数の減少に伴い、計算終了後の形状に関する打ち出し部を次のように変更する。

```

      tdet=7.62
      rdet=3.81
      tcov=0.1
      rtcov=0.1
      write(6,200) tdet,rdet,tcov,rtcov
200  FORMAT(/' Detector length=',G15.5,' cm'/
*      ' Detector radius=',G15.5,' cm'/
*      ' Al cover thickness=',G10.2,' cm'/
*      ' Al cover side thickness=',G10.2,' cm'/)

```

- 空気中での平均吸収エネルギーとその FSD を計算するルーチン、及び出力を計算するルーチンを加える。

```

!      -----
!      Absorbed energy in air
!      -----
      avab = 0.D0
      desc2 = 0.D0
      do j = 1, nbatch
         avab = avab + depepb(j)/nbatch
         desc2 = desc2 + depepb(j)*depepb(j)/nbatch
      end do
      sigab = sqrt((desc2 - avab*avab)/(nbatch-1))
      write(6,220) avab,sigab
220  FORMAT(' Absorbed energy in air =',G15.5,'+-',G15.5,' MeV/photon')
      avab = avab /33.97D-6 *1.602D-19
      sigab = sigab /33.97D-6 *1.602D-19
      write(6,225) avab,sigab
225  FORMAT(' Output current =',G15.5,'+-',G15.5,' C/photon')

```

新たな変数 avab, sigab を、ローカル変数の real*8 に追加する。

- ユニット 4 から読み込むデータを以下のようにする。

```

RCC   1  0.00      0.0      0.0      0.00      0.0      7.82
      3.81
RCC   2  0.00      0.0      0.1      0.00      0.0      7.72
      3.92
RCC   3  0.00      0.0     -1.0      0.00      0.0     10.0
      5.00
END
Z1      +1  -2
Z2      +2
Z3      +3  -1
END
0.662 MeV photon on Air ionozation chamber
      2      nmed
AIR-AT-NTP-IAPRIM      media(j,1) (24A1)
AL-IAPRIM      media(j,2) (24A1)

```

```

1 1 2 0. 0.561 0.0 Al
1 1 0 0 0 0 0 peang,edge,ray,pola,incoh,prof,impac
2 2 1 0. 0.561 0.0 Air
1 1 0 0 0 0 0 peang,edge,ray,pola,incoh,prof,impac
0 end of define
0.0 0.0 0.0 xin,yin,zin
1 irin
0.0 0.0 1.0 uin,vin,win
0 0 ix, jxx
100000 ncases (I10)
0.662 0 0 ekein(mev),iqin,isamp
0 iwatch
1 2 0 0 ibrdst,iprdst,ibrspl,nbrspl
0.10 0.20 estepe and estepe2

```

3. peps 用の入力データを以下のようにする。

```

MIXT
&INP NE=3,RHO= 1.2050E-03,RHOZ= 0.78,0.2103,0.0094,IAPRIM=1,
EFRACH=0.05,EFRACL=0.20,IRAYL=1,IBOUND=0,INCOH=0,
ICPROF=0,IMPACT=0 /END
AIR-AT-NTP-IAPRIM AIR-GAS
N O AR
ENER
&INP AE=0.521,AP=0.010,UE=2.511,UP=2.0 /END
PWL
&INP /END
DECK
&INP /END
ELEM
&INP IAPRIM=1,EFRACH=0.05,EFRACL=0.20,
IRAYL=1,IBOUND=0,INCOH=0,ICPROF=0,IMPACT=0 /END
AL-IAPRIM AL
AL
ENER
&INP AE=0.521,AP=0.010,UE=2.511,UP=2.0 /END
TEST
&INP /END
PWL
&INP /END
DECK
&INP /END

```

References

- [1] 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).


```

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

include 'user_auxcommons/edata.f'
include 'user_auxcommons/etaly1.f'
include 'user_auxcommons/instuf.f'
include 'user_auxcommons/lines.f'
include 'user_auxcommons/watch.f'

include 'auxcommons/etaly2.f' ! Added SJW for energy balance

!-----
! cg related COMMONs
!-----
include 'user_auxcommons/cg/tvalcg.f'
include 'user_auxcommons/cg/zondta.f'
include 'user_auxcommons/cg/rppdta.f'
include 'user_auxcommons/cg/sphdtac.f'
include 'user_auxcommons/cg/rccdta.f'
include 'user_auxcommons/cg/trcdta.f'
include 'user_auxcommons/cg/tordta.f'

common/totals/ ! Variables to score
* depe,deltae,spg(1,50),spe(1,50),spp(1,50),nreg
real*8 depe,deltae,spg,spe,spp
integer nreg

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

real*8 ! Local variables
* availke,avpe,avph,avspe,avspg,avsp,avte,desci2,ekin,pef,
* sigpe,sigte,sigph,sigspg,sigspe,sigspp,tef,wtin,wsum

real*8
* ph(50),phpb(50,50),spgpb(1,50,50),spepb(1,50,50),
* spppb(1,50,50),pefpb(50),tefpb(50)

real ! Local variables
* elow,eup,rdet,rtcov,rtgap,tcov,tdet,tgap

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

integer
* icases,idin,isam,isot,nlist,
* i,j,k,ireg,n,imed,ndet,nd,nbatch,ncaspb,nofbat,ie,
* itbody,izonad,
* igmmax,ifti,ifto

!-----
! Open files
!-----
open(UNIT= 4,FILE='egs5job.inp',STATUS='old')
open(UNIT= 6,FILE='egs5job.out6',STATUS='unknown')

!-----
! initialize cg related parameter
!-----
itbody=0
irppin=0
isphin=0
irccin=0
itorin=0
itrclin=0
izonin=0
izonad=0
itverr=0
igmmax=0
ifti = 4
ifto = 6
call geomgt(ifti,ifto,igmmax,itbody)

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

```

```

!-----
nreg=izonin
if (nreg.gt.mxreg) then
  write(6,100) nreg,mxreg
100  FORMAT(' NREG(=,I12,') must be less than MXREG(=,I12,')' /
* ' You must change MXREG in include/egs5_h.f.')
  stop
end if

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

!
=====
call getcg(nreg)
=====

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

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

110 write(6,110)
  FORMAT(/, ' ENERGY/COORDINATES/DIRECTION COSINES/ETC.',/,
*        6X, 'E',16X, 'X',14X, 'Y',14X, 'Z'/
*        1X, 'U',14X, 'V',14X, 'W',9X, 'IQ',4X, 'IR',3X, 'IARG',/)

!
=====
if (iwatch .gt. 0) call swatch(-99,iwatch)
=====

ndet=1

! -----
! Set isotropic source flag if uin=vin=win=0
! -----
isot=0 ! monodirectional
if (uin+vin+win.eq.0.0) then
  isot=1
115  write(6,115)
  FORMAT(' Isotropic source')
end if

! Energy bin width
deltae=ekein / 50

! Zero the variables
depe=0.D0
pef=0.D0
tef=0.D0
do j=1,50
  ph(J)=0.D0
  do nd=1,ndet
    spg(nd,j)=0.D0
    spe(nd,j)=0.D0
    spp(nd,j)=0.D0
  end do
end do

! Set number of batch and histories per batch
nbatch = 50
ncaspb = ncases / nbatch
nofbat = 0

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

do nofbat=1,nbatch
do icases=1,ncaspb
! -----
! Start of batch -loop
! Start of CALL SHOWER loop

```

```

! -----
!
! Select incident energy
! -----
eparte = 0.d0          ! Initialize some energy-balance
epartd = 0.d0          !   tallying parameters (SJW)

if (isamp .eq. 0) then          ! Monoenergetic case
  ekin = ekein
  wtin = 1.0
else if (isamp .eq. 1) then      ! Sample discrete energy from CDF
  call randomset(rnnow)
  i=0
120  continue
  i = i + 1
  if(ecdf(i) .le. rnnow) go to 120
  ekin = ebin(i)
  wtin = 1.0
else if (isamp .eq. 2) then      ! Sample DIRECTLY from CDF
  call edistr(ekin)
  wtin = 1.0
else if (isamp .eq. 3) then      ! Sample UNIFORMLY on energy
  call randomset(rnnow)          ! interval and WEIGHT
  ekin = esam1 + rnnow*delsam
  isam = 0
130  continue
  isam = isam + 1
  if (ekin .lt. ebin(isam)) go to 140
  go to 130
140  continue
  wtin = epdf(isam)
end if

wtsum = wtsum + wtin          ! Keep running sum of weights
etot = ekin + 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

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

! -----
! Print first NWRITE or NLINES, whichever comes first
! -----
if (ncount .le. nwrite .and. ilines .le. nlines) then
  ilines = ilines + 1
  write(6,150) etot,xin,yin,zin,uin,vin,win,iqin,irin,idin
150  FORMAT(4G15.7/3G15.7,3I5)
end if

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

! Added for energy balance tests (SJW)
if(DABS(eparte + partd - ekin)/ekin .gt. 1.d-10) then
  write(6,160) icases, eparte, partd
160  FORMAT('Error on # ',I6,' Escape = ',F9.5,' Deposit = ',F9.5)
endif

! If some energy is deposited inside detector add pulse-height
! and efficiency"
if (depe .gt. 0.D0) then
  ie=depe/deltae + 1
  if (ie .gt. 50) ie = 50
  ph(ie)=ph(ie)+wtin
  tef=tef + wtin
  if(depe .ge. ekin*0.999) pef=pef +wtin
  depe = 0.D0
end if

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

! =====
! if (iwatch .gt. 0) call swatch(-1,iwatch)

```



```

!
=====

end do                                     ! -----
                                           ! End of CALL SHOWER loop
                                           ! -----

! Calculate average value for this BATCH
do ie=1,50
  phpb(ie,nofbat) = ph(ie) /ncaspb
  ph(ie)=0.D0
end do
pefpb(nofbat)=pef / ncaspb
tefpb(nofbat)=tef /ncaspb
pef=0.D0
tef=0.D0
do nd=1,ndet
  do ie=1,50
    spgpb(nd,ie,nofbat)=spg(nd,ie)/ncaspb !photon spectrum
    spepb(nd,ie,nofbat)=spe(nd,ie)/ncaspb !electron spectrum
    spppb(nd,ie,nofbat)=spp(nd,ie)/ncaspb !positron spectrum
    spg(nd,ie)=0.D0
    spe(nd,ie)=0.D0
    spp(nd,ie)=0.D0
  end do
end do

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

tt=etime(tarray)
tt1=tarray(1)
cputime=tt1-tt0
write(6,170) cputime
170  format(/' Elapsed Time (sec)=' ,G15.5)

!
=====
!   if (iwatch .gt. 0) call swatch(-88,iwatch)
=====
!
! -----
! Write out the results
! -----
write(6,180) ncount,ncases,totke,iseed1,iseed2
180  FORMAT(/' Ncount=',I10,' (actual cases run)',/,
*      ' Ncases=',I10,' (number of cases requested)',/,
*      ' TotKE =' ,G15.5,' (total KE (MeV) in run)'/
*      ' Last iseed1 =' ,I12,' , iseed2 =' ,I12)

if (totke .le. 0.D0) then
  write(6,190) totke,availke,ncount
190  FORMAT(/' Stopped in MAIN with TotKE=' ,G15.5,/,
*      ' AvailKE=' ,G15.5, /, ' Ncount=' ,I10)
  stop
end if

tdet=7.62
rdet=3.81
tcov=0.1
rtcov=0.1
tgap=0.5
rtgap=0.5
write(6,200) tdet,rdet,tcov,rtcov,tgap,rtgap
200  FORMAT(/' Detector length=' ,G15.5,' cm'/
*      ' Detector radius=' ,G15.5,' cm'/
*      ' Al cover thickness=' ,G10.2,' cm'/
*      ' Al cover side thickness=' ,G10.2,' cm'/
*      ' Front gap =' ,G10.2,' cm'/' Side gap =' ,G10.2,' cm'/)

if (isamp.eq.0) then
  write(6,210) ekin
210  FORMAT(' Results for ' ,G15.5,' MeV photon'/)
else if (isamp.eq.1) then
  write(6,212) ekein
212  FORMAT(' Source energy is sampled from discrete ons.'/
*      ' Higest energy is ' ,G15.5,' MeV'/)
else if (isamp.eq.2) then
  write(6,214)
214  FORMAT(' Source enegy is sampled DIRECTLY from CDF'/)

```

```

else
  write(6,216)
216  FORMAT(' Source energy is sampled UNIFORMLY on energy interval'/
*      ' and use Weight'/)
end if

! -----
! Calculate average and its deviation
! -----

! -----
! Peak efficiency
! -----
avpe = 0.D0
desci2 = 0.D0
do j = 1, nbatch
  avpe = avpe + pefpb(j)/nbatch
  desc2 = desc2 + pefpb(j)*pefpb(j)/nbatch
end do
sigpe = sqrt((desc2 - avpe*avpe)/(nbatch-1))
avpe = avpe*100.0
sigpe = sigpe*100.0
write(6,220) avpe,sigpe
220  FORMAT(' Peak efficiency =',G15.5,'+-',G15.5,' %')

! -----
! Total efficiency
! -----
avte = 0.D0
desci2 = 0.D0
do j = 1, nbatch
  avte = avte + tefpb(j)/nbatch
  desc2 = desc2 + tefpb(j)*tefpb(j)/nbatch
end do
sigte = sqrt((desc2 - avte*avte)/(nbatch-1))
avte = avte*100.0
sigte = sigte*100.0
write(6,230) avte,sigte
230  FORMAT(' Total efficiency =',G15.5,'+-',G15.5,' %')

! -----
! Pulse height distribution
! -----
write(6,240)
240  FORMAT('/ Pulse height distribution ')
do ie=1,50
  elow=deltae*(ie-1)
  eup=deltae*ie
  if (elow .gt. ekein ) go to 260

  avph = 0.D0
  desc2 = 0.D0
  do j = 1, nbatch
    avph = avph + phpb(ie,j)/nbatch
    desc2 = desc2 + phpb(ie,j)*phpb(ie,j)/nbatch
  end do
  sigph = sqrt((desc2 - avph*avph)/(nbatch-1))
  avph = avph/deltae
  sigph= sigph/deltae
  write(6,250) eup,avph,sigph
250  FORMAT(' E (upper-edge --',G10.4,' MeV )=',G15.5,'+-',G15.5,
*      ' counts/MeV/incident');
end do

260  continue

! -----
! Particle spectrum. Incident particle spectrum to detector.
! -----
write(6,270)
270  FORMAT('/ Particle spectrum crossing the detector plane'/
*      30X,'particles/MeV/source photon'/
*      ' Upper energy',11X,' Gamma',18X,' Electron',
*      14X,' Positron')

do nd=1,ndet
  do ie=1,50

```

```

        elow=deltae*(ie-1)
        eup=deltae*ie
        if (elow .gt. ekein ) go to 290
!-----
! Gamma spectrum per MeV per source
!-----

        avspg = 0.D0
        desc12 = 0.D0
        do j = 1, nbatch
            avspg = avspg + spgpb(nd,ie,j)/nbatch
            desc12 = desc12 + spgpb(nd,ie,j)*spgpb(nd,ie,j)/nbatch
        end do
        sigspg = sqrt((desc12 - avspg*avspg)/(nbatch-1))
        avspg = avspg/deltae
        sigspg= sigspg/deltae

!-----
! Electron spectrum per MeV per source
!-----

        avspe = 0.D0
        desc12 = 0.D0
        do j = 1, nbatch
            avspe = avspe + spepb(nd,ie,j)/nbatch
            desc12 = desc12 + spepb(nd,ie,j)*spepb(nd,ie,j)/nbatch
        end do
        sigspe = sqrt((desc12 - avspe*avspe)/(nbatch-1))
        avspe = avspe/deltae
        sigspe= sigspe/deltae

!-----
! Positron spectrum per MeV per source
!-----

        avspg = 0.D0
        desc12 = 0.D0
        do j = 1, nbatch
            avspg = avspg + spppb(nd,ie,j)/nbatch
            desc12 = desc12 + spppb(nd,ie,j)*spppb(nd,ie,j)/nbatch
        end do
        sigspg = sqrt((desc12 - avspg*avspg)/(nbatch-1))
        avspg = avspg/deltae
        sigspg= sigspg/deltae

280      write(6,280) eup,avspg,sigspg,avspe,sigspe,avspg,sigspg
        FORMAT(G10.5,' MeV--',3(G12.5,'+-',G12.5))
        end do
        end do

290      continue
!=====
        call ecnsv1(1,nreg,totke)
        call ntally(1,nreg)
!=====

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

!-----
! Close files
!-----
        close(UNIT=4)
        close(UNIT=6)

        stop

        end

!-----last line of main code-----

!-----getcg.f-----
! Version: 040630-1300 KKK-LSCAT
! Reference: KEK Internal 2000-1
!-----
!23456789|123456789|123456789|123456789|123456789|123456789|123456789|12

```

Auxiliary subroutine for use with the EGS5 Code System

This is a data-entry subprogram for use with a cg geometry.
The data input is similar to that in ucrz.
However, this version is designed specifically to utilize
cg geometry.

SUBROUTINE ARGUMENT

nreg Number of regions in geometry (determined by data input).

UNIT ASSIGNMENTS

Unit 4 Input file.
Unit 6 Output file.
Unit 8 Echoes input cross-section data (assign a null file).
Unit 12 Input cross-section file from PEGS5.

INPUT FILE

=====
CG geometry related data must be written before following data.
=====

Record 1 title (80A1) Title line.

Record 2 nmed Number of media in problem.

Record 3 media(j,i) (24A1) Media names (j=1,24, I=1,nmed lines).

Record 4 irlinl,irlinu,medtmp, rhotmp, ecutin, pcutin
----- (3I5,3F10.3) Set material for region from irlinl to ielinh.
 medtmp : material number
 rhotmp : If rhotmp=0.0, the default
 value for that medium is used.
 ecutin, pcutin : KINETIC energy cutoffs
 for electrons and photons, respectively,
 in MeV. If > 0, ecut(i) and pcut(i) are
 set. Otherwise ae and ap are used (default).
 irlinl =0 means end of define.

 If medtmp not 0, following data follows.

Record 4a ipeangsw, Switches for PE-angle sampling,
----- iedgesw, K & L-edge fluorescence,
 iraysw, Rayleigh scattering,
 ipolarsw, Linearly-polarized photon scattering,
 incohrsw, S/Z rejection,
 iprofrsw, Doppler broadening,
 impacrsw electron impact ion-ization (0=off, 1=on).
 (7I5)
...+...1...+...2...+...3...+...4...+...5...+...6...+...7..

Record 5 xin,yin,zin Incident X,Y,Z coordinates (cm).

Record 6 irin Incident region.

Record 7 uin,vin,win Incident direction cosines (U,V,W).
----- If uin=vin=win=0, isotropic.

Record 8 ixx,jxx Starting random number seeding.
----- If ixx = 0, ixx is set to 123457.
 If jxx = 0, jxx is set to 654321.

Record 9 ncases Number of cases.

Record 10 ekein,iqin,isamp Kinetic energy (MeV), charge of inci-
----- dent beam, and sampling switch. If
 isamp=0, a monoenergetic beam (ekein)
 will be used. Otherwise, a spectrum
 input must follow (Records 10a through
 10b), which will be sampled from discrete
 energy (isamp=1), directly (isamp=2) or
 uniformly over the energy range (isamp=3)

```

! -----
! Record 10a ebinmin                Only required when isamp>1(see above).
!                                     Lowest energy (MeV) in spectrum.
! -----
! Record 10b ebin(i),epdf(i)       Only required when usamp>0(see above).
!                                     ebin(I) is the 'top-edge' of each
!                                     energy bin (MeV) and epdf(i) is the
!                                     corresponding probability for the bin.
!                                     For example, a cross section (mb) can
!                                     be used for epdf (but do not divide it
!                                     by dE). The last card is a delimiter
!                                     and should be blank (or contain 0.0).
!                                     The i-subscript runs from 1 to nebin
!                                     (nebin calculated after the delimiter)
! -----
! Record 11 iwatch                  Switche for tracking events with swatch:
!                                     (0=No, 1=each interaction,
!                                     2=each step)
! -----
! Record 12 ibrdst,iprdst,         Switches for bremsstrahlung and pair
! ----- ibrspl,nbrspl             production ANGLE SAMPLING, and brems-
!                                     strahlung SPLITTING:
!                                     ibrdst=0 No (use default: theta=m/E)
!                                     1 Yes (recommended)
!                                     iprdst=0 No (use default: theta=m/E)
!                                     1 Yes (low-order distribution)
!                                     2 Yes (recommended)
!                                     ibrspl=0 No
!                                     1 Yes (NBRSP=splitting factor)
! -----
! Record 13 estepe,estepe2
! -----

```

```

subroutine getcg(nreg)

implicit none

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

include 'include/egs5_bounds.f'     ! COMMONs required by EGS5 code
include 'include/egs5_brempr.f'
include 'include/egs5_edge.f'
include 'include/egs5_eiicom.f'
include 'include/egs5_elecin.f'
include 'include/egs5_media.f'
include 'include/egs5_misc.f'
include 'include/egs5_switches.f'
include 'include/egs5_thresh.f'
include 'include/egs5_useful.f'
include 'include/egs5_userpr.f'
include 'include/egs5_usersc.f'
include 'include/egs5_uservr.f'
include 'include/egs5_userxt.f'

include 'pegscommons/mscom.f'       ! PEGS common

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

include 'user_auxcommons/edata.f'
include 'user_auxcommons/instuf.f'
include 'user_auxcommons/watch.f'

include 'include/randomm.f'         ! Additional (non-EGS5) COMMON

integer nreg                         ! Arguments

real*8                               ! Local variables
* ecutmn,ek0,
* ecutin,deg2rad,
* pcutin,rhotmp,
* therad,totphi

integer i,iexp,ii,iiz,irlin,irlinl,irlinu,ixx,izn,j,
* jxx,k,n,medtmp,moreOutput,ner,nrbin,nzbin

```

```

data deg2rad/0.01745329/
data moreOutput/0/      ! Change this from 0 to 1 for more output

write(6,1100)
1100 FORMAT(//,T25,'+-----+',
*          /,T25,'| EGS5 User Code using subroutine Getcg |',
*          /,T25,'+-----+',
*          /,T25,'| NOTE:  cg geometry. |',
*          /,T25,'+-----+',
*          //)

! SJW 02-May-2002 New subroutine calls to initialize data no
! longer set in block data because of size issues

! =====
! call block_set           ! Initialize some general variables
! =====

! =====
! call region_init        ! Initialize some region variables
! =====

! -----
! Record 1: title
! -----
      read(4,101) title
101  FORMAT(80A1)
      write(6,102) title
102  FORMAT(' TITLE: '/1X,80A1/)

! -----
! Record 2: nmed
! -----
      read(4,*) nmed
      if (nmed .gt. MXMED) then
104  FORMAT(' *** Stopped in Getcg with nmed=',I5,' > MXMED')
      stop
      end if
      write(6,105) nmed
105  FORMAT(' nmed=',I5,/)

! -----
! Record 3: media
! -----
      do i=1,nmed
      read(4,106) (media(j,i),j=1,24)
106  FORMAT(24A1)
      write(6,107) i,(media(j,i),j=1,24)
107  FORMAT(' MEDIUM=',I5,' ==> ',24A1)
      end do

      do i=1,nreg           ! Set all regions to vacuum to begin with
      med(i) = 0
      end do

      write(6,108) ipeangsw,iedgesw,iraysw
108  FORMAT(//,' ipeangsw=',I5,
*          ' Photoelectric-angle sampling (0=off, 1=on)',
*          /,' iedgesw =',I5,
*          ' K/L-edge switch (0=off, 1=on)',
*          /,' iraysw =',I5,
*          ' Rayleigh scattering switch (0=off, 1=on)')

      write(6,109) ipolarsw,incohrsw,iprofrsw,impacrsw
109  FORMAT(//,' ipolarsw=',I5,
*          ' Linearly polarized photon switch (0=off, 1=on)',
*          /,' incohrsw=',I5,
*          ' S/Z rejection switch (0=off, 1=on)',
*          /,' iprofrsw=',I5,
*          ' Doppler broadening switch (0=off, 1=on)',
*          /,' impacrsw=',I5,
*          ' Electron impact ionization switch (0=off, 1=on)')

      write(6,140) nreg-1
140  FORMAT(//,' Assign medium and related flag for 1 to nreg-1 (=
*          ,I5,')')

```

```

! -----
! Record 4 irlinl, irlinu, medtmp, rhotmp, ecutin, pcutin
! -----
! Define to each region
! -----
142  continue
    read(4,143) irlinl, irlinu, medtmp, rhotmp, ecutin, pcutin
143  FORMAT(3I5,3F10.3)
    if (irlinl .eq. 0) go to 160

    if (medtmp.ne.0) then
! -----
! Record 4a: ipeangsw, iedgesw, iraysw, ipolarsw,
!           incohrsw, iprofrsw, impacrsw
! -----
        read(4,145) ipeangsw, iedgesw, iraysw, ipolarsw, incohrsw,
*         iprofrsw, impacrsw
145  FORMAT(7I5)

        write(6,146) irlinl, irlinu, medtmp, rhotmp, ecutin, pcutin
146  FORMAT(' Region from', I5, ' to', I5, ': medium =', I5, ', rhoh=',
*         G15.5/11X, ' ecut =', G15.5, ', pcut =', G15.5)

        write(6,150) ipeangsw, iedgesw, iraysw
150  FORMAT(11X, ' iphter=', I3, 3X, ' iedgfl=', I3, 3X, ' iraylr=', I3)
        write(6,152) ipolarsw, incohrsw, iprofrsw, impacrsw
152  FORMAT(11X, ' lpolar=', I3, 3X, ' incohr=', I3, 3X, ' iprofr=', I3,
*         3X, ' impacr=', I3)
    else
        write(6,153) irlin
153  FORMAT(' Region =', I5, ' is vacuum')
    end if

    do irlin=irlinl, irlinu
        med(irlin)=medtmp
        if (medtmp.ne.0) then
            if (rhotmp. gt. 0.) then
                rhor(irlin) = rhotmp
            end if
            if (ecutin .gt. 0.) then
                ecut(irlin) = pcutin
            end if
            if (pcutin .gt. 0.) then
                pcut(irlin) = pcutin
            end if
            iphter(irlin) = ipeangsw
            iedgfl(irlin) = iedgesw
            iraylr(irlin) = iraysw
            lpolar(irlin) = ipolarsw
            incohr(irlin) = incohrsw
            iprofr(irlin) = iprofrsw
            impac(irlin) = impacrsw
        end if
    end do
    go to 142

160  continue
! -----
! Record 5: xin, yin, zin
! -----
        read(4,*) xin, yin, zin

        write(6,180) xin, yin, zin
180  FORMAT(/, ' xin=', G15.7, 5X, ' yin=', G15.7, 5X, ' zin=', G15.7
*         /' (incident coordinates)')

! -----
! Record 5: irin
! -----
        read(4,*) irin
        write(6,190) irin
190  FORMAT(/, ' irin=', I5, ' (incident region)')

! -----
! Record 6: uin, vin, win
! -----

```

```

        read(4,*) uin,vin,win
        write(6,200) uin,vin,win
200    FORMAT(/,' uin=',G15.7,5X,' vin=',G15.7,5X,' win=',G15.7,
*        ' (incident direction cosines)')

! SJW 02-May-2002 Not needed for EGS5
! -----
! Record 7: ixj,jxx
! -----
        read(4,*) ixj,jxx
        if (ixj .eq. 0) ixj = 123457          ! Default seed
        if (jxx .eq. 0) jxx = 654321        ! Default seed
        write(6,210) ixj,jxx
210    FORMAT(/,' ixj=',I12,5X,' jxx=',I12,
*        ' (starting random-number seeds)')

! -----
! Save the starting random-number seeds
! -----
        iseed1=ixj
        iseed2=jxx

! =====
! call rmarin          ! Initialize the random-number generator
! =====

! -----
! Record 8: ncases
! -----
        read(4,*) ncases
        write(6,220) ncases
220    FORMAT(/,' ncases=',I12)

! -----
! Record 9: ekein,iqin,isamp
! -----
        read(4,*) ekein,iqin,isamp
        if (isamp .eq. 0) then                ! -----
                                                ! Monoenergetic case
                                                ! -----
                write(6,230) iqin,ekein
230    FORMAT(/,' MONOENERGETIC case has been selected with:',
*        '//',' iqin=',I5,' (incident charge of beam)',
*        '/',' ekein=',G15.5,' MeV (incident kinetic energy)')

        else if (isamp .gt. 0) then          ! -----
                                                ! Energy spectrum case
                                                ! -----

! -----
! Record 9a: ebinmin
! -----
        if(isamp.ne.1) then
                read(4,*) ebinmin          ! Lowest energy in spectrum (MeV)
                write(6,240) iqin,ebinmin
240    FORMAT(/,' Energy-SPECTRUM case has been selected with:',
*        '//',' iqin=',I5,' (incident charge of beam)',
*        '/',' ebinmin=',F10.3,' MeV (lowest energy bin)')
                end if

                if (isamp .eq. 1) then
                        write(6,245) isamp
245    FORMAT(' isamp =',I2,' (Sample from discrete energy)')
                elseif (isamp .eq. 2) then
                        write(6,250) isamp
250    FORMAT(' isamp =',I2,' (DIRECT-sampling over energy range)')
                else if (isamp .eq. 3) then
                        write(6,260) isamp
260    FORMAT(' isamp =',I2,
*        ' (UNIFORM-sampling over energy range) with WEIGHTING')
                end if

! -----
! Record 9b: ebin(i),epdf(i)
! -----
        i = 0
265    continue                ! -----
                                ! Start of energy-spectrum input loop
                                ! -----

```



```

        i = i + 1
        if (i .gt. MXEBIN) then
270          write(6,270) i
             FORMAT(//,' Stopped in getcg with I=',I6,' > MXEBIN')
             stop
        end if
        read(4,*) ebin(i),epdf(i)          ! ebin(i) is top-edge of bin
        if (i .gt. 1 .and. ebin(i) .le. ebin(i-1)) then
            go to 285
        else if (i .eq. 1 .and. ebin(i) .le. ebinmin) then
            go to 275
        end if
        go to 265

275  continue          ! Reach here when a read-error occurs
        write(6,280)
280  FORMAT(//,' Stopped in getcg with spectrum read-error')
        stop

285  continue          ! Reach here when delimiter card has been read

        nebin = i - 1          ! Number of energy bins read in
        totphi = 0.
        do i=1,nebin
            totphi = totphi + epdf(i)
        end do
        ecdf(1) = epdf(1)/totphi
        do i=2,nebin
            ecdf(i) = ecdf(i-1) + epdf(i)/totphi
        end do

        write(6,290) (i,ebin(i),epdf(i),ecdf(i),i=1,nebin)
290  FORMAT(/,' BIN    UPPER ENERGY    PROBABILITY    CUMULATIVE ',
*         /,' #          (MeV)          PROBABILITY',
*         /,'(I4,3X,F10.3,2F16.4))

! -----
! Set up energy-sampling interval
! -----
        esam1 = ebinmin
        esam2 = ebin(nebin)
        delsam = esam2 - esam1

        write(6,300) esam1,esam2
300  FORMAT(//,' Energy-sampling interval is:',/,
*         ' esam1 =',G15.5,' MeV to esam2 =',G15.5,' MeV',/)
        else
            write(6,310) isamp
310  FORMAT(/,' Stopped in getcg with bad isamp=',I10)
            stop
        end if

! -----
! Record 10: iwatch
! -----
        read(4,*) iwatch
        write(6,350) iwatch
350  FORMAT(//,' SWATCH tracking switch: iwatch=',I2,
*         ' (0=off, 1=each interaction, 2=each step)')

! -----
! Record 11: ibrdst,iprdst,ibrspl,nbrspl
! -----
        read(4,*) ibrdst,iprdst,ibrspl,nbrspl
        write(6,410) ibrdst,iprdst,ibrspl,nbrspl
410  FORMAT(/,' IBRDST=',I2,/, ' IPRDST=',I2,/, ' IBRSPL=',I2, ' (NBRSP='
*         ',I5,')')

        if (ibrspl .gt. 0) then
            if (nbrspl .gt. 0) then
                fbrspl = 1.0/float(nbrspl)
            else
                write(6,420) ibrspl,nbrspl
420  FORMAT(//,' Stopped in Getcg with IBRSPL=',I5, ' and NBRSP=',
*         I5)
                stop
            end if
        end if

```

```

! -----
! Run KEK version of PEGS5 before calling HATCH
! (method was developed by Y. Namito - 010306)
! -----
430 write(6,430)
   FORMAT(/,' PEGS5NB3-call comes next',/)
!
! =====
! call pegs5nb3
! =====
!
! -----
! Open files (before HATCH call)
! -----
   open(UNIT=KMPI,FILE='pgs5job.pegs5dat',STATUS='old')
   open(UNIT=KMPO,FILE='egs5job.dummy',STATUS='unknown')
440 write(6,440)
   FORMAT(/,' HATCH-call comes next',/)
!
! =====
! call hatch
! =====
!
! -----
! Close files (after HATCH call)
! -----
   CLOSE(UNIT=KMPI)
   CLOSE(UNIT=KMPO)
! SJW 02-May-2002 replace reading of PRESTA switches with
! estepe and estepe2, and call to presta_inputs with calls
! to check_limits and rmsfit
! Set minimum (total) energy
   ecutmn = 1.D10
   do i = 1,nreg
     if (ecut(i).gt.0.0) ecutmn=min(ecutmn,ecut(i))
   end do
   ek0 = ekein           ! Set maximum (kinetic) energy
!
! =====
! call presta_inputs(nreg,ecutmn,ek0)   ! Do PRESTA inputs/summary
! =====
! -----
! Record 12: estepe,estepe2
! -----
   read(4,*) estepe, estepe2
   write(6,450) estepe, estepe2
450  FORMAT(/,1X,' ESTEPE at EKMAX: ',F10.0,' (estepe)',
*      /,1X,' ESTEPE at ECUT: ',F10.0,' (estepe2)')
!
! -----
! Print values used for efrac1 and efrac
! -----
   write(6,*)
   write(6,*) ' EFRACL=',efrac1
   write(6,*) ' EFRACH=',efrach
!
! =====
! call check_limits(nreg,ecutmn,ek0)   ! Set energy step constants
! =====
!
! =====
! call rmsfit           ! read multiple scattering data
! =====
! -----
! All of the input data should have been read in at this point,
! but check to make sure that the incident kinetic energy is
! below the limit set by PEGS (i.e., UE and UP) for all media.
! -----
   do j=1,nmed
     if (ekein+RM .gt. ue(j)) then
       write(6,*)
*       'Stopped in SUBROUTINE getcg with ekein + RM > ue(j):'
       write(6,*) '   j = ',j
       write(6,*) '   ekein + RM = ',ekein+RM

```

```

        write(6,*) ' ue(j) = ',ue(j)
        stop
    end if
    if (ekein .gt. up(j)) then
        write(6,*)
    *   'Stopped in SUBROUTINE getcg with ekein > up(j):'
        write(6,*) ' j = ',j
        write(6,*) ' ekein = ',ekein
        write(6,*) ' up(j) = ',up(j)
        stop
    end if
end do

! -----
! Print various data associated with each media (not region)
! -----
460 write(6,460)
    FORMAT(/,' Quantities associated with each MEDIA:')
    do j=1,nmed
        write(6,470) (media(i,j),i=1,24)
470     FORMAT(/,1X,24A1)
        write(6,480) rho(j),rlc(j)
480     FORMAT(5X,' rho=',G15.7,' g/cu.cm      rlc=',G15.7,' cm')
        write(6,490) ae(j),ue(j)
490     FORMAT(5X,' ae=',G15.7,' MeV      ue=',G15.7,' MeV')
        write(6,500) ap(j),up(j)
500     FORMAT(5X,' ap=',G15.7,' MeV      up=',G15.7,' MeV',/)
    end do

! -----
! Print media and cutoff energies assigned to each region
! -----
    if (moreOutput .eq. 1) then
        do i=1,nreg
            if (med(i) .eq. 0) then
                write(6,510) i,ecut(i),pcut(i)
510     *   FORMAT(' medium(',I3,')=vacuum',18X,
                'ecut=',G10.5,' MeV, pcut=',g10.5,' mev')
            else
                write(6,520) i,(media(ii,med(i)),ii=1,24),ecut(i),pcut(i)
520     *   FORMAT(' medium(',I3,')=',24A1,
                'ecut=',G10.5,' MeV, pcut=',G10.5,' MeV')
            ! -----
            ! Print out energy information of K- and L-X-rays
            ! -----
            if (iedgfl(i) .ne. 0) then                ! Output X-ray energy
                ner = nne(med(i))
                do iiz=1,ner
                    izn = zelem(med(i),iiz) ! Atomic number of this element
                    write(6,530) izn
530     *   FORMAT(' X-ray information for Z=',I3)
                    write(6,540) (ekx(ii,izn),ii=1,10)
540     *   FORMAT(' K-X-ray energy in keV',/,
                    4G15.5,/,4G15.5,/,2G15.5)
                    write(6,550) (elx1(ii,izn),ii=1,8)
550     *   FORMAT(' L-1 X-ray in keV',/,4G15.5,/,4G15.5)
                    write(6,560) (elx2(ii,izn),ii=1,5)
560     *   FORMAT(' L-2 X-ray in keV',/,5G15.5)
                    write(6,570) (elx3(ii,izn),ii=1,7)
570     *   FORMAT(' L-3 X-ray in keV',/,4G15.5,/,3G15.5)
                end do
            end if
        end if
    end do
end if

return                                     ! -----
                                           ! Return to MAIN
                                           ! -----

end

!-----last line of getcg.f-----
!-----ausgab.f-----
! Version: 030831-1300
! Reference: SLAC-265 (p.19-20, Appendix 2)
!-----
!23456789|123456789|123456789|123456789|123456789|123456789|123456789|123456789|123456789|

```

```

-----
! Required subroutine for use with the EGS5 Code System
-----
! A simple AUSGAB to:
!
! 1) Score energy deposition
! 2) Print out stack information
! 3) Print out particle transport information (if switch is turned on)
!
-----

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 'user_auxcommons/aux_h.f'   ! Auxiliary-code "header" file
include 'user_auxcommons/etaly1.f'  ! Auxiliary-code COMMONs
include 'user_auxcommons/lines.f'
include 'user_auxcommons/ntaly1.f'
include 'user_auxcommons/watch.f'

include 'auxcommons/etaly2.f'       ! Added SJW for energy balance

common/totals/                      ! Variables to score
* depe,deltae,spg(1,50),spe(1,50),spp(1,50),nreg
real*8 depe,deltae,spg,spe,spp
integer nreg

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)

!
! -----
! Keep track of energy deposition (for conservation purposes)
! -----
!
if (iarg .lt. 5) then
  esum(iql+2,irl,iarg+1) = esum(iql+2,irl,iarg+1) + edepwt
  nsum(iql+2,irl,iarg+1) = nsum(iql+2,irl,iarg+1) + 1

! added SJW for particle by particle energy balance
  if(irl.eq.nreg) then
    eparte = eparte + edepwt
  else
    epartd = epartd + edepwt
  endif
end if

!
! -----
! Score energy deposition inside NaI detector
! -----
!
if (med(irl).eq. 1) then
  depe = depe + edepwt

!
! -----
! Score particle information if it enters from outside
! -----
!
if (irl .ne. irold .and. iarg .eq. 0) then
  if (iql .eq. 0) then           ! photon

```

```

        ie = e(np)/deltae + 1
        if(ie .gt. 50) ie = 50
        spg(1,ie) = spg(1,ie) + wt(np)
    elseif (iql .eq. -1) then          ! electron
        ie = (e(np) - RM)/deltae + 1
        if(ie .gt. 50) ie = 50
        spe(1,ie) = spe(1,ie) + wt(np)
    else                                ! positron
        ie = (e(np) - RM)/deltae + 1
        if(ie .gt. 50) ie = 50
        spp(1,ie) = spp(1,ie) + wt(np)
    end if
end if
end if

! -----
! Print out stack information (for limited number cases and lines)
! -----
if (ncount .le. nwrite .and. ilines .le. nlines) then
    ilines = ilines + 1
    write(6,101) e(np),x(np),y(np),z(np),u(np),v(np),w(np),
*           iql,irl,iarg
101  FORMAT(4G15.7/3G15.7,3I5)
end if

! -----
! Print out particle transport information (if switch is turned on)
! -----
if (iwatch .gt. 0) call swatch(iarg,iwatch)

return
end

!-----last line of ausgab.f-----
!-----howfar.f-----
! Version: 040727-1300
! 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

include 'include/egs5_h.f'
include 'include/egs5_epcont.f'
include 'include/egs5_stack.f'
include 'include/egs5_thresh.f'

! include 'user_auxcommons/aux_h.f'
include 'user_auxcommons/cg/tvalcg.f'
include 'user_auxcommons/cg/zondta.f'
include 'user_auxcommons/cg/rppdta.f'
include 'user_auxcommons/cg/sphdtac.f'
include 'user_auxcommons/cg/rccdta.f'
include 'user_auxcommons/cg/trcdta.f'
include 'user_auxcommons/cg/tordta.f'

real*8 atvaltmp,xidd,yidd,zidd          ! Local variables
real delhow,tval,tval0,tval10,tval00,tvalmn,udotau,udotav,
*   udotaw,xiss,xl,yiss,yl,ziss,zl

integer i,ihitcg,irl,irlfg,irlold,irnear,irnext,itvlfj,j,jjj

```

```

IRL=IR(NP)
IF (IRL.LT.1.OR.IRL.GE.IZONIN) THEN
  IDISC=1
  RETURN
END IF
TVAL=1.E+30
ITVALM=0
DO I=1,NBBODY(IRL)
  DO J=1,IRPPIN
    IF (ABS(NBZONE(I,IRL)).EQ.NBRPP(J)) THEN
      UDOTAU=U(NP)
      UDOTAV=V(NP)
      UDOTAW=W(NP)
      XL=X(NP)
      YL=Y(NP)
      ZL=Z(NP)
      CALL RPPCG1(J,XL,YL,ZL,UDOTAU,UDOTAV,UDOTAW)
    END IF
  end do
  DO J=1,ISPHIN
    IF (ABS(NBZONE(I,IRL)).EQ.NBSPH(J)) THEN
      UDOTAU=U(NP)
      UDOTAV=V(NP)
      UDOTAW=W(NP)
      XL=X(NP)
      YL=Y(NP)
      ZL=Z(NP)
      CALL SPHCG1(J,XL,YL,ZL,UDOTAU,UDOTAV,UDOTAW)
    END IF
  end do
  DO J=1,IRCCIN
    IF (ABS(NBZONE(I,IRL)).EQ.NBRCC(J)) THEN
      UDOTAU=U(NP)
      UDOTAV=V(NP)
      UDOTAW=W(NP)
      XL=X(NP)
      YL=Y(NP)
      ZL=Z(NP)
      CALL RCCCG1(J,XL,YL,ZL,UDOTAU,UDOTAV,UDOTAW)
    END IF
  end do
  DO J=1,ITRCIN
    IF (ABS(NBZONE(I,IRL)).EQ.NBTRC(J)) THEN
      UDOTAU=U(NP)
      UDOTAV=V(NP)
      UDOTAW=W(NP)
      XL=X(NP)
      YL=Y(NP)
      ZL=Z(NP)
      CALL TRCCG1(J,XL,YL,ZL,UDOTAU,UDOTAV,UDOTAW)
    END IF
  end do
  DO J=1,ITORIN
    IF (ABS(NBZONE(I,IRL)).EQ.NBTOR(J)) THEN
      UDOTAU=U(NP)
      UDOTAV=V(NP)
      UDOTAW=W(NP)
      XL=X(NP)
      YL=Y(NP)
      ZL=Z(NP)
      CALL TORCG1(J,XL,YL,ZL,UDOTAU,UDOTAV,UDOTAW)
    END IF
  end do
end do
IRNEAR=IRL
IF (ITVALM.EQ.0) THEN
  TVALO=1.E-4
  XISS=X(NP)+TVALO*U(NP)
  YISS=Y(NP)+TVALO*V(NP)
  ZISS=Z(NP)+TVALO*W(NP)
2291 IF(X(NP).NE.XISS.OR.Y(NP).NE.YISS.OR.Z(NP).NE.ZISS) GO TO 2292
  TVALO=TVALO*10.
  XISS=X(NP)+TVALO*U(NP)
  YISS=Y(NP)+TVALO*V(NP)
  ZISS=Z(NP)+TVALO*W(NP)
2292 GO TO 2291
CONTINUE
XIDD=DBLE(X(NP))+DBLE(TVALO)*DBLE(U(NP))
YIDD=DBLE(Y(NP))+DBLE(TVALO)*DBLE(V(NP))
ZIDD=DBLE(Z(NP))+DBLE(TVALO)*DBLE(W(NP))
CALL SRZONE(XIDD,YIDD,ZIDD,IRNEXT)
IF (IRNEXT.NE.IRL) THEN

```

```

TVAL=0.0
IRNEAR=IRNEXT
ELSE
TVAL00=0.0
TVAL10=10.0*TVAL0
IRLOLD=IRL
IRLFG=0
2301 IF (IRLFG.EQ.1) GO TO 2302
      TVAL00=TVAL00+TVAL10
      IF (TVAL00.GT.1.0E+06) THEN
        WRITE(6,2310) IQ(NP),IR(NP),X(NP),Y(NP),Z(NP),U(NP),V(NP),
*         W(NP),TVAL00
2310 *   FORMAT(' TVAL00 ERROR : IQ,IR,X,Y,Z,U,V,W,TVAL=',2I3,
*         1P7E12.5)
        STOP
      END IF
      XIDD=DBLE(X(NP))+DBLE(TVAL00)*DBLE(U(NP))
      YIDD=DBLE(Y(NP))+DBLE(TVAL00)*DBLE(V(NP))
      ZIDD=DBLE(Z(NP))+DBLE(TVAL00)*DBLE(W(NP))
      CALL SRZOLD(XIDD,YIDD,ZIDD,IRLOLD,IRLFG)
      GO TO 2301
2302 CONTINUE
      TVAL=TVAL00
      DO J=1,10
        XIDD=DBLE(X(NP))+DBLE(TVAL00)*DBLE(U(NP))
        YIDD=DBLE(Y(NP))+DBLE(TVAL00)*DBLE(V(NP))
        ZIDD=DBLE(Z(NP))+DBLE(TVAL00)*DBLE(W(NP))
        CALL SRZONE(XIDD,YIDD,ZIDD,IRNEXT)
        IF (IRNEXT.NE.IRLOLD) THEN
          TVAL=TVAL00
          IRNEAR=IRNEXT
        END IF
        TVAL00=TVAL00-TVAL0
      end do
      IF (IRL.EQ.IRNEAR) THEN
        WRITE(0,*) 'IRL,TVAL=',IRL,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
        END IF
      end do
    end do
    ITVFLG=0
    TVALMN=TVAL
    DO JJJ=1,ITVALM
      IF (TVALMN.GT.ATVAL(JJJ)) THEN
        TVALMN=ATVAL(JJJ)
      END IF
      DELHOW=1.E-4
      TVAL0=ATVAL(JJJ)+DELHOW
      XISS=X(NP)+TVAL0*U(NP)
      YISS=Y(NP)+TVAL0*V(NP)
      ZISS=Z(NP)+TVAL0*W(NP)
2361 IF(X(NP).NE.XISS.OR.Y(NP).NE.YISS.OR.Z(NP).NE.ZISS) GO TO 2362
        DELHOW=DELHOW*10.
        TVAL0=ATVAL(JJJ)+DELHOW
        XISS=X(NP)+TVAL0*U(NP)
        YISS=Y(NP)+TVAL0*V(NP)
        ZISS=Z(NP)+TVAL0*W(NP)
2362 GO TO 2361
      CONTINUE
      XIDD=DBLE(X(NP))+DBLE(TVAL0)*DBLE(U(NP))
      YIDD=DBLE(Y(NP))+DBLE(TVAL0)*DBLE(V(NP))
      ZIDD=DBLE(Z(NP))+DBLE(TVAL0)*DBLE(W(NP))
      CALL SRZONE(XIDD,YIDD,ZIDD,IRNEXT)
      IF ((IRNEXT.NE.IRL.OR.ATVAL(JJJ).GE.1.).AND.TVAL.GT.
*     ATVAL(JJJ)) THEN
        TVAL=ATVAL(JJJ)
        IRNEAR=IRNEXT
        ITVFLG=1
        GOTO 2370
      END IF
    end do
2370 IF (ITVFLG.EQ.0) THEN
      TVAL0=1.E-4
      XISS=X(NP)+TVAL0*U(NP)

```

```

      YISS=Y(NP)+TVALO*V(NP)
      ZISS=Z(NP)+TVALO*W(NP)
2381  IF(X(NP).NE.XISS.OR.Y(NP).NE.YISS.OR.Z(NP).NE.ZISS) GO TO 2382
      TVALO=TVALO*10.
      XISS=X(NP)+TVALO*U(NP)
      YISS=Y(NP)+TVALO*V(NP)
      ZISS=Z(NP)+TVALO*W(NP)
2382  GO TO 2381
      CONTINUE
      IF (TVALMN.GT.TVALO) THEN
        TVAL=TVALMN
      ELSE
        TVAL=TVALO
      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(6,2390) IQ(NP),IR(NP),X(NP),Y(NP),Z(NP),U(NP),V(NP),W(NP)
          ,TVAL
2390  *   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
      END IF
      RETURN
      END

```

!-----last line of subroutine howfar-----