egs5 サンプルプログラム (ucnaicgv.f)
NaI 検出器の応答計算 (cg Version)
(CGview による飛跡表示オプション付)
(September 8 2005, 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. Cobinatrial Geometry (CG)

1.1. Body の定義

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 $\theta 1$ と終わりの角度 $\theta 2$ を指定する。トーラス全体を使用する場合には、$\theta 1$=0, 及び $\theta 2$=2π とする。

Table 1 Data requied 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	
		$\theta 1$	$\theta 2$	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 の内側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 の外側の領域がリージョンとなる。body 番号の後に+又は – 記号と body 番号が続く場合には、間に AND 記号あるのと同じである。従って、+1 +2 は、body 1 の内側でなおかつ body 2 の内側を意味するので、body 1 と body 2 の重なった領域となる。一方、+1 –2 は、body 1 の内側でなおかつ body 2 の外側を意味するので、body 1 の領域中で body 2 と重なっていない領域を意味することになる。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. Torii and T. Sugita[1] の Appendix A を参照

1.3. リージョン定義の例

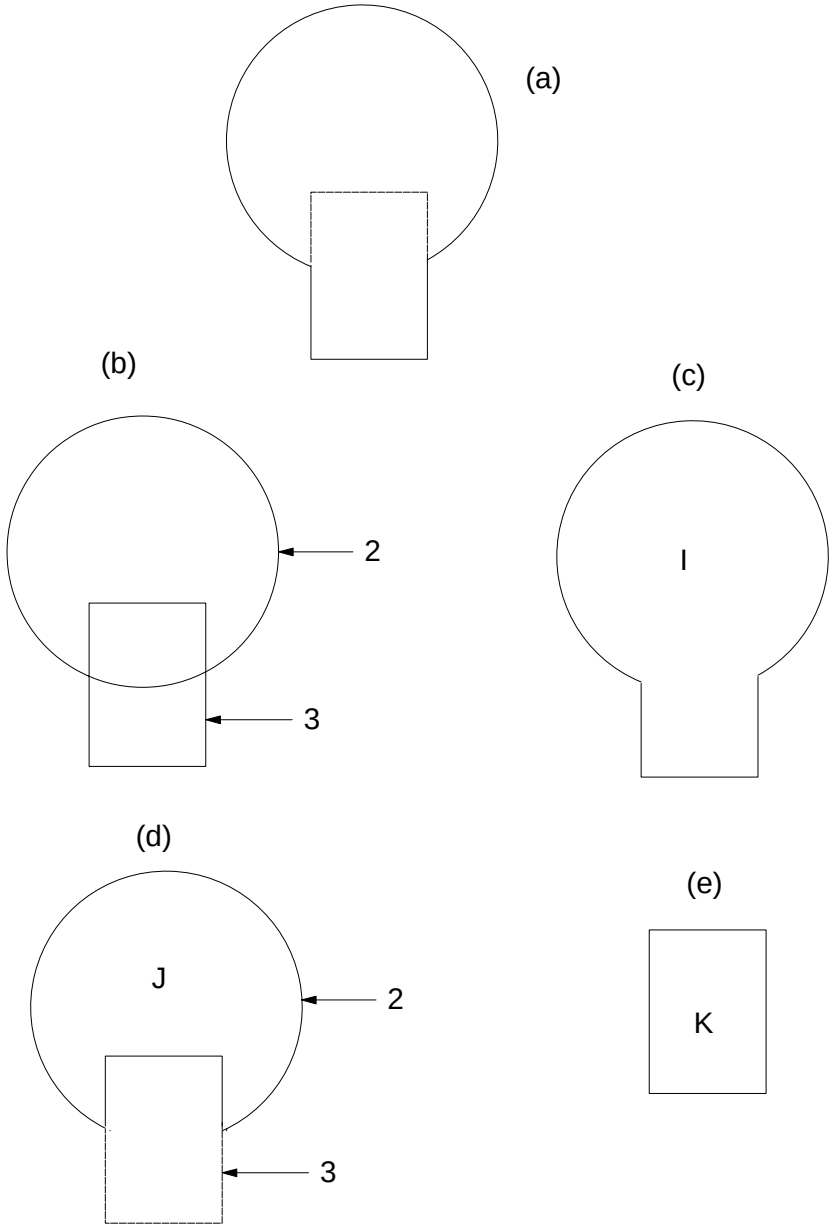

Figure 1: Examples of Combinatorial Geometry Method.

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、

$$\mathrm{I} = +2\mathrm{OR} + 3$$

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事を意味している。

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、

$$\mathrm{J} = +2 - 3$$

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

$$\mathrm{K} = +3$$

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。

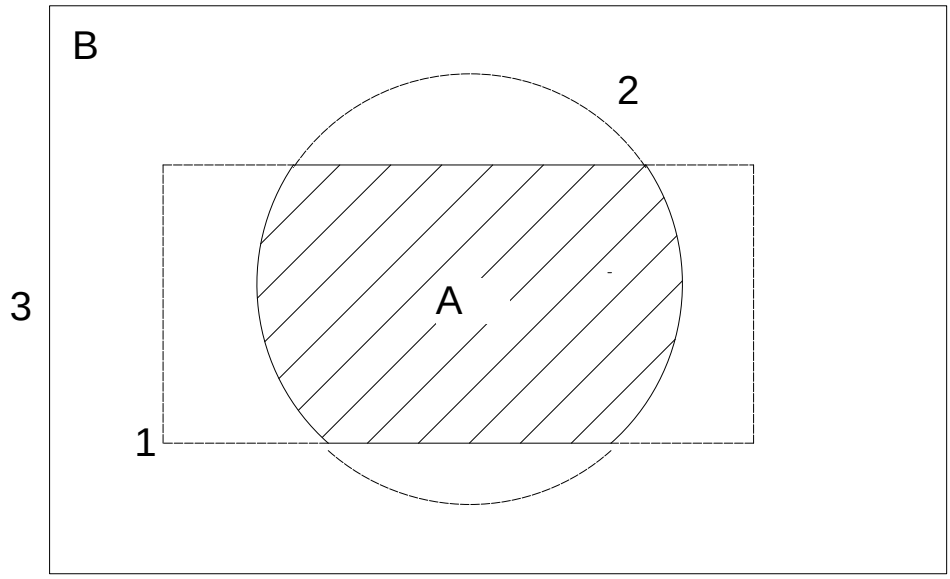

Figure 2: Use of OR operator.

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、

$$\mathrm{A} = +1 + 2$$

そして

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

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 ucnaicgv.fの概要

ucnaicgv.fは、CG を使って形状を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

ucnai.fr 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。

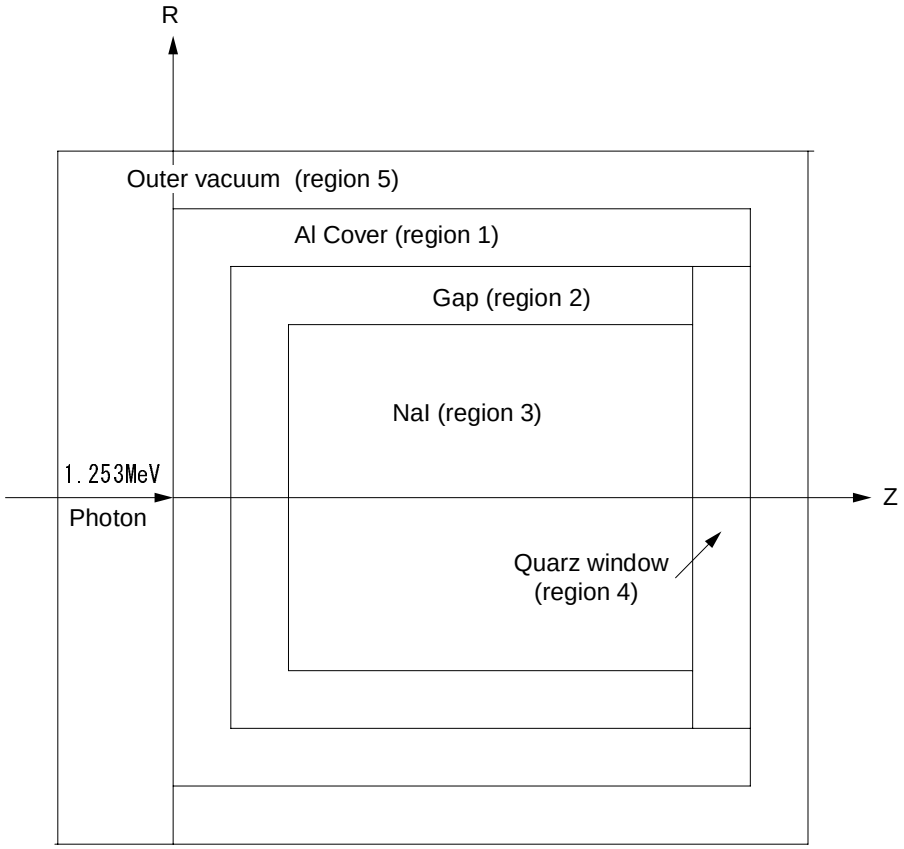

Figure 3: Geometry of ucnaicgv.f

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
  2    0    1    3    0
```

最後の行は、対応するリージョンの物質番号である。

1. 線源条件
   - 入射粒子は、エネルギー 1.253MeV の光子
   - 入射粒子の角度は、キーボード入力により、原点への垂直入射と、点等方線源とすることができる。
2. 得られる情報
   - CGview 用の飛跡データ（飛跡モードを選択した場合）
   - 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: Step 1

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 に関係している include ディレクトリ、pegs に関係している pegscommons と、egs5 の著者から提供しているジオメトリー関係のサブルーティン等ユーザーコードにのみ関係する auxcommons 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、egs5 に直接関係する配列の大きさを変更場合は、/include/egs5_h.f 内の、その他の場合は、/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_bounds.f'
      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等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 '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/watch.f'

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

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

最後の include 文が、CG に関連したもので、CG を使用する場合には常にこの表現とする。
個々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,deltae,spg(1,50),spe(1,50),spp(1,50),nreg
      real*8 depe,deltae,spg,spe,spp
      integer 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。

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

このユーザーコードでは、ユニット 6 は、コンソール出力として使用しているので、結果の出力は、ユニット 1 に設定している。ユニット 39 は、飛跡情報の出力ファイルである。

その後、各カウンターをリセットするサブルーティン counters_out(0) を call する。

3.2. メインプログラム: Step 2

物質データ及び各物質の Characteristic Distance を設定した後で、pegs5 を call する。Chracteristic Distance は、当該物質で構成されるリージョンの最も小さいサイズ (1cmx1cmx1cm の立方体であれば、1cm) に設定する。

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

!     ---------------------------------
!     define media before calling PEGS5
!     ---------------------------------

      nmed=3
      medarr(1)='NAI                     '
      medarr(2)='AL                      '
      medarr(3)='QUARTZ                  '

      do j=1,nmed
        do i=1,24
```

```
          media(i,j)=medarr(j)(i:i)
        end do
      end do

      chard(1) = 3.81d0       !  optional, but recommended to invoke
      chard(2) = 0.1d0        !  automatic step-size control
      chard(3) = 0.5d0

      write(1,*) 'chard =',(chard(j),j=1,3)

!     ---------------------------------
!     Run KEK PEGS5 before calling HATCH
!     ---------------------------------
      write(1,100)
100   FORMAT(' PEGS5-call comes next'/)

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

3.3. メインプログラム: Step 3

最初に、cg に関係したパラメーターを設定し、CG データの開始を示す CSTA を書き込み、その後 CG の入力データを読み込み、cg データの処理を行うサブルーティン geomgt を call する。CG を使用する場合には、この部分は変更する必要はない。CG データは、表示用ファイルに書く必要があるので、出力ユニットである ifto は、39 に設定している。その後、CG データの終了を意味する CEND を出力後、cg データから、リージョン総数である nreg を引き出す。

```
!-----------------------------------------------
!     initilize cg related parameter
!-----------------------------------------------
      npreci=2     ! PICT data mode for CGView

      itbody=0
      irppin=0
      isphin=0
      irccin=0
      itorin=0
      itrcin=0
      izonin=0
      itverr=0
      igmmax=0
      ifti = 4     ! Input unit number for cg-data
      ifto = 39    ! Output unit number for PICT

      write(39,100)
100   FORMAT('CSTA')
      call geomgt(ifti,ifto)
      write(39,110)
110   FORMAT('CEND')

!--------------------------------
!     Get nreg from cg input data
!--------------------------------
      nreg=izonin
      if (nreg.gt.mxreg) then
        write(1,90) nreg,mxreg
90      FORMAT(' NREG(=',I12,') must be less than MXREG(=',I12,')' /
     *  ' You must change MXREG in include/egs5_h.f.')
        stop
      end if
```

各リージョンの物質番号、egs カットオフエネルギー、オプションの設定（このユーザーコードでは、NaI の領域で特性 X 線を発生させる）を行う。物質番号の設定は、cg データの最後で定義したものと対応していなければならない。

最大エネルギー及びカットオフエネルギーの電子に対するエネルギーヒンジの長さを決める estepe 及び estepe2 を設定する。エネルギーヒンジは、結果の大きい影響を与えないので、通常はこの例の値で良い。

Ranlux 乱数のシード inseed の値を設定し、初期化する。

```
!   Set medium index for each region
!     Vacuum region
      med(nreg)=0
      med(2)=0      ! Inside vacuum
!     Al region
      med(1)=2

!     NaI detector region
      med(3)=1
      iedgfl(3)=1   ! 1:Produce flourscent X-rays
                    ! 0:Flourscent X-ray is not produced

!     Quartz region
      med(4)=3

      do i=1,4
        if(i.ne.2) ecut(i)=0.561
      end do

!     --------------------------------
!     Set parameter estepe and estepe2
!     --------------------------------
      estepe=0.10
      estepe2=0.20
      write(1,140) estepe, estepe2
140   FORMAT(1X,'ESTEPE at EKMAX: ',F10.5,' (estepe)',
     *       /,1X,'ESTEPE at ECUT:  ',F10.5,' (estepe2)')

!     ------------------------------------------------------
!     Random number seeds.  Must be defined before call hatch
!     or defaults will be used.  inseed (1- 2^31)
!     ------------------------------------------------------
      luxlev = 1
      inseed=1
      write(1,150) inseed
150   FORMAT(/,' inseed=',I12,5X,
     *         ' (seed for generating unique sequences of Ranlux)')

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

3.4. メインプログラム: Step 4

入射粒子のパラメータを設定する。この例では、単一エネルギーの光子 (1.253MeV) が、検出器の中心に入射するとしている。入射粒子の方向については、垂直入射とするか、入射点から等方にするかをコンソール入力で指定出来るようにしてる。

```
! Define initial variables for incident particle normally incident
! on the slab
      iqin=0              ! Incident charge - photons
      ekein=1.253         ! Kinetic energy
      xin=0.0             ! Incident at origin
      yin=0.0
      zin=0.0
      uin=0.0             ! Moving along z axis
      vin=0.0
      win=1.0
      irin=2              ! Starts in region 2, could be 1
      wtin=1.0            ! Weight = 1 since no variance reduction used

      deltae=0.05         ! Energy bin of response

! --------------------------
! Set isotropic source flag
! --------------------------
      write(6,160)
```

```
160   format('Key in source type. 0:monodirectional, 1:point isotropic')
      read(5,*) isot
```

3.5. メインプログラム: Step 5

最大電子エネルギー（全エネルギー）を表す emaxe を設定後に subroutine hatch を call する。

hatch で読み込まれた物質データや、リージョンに設定した情報を確認のために出力するようにしている。

飛跡用のファイルに、各リージョンの物質番号を出力した後、表示モードをキーボードから指定する。飛跡表示モード (0) か、計算モード (1) の選択が出来る。

```
        write(39,310)
310     FORMAT('MSTA')
        write(39,320) nreg
320     FORMAT(I4)
        write(39,330) (med(i),i=1,nreg)
330     FORMAT(15I4)
        write(39,340)
340     FORMAT('MEND')

!-----------------------------------
!     Selection mode form Keyboard.
!-----------------------------------
      write(6,350)
350   FORMAT(' Key in mode. 0:trajectory display, 1:dose calculation')
      read(5,*) imode
```

3.6. メインプログラム: Step 6

普通のユーザーコードでは、このステップで形状に関する情報（平板、円筒、球等）を記述するが、本ユーザーコードでは cg で形状を指定しているので、このステップで記述する事項はない。

3.7. メインプログラム: Step 7

ausgab に必要な設定を行う。エネルギービンの幅を、入射エネルギーとビン数 (50) から計算する。求めたい各データの初期化の後、計算したいヒストリー数をキーボードから入力する。飛跡表示モードを選択した場合には、ヒストリー数を少なくする場合も多いので、バッチ数 (nbatch) も指定するようにしている。計算モードの場合は、バッチ数は 50 に固定している。

```
!     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 histories, number of batch and histories per batch
      write(6,*) (' Key in number of cases.')
      read(5,*) ncases
      if(imode.eq.0) then
         write(6,*) (' Key in number of batch (=<50).')
         read(5,*)  nbatch
      else
        nbatch = 50
      end if
```

```
      ncaspb = ncases / nbatch
```

3.8. メインプログラム: Step 8

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。

各バッチの先頭で、飛跡表示ファイルにバッチ番号を出力する。

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。

飛跡ファイルに残っているデータとバッチ終了の情報を出力する。

```
                                          ! -------------------------
      do nofbat=1,nbatch                  ! Start of batch -loop
                                          ! -------------------------
         write(39,370) nofbat
370      format('0',I5)

         do icases=1,ncaspb               ! Start of CALL SHOWER loop

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

           ekin = ekein
           wtin = 1.0

           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.
380          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 380
             win = zi0/rr0
             uin = xi0/rr0
             vin = yi0/rr0
           end if

!        ------------------------------------------------
!        Print first NWRITE or NLINES, whichever comes first
!        ------------------------------------------------
           if (ncount .le. nwrite .and. ilines .le. nlines) then
             ilines = ilines + 1
             write(1,390) etot,xin,yin,zin,uin,vin,win,iqin,irin,idin
390          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(1,400) icases, eparte, epartd
400          FORMAT('Error on # ',I6,' Escape = ',F9.5,
```

```
     *             ' Deposit = ',F9.5)
          end if

!       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. ekein*0.999) pef=pef +wtin
            depe = 0.D0
          end if

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

                                                ! ----------------------
        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

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

        write(39,410)               ! Set end of batch for CG View
410     FORMAT('9')

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

3.8.1. 統計誤差: x をモンテカルロ計算で計算したい量（スコアーする量）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-CG 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

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{1}$$

より求める。

- x_i の分散を以下の式で計算する。

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

- $\overline{x}$ の統計誤差は、

$$s_{\overline{x}}^2 = \frac{s_x^2}{n} \tag{3}$$

  となる。

- FSD(fractional standard deviation) として

$$\mathrm{FSD} = s_{\overline{x}}/\overline{x} \tag{4}$$

  を用いる。

3.9. メインプログラム: Step 9

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(1,470) avpe,sigpe
470   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(1,480) avte,sigte
480   FORMAT(' Total efficiency =',G15.5,'+-',G15.5,' %')
```

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。

3.10. 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. 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
```

3.11. 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。

CG を使用するユーザーコードでは、常に ucnaicgv.f の cg ルーチン用 howfar を使用する。

4. ucnai.f と ucnaicgv.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 に較べ、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、円筒平板形状を使用している ucnai.f と ucnaicgv.f で全く同じ条件の計算を行うと、ucnai.f の方が 1.6 倍速いという結果が得られている。[§]

[§]CG のスピードアップ（使用している形状により効果は異なる）は、鳥居、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. 1.253MeV 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。
2. 線源を、Cs-137 の単一エネルギー光子 (0.662MeV) に変える。
3. 線源を、Co-60 に変え、1.173MeV と 1.333MeV 光子を同じ確率で発生させる。
4. 1.253MeV 線源で、検出器の有感領域の厚さを 2 倍する。

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、1.253MeV 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、1.253MeV 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、1.602×10^{-19}C/e とする。

6. 実習課題の解答例

比較のために、ucnaicgv.f の計算モードで、ヒストリー数 10000 の場合の結果 (egs5job.out) を別な名称のファイル名 (例えば、nai.out) で保存しておく。

6.1. 実習課題1

1. Point source

   - ucnaicgv.f を egs5run で実行する。
   - 線源の種類の選択で、1 の等方線源を選択する。
     1 を選択すると、478 行目からの以下のサンプリングが有効になり、粒子の方向が等方的になる。

```
380         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 380
            win = zi0/rr0
            uin = xi0/rr0
            vin = yi0/rr0
```

   - 上記で使用されているのは、”Rejection 法”と呼ばれている手法で、半径 1cm の半球内の任意の点をサンプリングし、その方向を粒子の方向としている。
   - モードの選択で、0 の飛跡表示モードを選ぶ。
   - ヒストリー数として、100 を、バッチ数として 2 を入力する。
   - 計算が終了したら、CGview の”体系・飛跡データの読み込み”で、egs5job.pic を選択し、等方線源となっていることを確認する。
   - egs5job を入力し、再度実行する。モードの選択で、1 の計算モードを選び、ヒストリー数として、10000 を入力する。
   - 計算が終了したら、egs5job.out を調べ、ビーム入射の場合と結果を比較する。

2. ^{137}Cs source

   - cp ucnaicgv.f ucnaicgv1.f
   - ucnaicgv1.f の 279 行目の ekein の値を 0.662 に変更する。
   - cp ucnaicgv.data ucnaicgv1.data
   - cp ucnaicgv.inp ucnaicgv1.inp
   - ucnaicgv1.f を egs5run で実行する。
     ユニット 4、ユニット 25 のファイル名は、何も入力しないでリターンする。
   - 線源の種類の選択で、0 の一方向線源を選択する。
   - モードの選択で、1 の計算モードを選ぶ。
   - ヒストリー数として、10000 を入力する。
   - 計算が終了したら、egs5job.out を調べ、1.253MeV の結果と比較する。

3. ^{60}Co source

   - cp ucnaicgv.f ucnaicgv2.f
   - ucnaicgv2.f を以下のように変更する。
     - 新たなローカル変数として 121 行に ekein0 を追加する。
     - 279 行を ekein=1.333 に変更する。
     - 各ヒストリー毎に、どちらのエネルギーになるかを決定する必要があるので、 469 行目に以下のサンプリングルーチンを挿入する。

```
call randomset(rnnow)
if(rnnow.le.0.5) then
  ekein0=1.173            ! lower energy photon
else
  ekein0=1.333            ! higher energy photon
end if
```

  - 次の ekin = ekein の ekein を ekein0 に変更する。
  - 610 行目からの入射エネルギー出力部を以下のように変更する。

```
      write(1,460)
460   FORMAT(' Results for Co-60 gamma-ray (1.173 and 1.333 MeV)'/)
```

- cp ucnaicgv.data ucnaicgv2.data
- cp ucnaicgv.inp ucnaicgv2.inp
- ucnaicgv2.f を egs5run で実行する。
  ユニット 4、ユニット 25 のファイル名は、何も入力しないでリターンする。
- 線源の種類の選択で、0 の一方向線源を選択する。
- モードの選択で、1 の計算モードを選ぶ。
- ヒストリー数として、10000 を入力する。
- 計算が終了したら、egs5job.out を調べ、1.253MeV の結果と比較する。波高分布に 2 つの入射エネルギーに対応したピークがあることを確認する。

4. 長さ 2 倍の NaI

- cp ucnaicgv.f ucnaicgv3.f
- cp ucnaicgv.data ucnaicgv3.data
- cp ucnaicgv.inp ucnaicgv3.inp
- ucnaicgv3.f を以下のように修正する。
  - 589 行目を tdet=7.62*2.0 に
  - この修正は、計算とは直接関係しないことである。CG を使用する場合、CG 入力データから検出器のサイズ等を直感できないので、どの様な形状の計算か判るようにするためのものである。実際の形状の変更は、ucnaicgv3.data で行う。
- ucnaicgv3.data を以下の様に変更する。

```
RCC    1       0.00          0.0          0.0          0.00         0.0
              16.34          4.41
RCC    2       0.00          0.0          0.1          0.00         0.0
              15.74          4.31
RCC    3       0.00          0.0          0.6          0.00         0.0
              15.24          3.81
RCC    4       0.00          0.0        15.84          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
```

- ucnaicgv3.f を egs5run で実行する。
  ユニット 4、ユニット 25 のファイル名は、何も入力しないでリターンする。
- 線源の種類の選択で、0 の一方向線源を選択する。
- モードの選択で、0 の計算モードを選ぶ。
- ヒストリー数 100、バッチ数 5 を入力する。
- 計算が終了したら、CGview の”体系・飛跡データの読み込み”で、egs5job.pic を選択し、検出器の長さが 2 倍になっていることを確認する。
- egs5job を入力し、再度実行する。モードの選択で、1 の計算モードを選び、ヒストリー数として、10000 を入力する。
- 計算が終了したら、egs5job.out を調べ、元の長さの結果と比較する。

6.2. 実習課題2

1. cp ucnaicgv.f ucnaicgv4.f
2. cp ucnaicgv.data ucnaicgv4.data
3. cp ucnaicgv.inp ucnaicgv4.inp
4. ucnaicgv.4.f の 169 行目を下記に変更する。

```
      medarr(1)='GE                          '
```

   この部分は、subroutine hatch で読み込む物質 1 として使用する"物質データ"の名称を指定する。使用する名称は、ユニット 25 から読み込む pegs5 の入力データで定義する物質の名称と一致していなければならない。順番は pegs5 の入力データ (ユニット 25 で使用するファイル) と違っていても良い。

5. ucnaicgv4.inp の 1 行目から 5 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る。

```
ELEM
 &INP EFRACH=0.05,EFRACL=0.20,
      IRAYL=1,IBOUND=0,INCOH=0,ICPROF=0,IMPACT=0 /END
GE                              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
```

6. ucnaicgv4.f を egs5run で実行する。
   ユニット 4、ユニット 25 のファイル名は、何も入力しないでリターンする。
7. 線源の種類の選択で、0 の一方向線源を選択する。
8. モードの選択で、1 の計算モードを選ぶ。
9. ヒストリー数として、10000 を入力する。
10. 計算が終了したら、egs5job.out を調べ、NaI の結果と比較する。

6.3. 実習課題3

1. cp ucnaicgv.f ucioncgv.f
2. cp ucnaicgv.data ucioncgv.data
3. cp ucnaicgv.inp ucioncgv.inp
4. ucioncgv.f を以下のように修正する。
   - ローカル変数の real*8 に、バッチ毎の、空気中での吸収エネルギー `depepb(50)` と計算結果の統計処理で使用する変数 `avab,sigab,depes` を追加する。(125 行目及び 121 行目)
   - `pegs5` を呼ぶ前の物質定義の部分を以下の様に変更する。(168 から 181 行目)

```
      nmed=2
      medarr(1)='AIR-AT-NTP              '
      medarr(2)='AL                      '

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

      chard(1) = 3.81d0       !  optional, but recommended to invoke
      chard(2) = 0.5d0        !  automatic step-size control
```

- 231 行目からの各リージョンへの物質の割り当て等の指定の部分を以下の様に修正する。

```
!   Set medium index for each region
!     Vacuum region
      med(nreg)=0
!     Al region
      med(1)=2

!     Air detector region
      med(2)=1
      iedgfl(2)=1   ! 1:Produce flourscent X-rays
                    ! 0:Flourscent X-ray is not produced
```

- 417 行目の次に、depes=0.D0 を、516 行目の後に depes=depes+depe を追加する。
- 542 行目の後に次の 2 行を加える。

```
        depepb(nofbat)=depes/ncaspb
        depes=0.D0
```

- 円筒と平板数の減少に伴い、計算終了後の形状に関する 585 行目からの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(1,450) tdet,rdet,tcov,rtcov
450   FORMAT(/' Detector length=',G15.5,' cm'/
     *       ' Detector radius=',G15.5,' cm'/
     *       ' Al cover thickness=',G10.2,' cm'/
     *       ' Al cover side thickness=',G10.2,' cm'/)
```

- 全検出効率の出力の後ろに、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
      desci2 = 0.D0
      do j = 1, nbatch
        avab = avab + depepb(j)/nbatch
        desci2 = desci2 + depepb(j)*depepb(j)/nbatch
      end do
      sigab = sqrt((desci2 - avab*avab)/(nbatch-1))
      write(1,482) avab,sigab
482   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(1,484) avab,sigab
484   FORMAT(' Output current =',G15.5,'+-',G15.5,' C/photon')
```

5. ucioncgv.data を以下のように変更する。

```
  RCC    1       0.00          0.0          0.0          0.00         0.0
                 8.62         4.31
  RCC    2       0.00          0.0          0.5          0.00         0.0
                 7.62         3.81
  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
    2    1    0
```

6. ucnioncgv.inp を以下のように変更する。

```
MIXT
 &INP NE=3,RHO= 1.2050E-03,RHOZ= 0.78,0.2103,0.0094,
      EFRACH=0.05,EFRACL=0.20,IRAYL=1,IBOUND=0,INCOH=0,
      GASP=0.93174,ICPROF=0,IMPACT=0 /END
AIR-AT-NTP                      AIR-GAS
N  O  AR
ENER
 &INP AE=0.521,AP=0.010,UE=2.511,UP=2.0 /END
PWLF
 &INP  /END
DECK
 &INP  /END
ELEM
 &INP EFRACH=0.05,EFRACL=0.20,
      IRAYL=1,IBOUND=0,INCOH=0,ICPROF=0,IMPACT=0 /END
AL                              AL
AL
ENER
 &INP AE=0.521,AP=0.010,UE=2.511,UP=2.0 /END
TEST
 &INP  /END
PWLF
 &INP  /END
DECK
 &INP  /END
```

7. ucioncgv.f を egs5run で実行する。
   ユニット 4 及びユニット 25 のファイル名は、何も入力しないでリターンする。
8. 線源の種類の選択で、0 の一方向線源を選択する。
9. モードの選択で、0 の飛跡表示モードを選ぶ。
10. ヒストリー数 1000、バッチ数 1 を入力する。
11. 計算が終了したら、CGview の”体系・飛跡データの読み込み”で、egs5job.pic を選択し、形状及び飛跡が NaI の場合と違っていることを確認する。
12. egs5job を入力し、再度実行する。モードの選択で、1 の計算モードを選び、ヒストリー数として、100000 を入力する。
13. 計算が終了したら、egs5job.out を調べ、計算したい情報が得られていることを確認する。

References

[1] T. Torii and T. Sugita, "Development of PRESTA-CG Incorprating Combinatorial Geometry in EGS4/PRESTA", *JNC TN1410 2002-201*, Japan Nuclear Cycle Development Institute (2002).

Appendix 1 Full listings of ucnaicgv.f

```
!*********************************************************************
!***************************  KEK, High Energy Accelerator Research  *
!***************************  Organization                           *
!*** u c n a i c g v *********                                       *
!***************************     EGS5.0 USER CODE - 16 Jul 2005/1300 *
!*********************************************************************
!* This is a general User Code based on the cg geometry scheme.       *
!*********************************************************************
!                                                                     *
!  PROGRAMMERS:  H. Hirayama                                          *
!                Radiation Science Center                             *
!                Applied Science Laboratory                           *
!                KEK, High Energy Accelerator Research Organization   *
!                1-1, Oho, Tsukuba, Ibaraki, 305-0801                 *
!                Japan                                                *
!                                                                     *
!                E-mail:      hideo.hirayama@kek.jp                   *
!                Telephone:   +81-29-864-5489                         *
!                Fax:         +81-29-864-1993                         *
!                Based on ucrtz_sampl4 by Nelson and James.           *
!                                                                     *
!*********************************************************************
!*********************************************************************
! The ucnaicgv.f User Code requires a cg-input file only              *
! (e.g., ucnaicgv.data).                                              *
! The following shows the geometry for ucnaicg.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 corresponds to ucnai3cgp.mor for egs4.               *
! Use Ranlux random number generator.                                 *
!*********************************************************************
!                                                                     *
!             ------------------------                                *
!             cg Geometry (ucnaicgv)                                  *
!             ------------------------                                *
!                                                                     *
!                                                                     *
!                R                                                    *
!                ^                                                    *
!                |                                                    *
!           +----+----+----+--------+------+---                       *
!           |                                                         *
!           |     Outer vacuum region                                 *
!           +    +----+----+--------+------+--- r=4.41 cm             *
!           |    |             Al            |                        *
!           |    +    +----+--------+------+--- R=4.31                *
!           |    |    |        Gap   |      |                        *
!           +    +    +    +---------+      +--- R=3.81              *
!           |    |    |    |         |      |                        *
!           |    | Al | Gap|   NaI   |Quartz|                        *
! 1.253 MeV |    |    |    |         |      |                        *
!      ==========>+----+----+---------+------+--------> Z             *
!     photons    0   0.1  0.6       8.22   8.72 cm                   *
!                                                                     *
!                                                                     *
!                                                                     *
!*********************************************************************
!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'

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

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

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

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

      real*8                                               ! Local variables
     * availke,avpe,avph,avspe,avspg,avspp,avte,desci2,ekin,pef,
     * rr0,sigpe,sigte,sigph,sigspg,sigspe,sigspp,tef,wtin,wtsum,
     * xi0,yi0,zi0

      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
     * i,icases,idin,ie,ifti,ifto,ii,iiz,imed,ireg,isam,
     * isot,izn,nlist,j,k,n,ndet,nd,nbatch,ncaspb,
     * ner,nofbat

      character*24 medarr(3)

!     ----------
!     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 getcg etc. Unit for pict must be 39.
!-----------------------------------------------------------------

      open(1,FILE='egs5job.out',STATUS='unknown')
      open(UNIT= 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=3
      medarr(1)='NAI                     '
      medarr(2)='AL                      '
      medarr(3)='QUARTZ                  '

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

      chard(1) = 3.81d0       !  optional, but recommended to invoke
      chard(2) = 0.1d0        !  automatic step-size control
      chard(3) = 0.5d0

      write(1,*) 'chard =',(chard(j),j=1,3)

!     ---------------------------------
!     Run KEK PEGS5 before calling HATCH
!     ---------------------------------
      write(1,130)
130   FORMAT(' PEGS5-call comes next'/)

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

!-----------------------------------------------------------------------
! Step 3: Pre-hatch-call-initialization
!-----------------------------------------------------------------------
!-----------------------------------------------
!     Initialize cg related parameter
!-----------------------------------------------
      npreci=2     ! PICT data mode for CGView

      itbody=0
      irppin=0
      isphin=0
      irccin=0
      itorin=0
      itrcin=0
      izonin=0
      itverr=0
      igmmax=0
      ifti = 4     ! Input unit number for cg-data
      ifto = 39    ! Output unit number for PICT

      write(39,100)
100   FORMAT('CSTA')
      call geomgt(ifti,ifto)
      write(39,110)
110   FORMAT('CEND')

!-------------------------------
!     Get nreg from cg input data
!-------------------------------
      nreg=izonin
      if (nreg.gt.mxreg) then
        write(1,120) nreg,mxreg
120     FORMAT(' NREG(=',I12,') must be less than MXREG(=',I12,')' /
     *  ' You must change MXREG in include/egs5_h.f.')
        stop
      end if
!   Set medium index for each region
!     Vacuum region
      med(nreg)=0
      med(2)=0      ! Inside vacuum
!     Al region
      med(1)=2

!     NaI detector region
      med(3)=1
```

```
      iedgfl(3)=1   ! 1:Produce flourscent X-rays
                    ! 0:Flourscent X-ray is not produced

!     Quartz region
      med(4)=3

      do i=1,4
        if(i.ne.2) ecut(i)=0.561
      end do

!     ------------------------------
!     Set parameter estepe and estepe2
!     ------------------------------
      estepe=0.10
      estepe2=0.20
      write(1,140) estepe, estepe2
140   FORMAT(1X,'ESTEPE at EKMAX: ',F10.5,' (estepe)',
     *       /,1X,'ESTEPE at ECUT:  ',F10.5,' (estepe2)')

!     ------------------------------------------------------
!     Random number seeds.  Must be defined before call hatch
!     or defaults will be used.  inseed (1- 2^31)
!     ------------------------------------------------------
      luxlev = 1
      inseed=1
      write(1,150) inseed
150   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
      iqin=0              ! Incident charge - photons
      ekein=1.253         ! Kinetic energy
      xin=0.0             ! Incident at origin
      yin=0.0
      zin=0.0
      uin=0.0             ! Moving along z axis
      vin=0.0
      win=1.0
      irin=1              ! Starts in region 2, could be 1
      wtin=1.0            ! Weight = 1 since no variance reduction used

      deltae=0.05         ! Energy bin of response

! -------------------------
! Set isotropic source flag
! -------------------------
      write(6,160)
160   format('Key in source type. 0:monodirectional, 1:point isotropic')
      read(5,*) isot

!-----------------------------------------------------------------------
! Step 5:   hatch-call
!-----------------------------------------------------------------------
!-----------------------------------------------------------------------
! Maximum total energy of an electron for this problem must be
! defined before hatch call
      emaxe = ekein + RM         ! photon

      write(1,170)
170   format(/' Start ucnaicgv '/
     *' 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(1,180)
180   FORMAT(/,' HATCH-call comes next',/)

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

```
!     ==========

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

! ------------------------------------------------------------
! Print various data associated with each media (not region)
! ------------------------------------------------------------
      write(1,190)
190   FORMAT(/,' Quantities associated with each MEDIA:')
      do j=1,nmed
        write(1,200) (media(i,j),i=1,24)
200     FORMAT(/,1X,24A1)
        write(1,210) rhom(j),rlcm(j)
210     FORMAT(5X,' rho=',G15.7,' g/cu.cm     rlc=',G15.7,' cm')
        write(1,220) ae(j),ue(j)
220     FORMAT(5X,' ae=',G15.7,' MeV    ue=',G15.7,' MeV')
        write(1,230) ap(j),up(j)
230     FORMAT(5X,' ap=',G15.7,' MeV    up=',G15.7,' MeV',/)
      end do

! ------------------------------------------------------
! Print media and cutoff energies assigned to each region
! ------------------------------------------------------
      do i=1,nreg
        if (med(i) .eq. 0) then
          write(1,240) i
240       FORMAT(' medium(',I3,')=vacuum')
        else
          write(1,250) i,(media(ii,med(i)),ii=1,24),ecut(i),pcut(i)
250       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(1,260) izn
260           FORMAT('   X-ray information for Z=',I3)
              write(1,270) (ekx(ii,izn),ii=1,10)
270           FORMAT('   K-X-ray energy in keV',/,
     *               4G15.5,/,4G15.5,/,2G15.5)
              write(1,280) (elx1(ii,izn),ii=1,8)
280           FORMAT('   L-1 X-ray in keV',/,4G15.5,/,4G15.5)
              write(1,290) (elx2(ii,izn),ii=1,5)
290           FORMAT('   L-2 X-ray in keV',/,5G15.5)
              write(1,300) (elx3(ii,izn),ii=1,7)
300           FORMAT('   L-3 X-ray in keV',/,4G15.5,/,3G15.5)
            end do
          end if
        end if
      end do

      write(39,310)
310   FORMAT('MSTA')
      write(39,320) nreg
320   FORMAT(I4)
      write(39,330) (med(i),i=1,nreg)
330   FORMAT(15I4)
      write(39,340)
340   FORMAT('MEND')

!-----------------------------------
!     Selection mode from Keyboard.
!-----------------------------------
      write(6,350)
350   FORMAT(' Key in mode. 0:trajectory display, 1:dose calculation')
      read(5,*) imode

!-----------------------------------------------------------------------
! 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)
!     ========================

      write(1,360)
360   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',/)

      ndet=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 histories, number of batch and histories per batch
      write(6,*) (' Key in number of cases.')
      read(5,*) ncases
      if(imode.eq.0) then
         write(6,*) (' Key in number of batch (=<50).')
         read(5,*)  nbatch
      else
        nbatch = 50
      end if
      ncaspb = ncases / nbatch

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

!-----------------------------------------------------------------------
! Step 8:  Shower-call
!-----------------------------------------------------------------------
                                          ! -------------------------
      do nofbat=1,nbatch                  ! Start of batch -loop
                                          ! -------------------------
         write(39,370) nofbat
370      format('0',I5)

         do icases=1,ncaspb               ! Start of CALL SHOWER loop

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

           ekin = ekein
           wtin = 1.0

           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.
380          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 380
            win = zi0/rr0
            uin = xi0/rr0
            vin = yi0/rr0
          end if
!
!-----------------------------------------
!     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)
            call rstnxt(iqin+2,0,irinn)
          else
            irinn=irin
          end if

!       ------------------------------------------------
!       Print first NWRITE or NLINES, whichever comes first
!       ------------------------------------------------
          if (ncount .le. nwrite .and. ilines .le. nlines) then
            ilines = ilines + 1
            write(1,390) etot,xin,yin,zin,uin,vin,win,iqin,irinn,idin
390         FORMAT(4G15.7/3G15.7,3I5)
          end if

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

!       Added for energy balance tests (SJW)
          if(DABS(eparte + epartd - ekin)/ekin .gt. 1.d-10) then
            write(1,400) icases, eparte, epartd
400         FORMAT('Error on # ',I6,' Escape = ',F9.5,
     *             ' Deposit = ',F9.5)
          end if

!       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. ekein*0.999) pef=pef +wtin
            depe = 0.D0
          end if

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

                                                ! ------------------------
        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

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

        write(39,410)               ! Set end of batch for CG View
410     FORMAT('9')

                                                ! ------------------
      end do                                    ! End of batch loop
                                                ! ------------------
      tt=etime(tarray)
      tt1=tarray(1)
      cputime=tt1-tt0
      write(1,420) cputime
420   format(' Elapsed Time (sec)=',G15.5)

!-----------------------------------------------------------------------
! Step 9:  Output-of-results
!-----------------------------------------------------------------------
      write(1,430) ncount,ncases,totke
430   FORMAT(/,' Ncount=',I10,' (actual cases run)',/,
     *       ' Ncases=',I10,' (number of cases requested)',/,
     *       ' TotKE =',G15.5,' (total KE (MeV) in run)')

      if (totke .le. 0.D0) then
        write(1,440) totke,availke,ncount
440     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(1,450) tdet,rdet,tcov,rtcov,tgap,rtgap
450   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'/)

      write(1,460) ekin
460   FORMAT(' Results for ',G15.5,'MeV photon'/)

!     ----------------------------------
!     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(1,470) avpe,sigpe
470   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(1,480) avte,sigte
480   FORMAT(' Total efficiency =',G15.5,'+-',G15.5,' %')

!     -------------------------
!     Pulse height distribution
!     -------------------------
      write(1,490)
490   FORMAT(/' Pulse height distribution ')
      do ie=1,50
        elow=deltae*(ie-1)
        eup=deltae*ie
        if (elow .gt. ekein ) go to 510

        avph = 0.D0
        desci2 = 0.D0
        do j = 1, nbatch
          avph = avph + phpb(ie,j)/nbatch
          desci2 = desci2 + phpb(ie,j)*phpb(ie,j)/nbatch
        end do
        sigph = sqrt((desci2 - avph*avph)/(nbatch-1))
        avph = avph/deltae
        sigph= sigph/deltae
        write(1,500) eup,avph,sigph
500     FORMAT(' E (upper-edge --',G10.4,' MeV )=',G15.5,'+-',G15.5,
     *         ' counts/MeV/incident');
       end do

510    continue

!     ------------------------------------------------------
!     Particle spectrum.  Incident particle spectrum to detector.
!     ------------------------------------------------------
      write(1,520)
520   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 540

!     ---------------------------------
!     Gamma spectrum per MeV per source
!     ---------------------------------

          avspg = 0.D0
          desci2 = 0.D0
          do j = 1, nbatch
            avspg = avspg + spgpb(nd,ie,j)/nbatch
            desci2 = desci2 + spgpb(nd,ie,j)*spgpb(nd,ie,j)/nbatch
          end do
          sigspg = sqrt((desci2 - avspg*avspg)/(nbatch-1))
          avspg = avspg/deltae
          sigspg= sigspg/deltae

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

          avspe = 0.D0
          desci2 = 0.D0
          do j = 1, nbatch
            avspe = avspe + spepb(nd,ie,j)/nbatch
            desci2 = desci2 + spepb(nd,ie,j)*spepb(nd,ie,j)/nbatch
          end do
          sigspe = sqrt((desci2 - avspe*avspe)/(nbatch-1))
          avspe = avspe/deltae
          sigspe= sigspe/deltae

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

          avspp = 0.D0
          desci2 = 0.D0
```

```
          do j = 1, nbatch
            avspp = avspp + spppb(nd,ie,j)/nbatch
            desci2 = desci2 + spppb(nd,ie,j)*spppb(nd,ie,j)/nbatch
          end do
          sigspp = sqrt((desci2 - avspp*avspp)/(nbatch-1))
          avspp = avspp/deltae
          sigspp= sigspp/deltae

          write(1,530) eup,avspg,sigspg,avspe,sigspe,avspp,sigspp
530       FORMAT(G10.5,' MeV--',3(G12.5,'+-',G12.5))
        end do
      end do

540   continue

      nlist=1

      if(imode.ne.0) then
        close(1,status='keep')
        open(6,file='egs5job.out',access='append')

!       ============================
        call ecnsv1(nlist,nreg,totke)
        call ntally(nlist,nreg)
!       ============================
      end if

600   continue
!     ====================
      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:
!
!   1) Score energy deposition
!   2) Score particle information enter to detector from outside
!   3) Print out particle transport information
!   4) call plotxyz if 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/etaly1.f'         ! Auxiliary-code COMMONs
      include 'auxcommons/lines.f'
      include 'auxcommons/ntaly1.f'
      include '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),imode
      real*8 depe,deltae,spg,spe,spp
```

```
      integer imode

      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(1,101) 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)
!                       ========================

!     -----------------------------------
```

```
!     Output particle information for plot
!     ------------------------------------
      if (imode.eq.0) then
        call plotxyz(iarg,np,iq(np),x(np),y(np),z(np),e(np),ir(np),
     *       wt(np))
      end if


      return

      end

!-------------------------last line of ausgab.f-----------------------
!-----------------------------howfar.f--------------------------------
! Version:   050716-1300
! 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
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,itvlfg,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 atvaltmp
      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
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,nbbody(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
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
  320   continue
c       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
        itvlfg=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
            itvlfg=1
            goto 425
          end if
        end do
  425   continue
        if(itvlfg.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---------------------
```