

**Lecture Notes of
Calculation of air absorbed dose
with ionization chamber
(cg Version)**

H. Hirayama and Y. Namito



High Energy Accelerator Reserach Organization

©High Energy Accelerator Research Organization (KEK), 2021

KEK Reports are available from

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

Phone: +81-29-864-5137
Fax: +81-29-864-4604
E-mail: irdpub@mail.kek.jp
Internet: <https://www.kek.jp/en/>

Lecture Notes of
Calculation of air absorbed dose
with ionization chamber
(cg Version)

H. Hirayama and Y. Namito



High Energy Accelerator Reserach Organization

Contents

Japanese Parts	1
1 Combinatorial Geometry (CG)	2
1.1 Body の定義	2
1.2 リージョンの定義	2
1.3 リージョン定義の例	3
2 サンプルプログラム ucionch_cgv.f の概要	5
2.1 CG 入力データ	5
3 ユーザーコードの内容	6
3.1 メインプログラム: Step 1	6
3.2 メインプログラム: Step 2	7
3.3 メインプログラム: Step 3	8
3.4 メインプログラム: Step 4	9
3.5 メインプログラム: Step 5	10
3.6 メインプログラム: Step 6	10
3.7 メインプログラム: Step 7	10
3.8 メインプログラム: Step 8	11
3.8.1 統計誤差	12
3.9 メインプログラム: Step 9	13
3.10 Subroutine ausgab	15
3.11 Subroutine howfar	16
3.12 function encoea	17
4 計算結果	17
5 実習課題	18
5.1 実習課題 1 : 線源光子のエネルギーを変えた計算	18
5.2 実習課題 2 : 0.662MeV 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、等方線源に変更した計算	18
6 実習課題の解答例	19
6.1 実習課題 1	19
6.2 実習課題 2	20
Appendix: Full listings of ucionch_cgv.f	24

EGS5 サンプルプログラム (ucionch_cgv.f)
電離箱による空気吸収線量の計算
(Japanese Parts)

1 Combinatorial Geometry (CG)

1.1 Body の定義

EGS 用 CG [1] では、以下のような立体 (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 \mathbf{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 \mathbf{V} と、中心からの高さベクトル \mathbf{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 \mathbf{V} 、底面中心からの上面中心への高さベクトル \mathbf{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 \mathbf{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 θ_1 と終わりの角度 θ_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 $\theta_2=2\pi$ とする。

Table 1: 各形状の立体とその記述のためのデータ

形状	通番	各形状の立体を定義するデータ					
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	
		θ_1	θ_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 記号が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。

1.3 リージョン定義の例

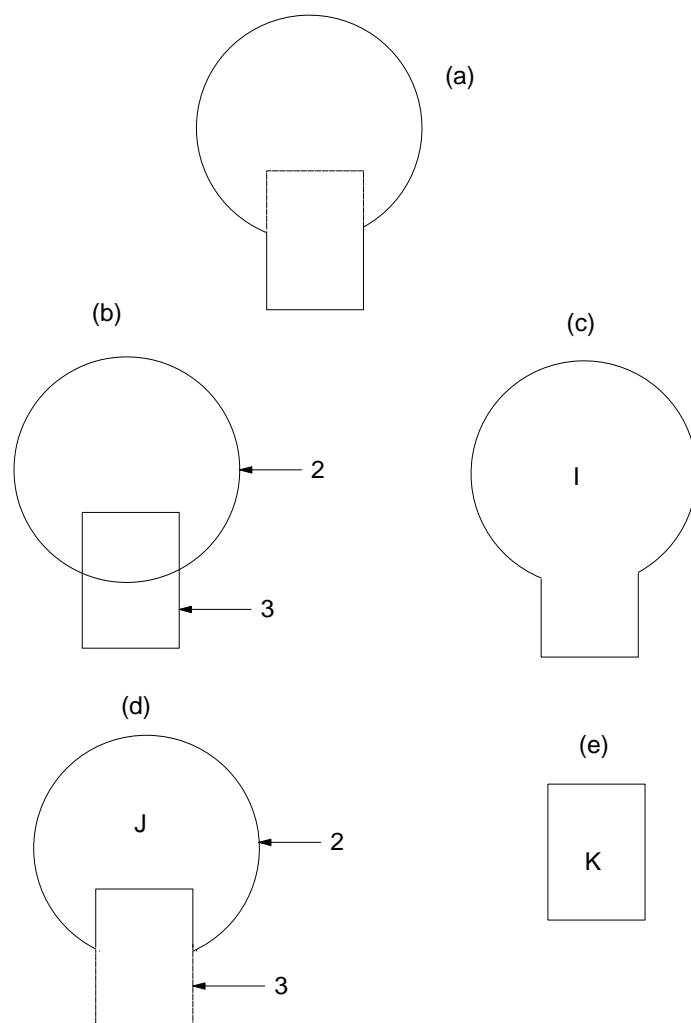


Figure 1: Combinatorial Geometry の例

第1図に示すような、球 (body 2) に円筒 (body 3) が挿入している様な体系を考える。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 に属さ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。

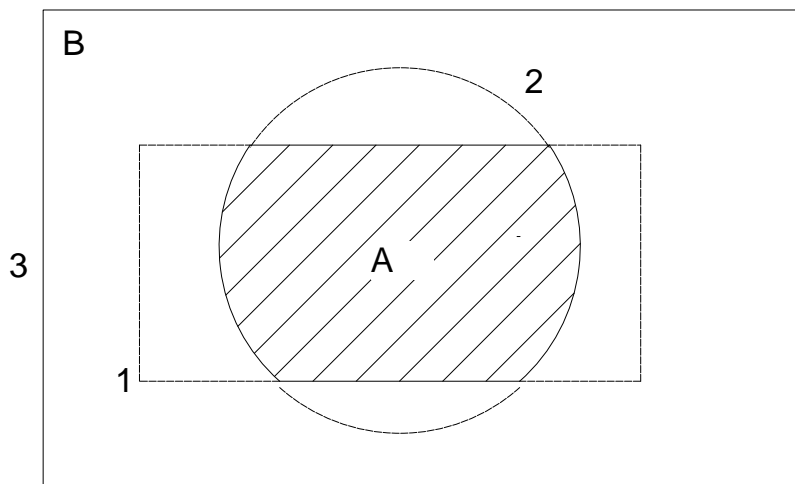


Figure 2: Use of OR operator.

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

$$B = +3 - 1OR + 3 - 2$$

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 ucionch_cgv.fの概要

ucionch_cgv.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のデータファイルに記述する。

2.1 CG 入力データ

図3に示すように円筒の組み合わせで体系を定義している。

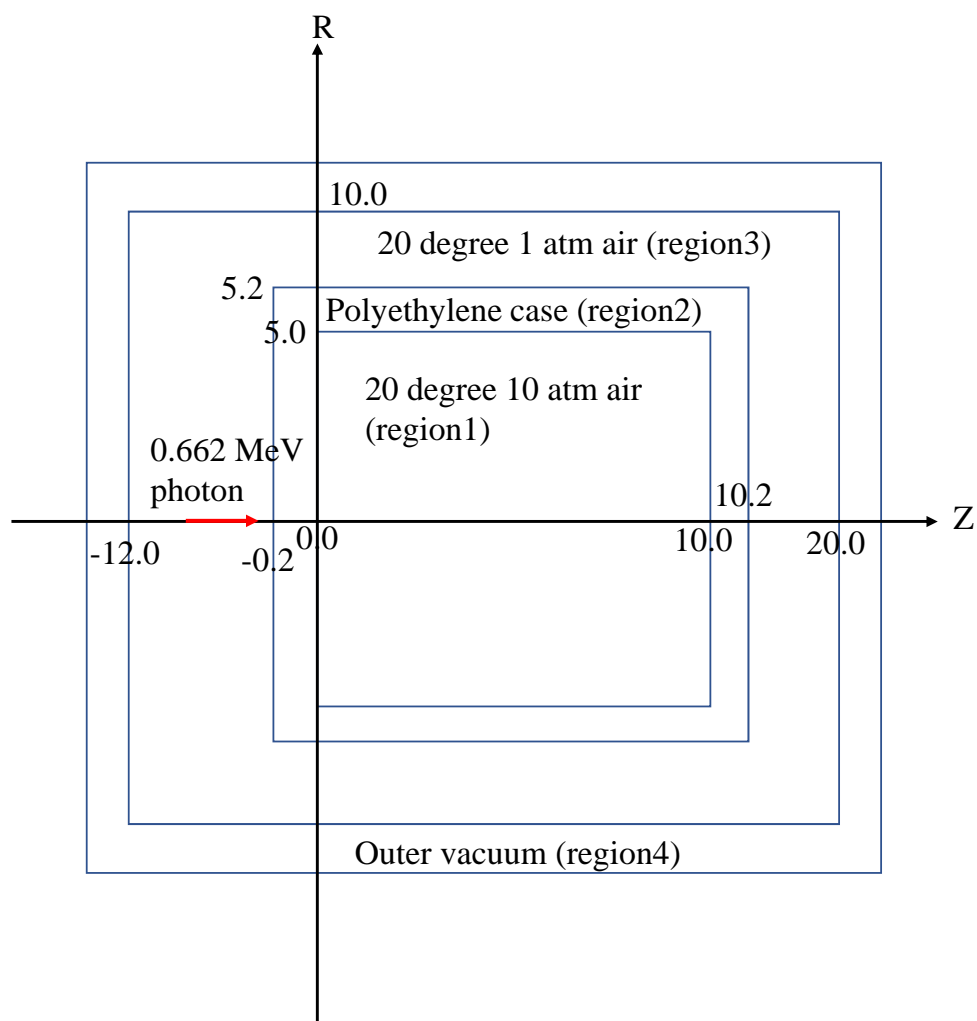


Figure 3: ucionch_cgv.fのジオメトリー

この形状の入力データは、以下のように記述する。

```

RCC  1      0.00      0.0      0.0      0.00      0.0
      10.0      5.00
RCC  2      0.00      0.0     -0.2      0.00      0.0
      10.4      5.20
RCC  3      0.00      0.0    -10.0      0.00      0.0
      30.0      10.0
RCC  4      0.00      0.0    -12.0      0.00      0.0
      34.0      12.0
END

```

```

Z1      +1
Z2      +2      -1
Z3      +3      -2
Z4      +4      -3
END
  1      2      1      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 0.662MeV の光子
- 入射粒子の角度は、Z-軸に沿って (0.0, 0.0, -5.0) から垂直入射

2. 得られる情報

- CGview 用の飛跡データ (egs5job.pic)
- 計算結果 (egs5job.out)
 - 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
 - 電離箱内の 1 気圧空気の吸収エネルギーから求めた空気吸収線量
 - * LARCH 機能を使って、電離箱内の空気にエネルギーを付与する電子の生成場所（電離箱内の空気中、前面のケース、後面のケース及び側面のケース）毎の吸収エネルギーと合計のエネルギーから求めた空気吸収線量
 - * 空気のエネルギー質量エネルギー吸収係数と光子の飛程長から求めた空気吸収線量
 - * 光子の飛程長から求めた電離箱内での光子の平均スペクトル

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 に直接関係する `include` 関係のファイルは、`include/ディレクトリ` (egs に関係するもの)、`pegscommons/` (pegs に関係するもの) および `auxcommons/` (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もの) とリンクすることにより、使用できるようにしている。¹

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_brempr.f'
include 'include/egs5_edge.f'
include 'include/egs5_media.f'

```

¹これらの設定は、egs5run スクリプト又は egs5run.bat で設定される。

```

include 'include/egs5_misc.f'
include 'include/egs5_stack.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'

```

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。²

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。

```

! -----
! 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'

! -----
! 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。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
* dose(5),dosep,deltae,spec(50),maxpict
real*8 dose,dosep,deltae,spec
integer maxpict

```

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、pegs5 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(6,FILE='egs5job.out',STATUS='unknown')
open(4,FILE='egs5job.inp',STATUS='old')
open(39,FILE='egs5job.pic',STATUS='unknown')

```

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

ユーザーコードで使用する物質数を nmed で定義する。

物質データ及び各物質の Characteristic Dimension を設定した後で、pegs5 を call する。medarr のデータは必ず 24 文字分を指定する必要がある。物質名が 24 文字未満の場合には空白を補って、

²EGS4 の COMIN マクロに対応する扱いである。

合計 24 文字とする。Characteristic Dimension 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 (1 cm × 1 cm × 1 cm の立方体であれば 1cm) に設定する。

```

nmed=2
if(nmed.gt.MXMED) then
  write(6,'(A,I4,A,I4,A/A)')
*   ' nmed (' ,nmed,' ) larger than MXMED (' ,MXMED,' )',
*   ' MXMED in iclude/egs5_h.f must be increased.'
  stop
end if

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

medarr(1)='20 degree 1 atm air      '
medarr(2)='POLYETHYLENE            '

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

chard(1) = 5.0d0      ! automatic step-size control
chard(2) = 0.2d0

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

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

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

```

3.3 メインプログラム: Step 3

飛跡データファイルのフォーマットを指定する `nprec` を設定する。このユーザーコードでは、フリーフォーマットの 3 を指定する。計算結果の出力ファイルに、CG データの開始を示す `CG data` を書き込み、その後 CG の入力データを読み込み、CG データを指定したファイル (この場合は、6) に出力する処理を行うサブルーティン `geomgt` を call する。その後、CG データの終了を意味する `End of CG data` を出力する。次に、飛跡データファイルに必要な情報を出力する。出力ユニットである `ifto` は、39 に設定している。PICT のデータモードを示す文字列 (`CSTA-FREE` 又は、`CSTA`) を出力し、再度 subroutine `geomgt` により CG データを飛跡データファイルに出力する。最後に CG データの終了を意味する `CEND` を出力する。これらの処理後、`cg` データから、リージョン総数である `nreg` を引き出す。

CG を使用する場合には、この部分が必ず必要であり、変更する必要はない。

```

write(6,*) 'Read cg-related data'

!-----
! initialize cg related parameter
!-----
nprec=3      ! PICT data mode for CGView in free format

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

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

```

```

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

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

```

```

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

```

物質番号の設定を、CG データの最後で定義した情報から読み込んだ後、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、光電子の角度分布、特性 X 線の発生）を行う。

anlux 乱数のシード inseed の値を設定し、初期化する。

```

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

!  Set option except vacuum region
  do i=1,nreg
    if(med(i).ne.0) then
      iphter(i) = 1      ! Switches for PE-angle sampling
      iedgfl(i) = 1      ! K & L-edge fluorescence
      iauger(i) = 0      ! K & L-Auger
      iraylr(i) = 0      ! Rayleigh scattering
      lpolar(i) = 0      ! Linearly-polarized photon scattering
      incohr(i) = 0      ! S/Z rejection
      iprofr(i) = 0      ! Doppler broadening
      impacr(i) = 0      ! Electron impact ionization
    end if
  end do

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

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

```

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、垂直に検出器の中心に入射するとしている。なお CG を用いる場合には、`irin=0` と指定することにより `irin` を自動的に設定することができる。このユーザーコードでは線源位置は固定なので、`irin=0` の場合は、直後に `irin` を CG により求めるルーチンが書かれている。

```

! Define initial variables for incident particle normally incident
! on the slab
  iqin=0          ! Incident particle charge - photons
  ekein=0.662     ! Incident particle kinetic energy
  xin=0.0         ! Source position
  yin=0.0
  zin=-5.0
  uin=0.0         ! Moving along z axis
  vin=0.0
  win=1.0
  irin=0          ! Starting region (0: Automatic search in CG)
  wtin=1.0        ! Weight = 1 since no variance reduction used

```

```

! pdf data for many source
deltae=0.05 ! Energy bin of response

!-----
! Get source region from cg input data
!-----
!
if(irin.le.0.or.irin.gt.nreg) then
  call srzone(xin,yin,zin,iqin+2,0,irin)
  if(irin.le.0.or.irin.ge.nreg) then
    write(6,fmt="( ' Stopped in MAIN. irin = ',i5)")irin
    stop
  end if
  call rstnxt(iqin+2,0,irin)
end if

```

3.5 メインプログラム: Step 5

pegsで作成した物質データの最高エネルギー（全エネルギー）の最小値 `emaxe` を `hatch` で求めるために、`emaxe` を 0 に設定後に、subroutine `hatch` を呼ぶ。`emaxe` は、入射粒子のエネルギーが、物質データの最高エネルギーを超えていないことをチェックするために用いる。`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。

```

      emaxe = 0.D0 ! dummy value to extract min(UE,UP+RM).
      write(6,'(/A)') ' 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')

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

```

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。`Ncases` は、ヒストリー数で、`maxpict` は、飛跡情報を記録するヒストリー数である。

```

! Energy bin width
deltae=ekein / 50

! Zero the variables
do i=1,5
  dose(i)=0.D0
  doses(i)=0.D0
  dose2s(i)=0.D0
end do
dosep=0.D0
doseps=0.D0
dosep2s=0.D0
do j=1,50
  spec(j)=0.D0
end do

! Set histories
ncases=10000000

```

```
! Set maximum number for pict
maxpict=50
```

3.8 メインプログラム: Step 8

飛跡表示ファイルにバッチ番号を出力した後で、ncases ヒストリーだけ subroutine shower を call する。入射粒子の運動エネルギーに対応する電子の全エネルギーが物質データで設定した最高エネルギー emaxe を超えていないか調べる。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 wtin を加え、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 wtin を加える。また、吸収エネルギーの値により、波高分布の対応するチャンネルに wtin を加える。統計誤差評価のために、粒子スペクトルを含め上記の変数のヒストリー毎の結果の自乗値の和を求めておく。

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

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

do i=1,ncases                                ! -----
                                           ! Start of shower call-loop
                                           ! -----
    latchi=0

! -----
! Select incident energy
! -----

    wtin = 1.0

    wtsum = wtsum + wtin                      ! Keep running sum of weights
    etot = ekein + iabs(iqin)*RM              ! Incident total energy (MeV)
    if(iqin.eq.1) then                        ! Available K.E. (MeV) in system
        availke = ekein + 2.0*RM              ! for positron
    else                                       ! Available K.E. (MeV) in system
        availke = ekein                       ! for photon and electron
    end if

    totke = totke + availke                   ! Keep running sum of KE

! -----
! Select incident angle
! -----

! -----
! Print first NWRITE or N_LINES, whichever comes first
! -----
    if (ncount .le. nwrite .and. ilines .le. nlines) then
        ilines = ilines + 1
        write(6,280) etot,xin,yin,zin,uin,vin,win,iqin,irin,idin
280      FORMAT(7G15.7,3I5)
    end if

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

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



```

1      ' normalized.',3e12.5)")uin,vin,win
      stop
      end if

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

!      -----
!      Sum variable and its square.
!      -----

      do ie=1,5
        doses(ie)=doses(ie)+dose(ie)
        dose2s(ie)=dose2s(ie)+dose(ie)*dose(ie)
        dose(ie)=0.d0
      end do
      doseps=doseps+dosep
      dosep2s=dosep2s+dosep*dosep
      dosep=0.d0
      do ie=1,50
        specs(ie)=specs(ie)+spec(ie)
        spec2s(ie)=spec2s(ie)+spec(ie)*spec(ie)
        spec(ie)=0.D0
      end do

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

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

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

!      =====
!      if(iwatch.gt.0) call swatch(-88,iwatch)
!      =====

      call plotxyz(99,0,0,0.D0,0.D0,0.D0,0.D0,0,0,0.D0,0.D0)
      write(39,fmt="( '9' )")          ! Set end of batch for CG View

      tt=etime(tarray)
      tt1=tarray(1)
      cputime=tt1-tt0
      write(6,300) cputime
300  format(' Elapsed Time (sec)=',G15.5)

```

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。ucnaicgv.f では、次のような MCNP 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。
- x_i を i 番目のヒストリーの結果とする。
- x の平均値を計算する:

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

- x_i の分散値を以下の式から求める。:

$$s^2 = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2 \simeq \overline{x^2} - \bar{x}^2 \quad (\overline{x^2} = \frac{1}{N} \sum_{i=1}^N x_i^2). \quad (2)$$

- \bar{x} の分散値は、

$$s_{\bar{x}}^2 = \frac{1}{N} s^2 \simeq \frac{1}{N} [\overline{x^2} - \bar{x}^2] \quad (3)$$

となる。

- 統計誤差として、

$$s_{\bar{x}} \simeq \left[\frac{1}{N} (\overline{x^2} - \bar{x}^2) \right]^{1/2} \quad (4)$$

を用いる。

先の計算した結果とその自乗の和は、上記の処理に用いる。

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。電離箱中での空気 9 の吸収エネルギーから求めた発生光子当たりの「空気の吸収線量」、空気の質量エネルギー吸収係数と電離箱中での空気の飛程長から求めた発生光子当たりの「空気の吸収線量」及び電離箱中での空気の飛程長から求めた電離箱中での光子の平均スペクトルについて、ヒストリー毎の結果の和と結果の自乗和から、平均値とその統計誤差を求めて出力する。

光子の飛程長を使った計算では、各粒子の検出器中での飛程長 (ustep) に光子のエネルギーと空気の質量エネルギー吸収係数を掛けたの合計及び、光子のエネルギー区分毎の飛程長の合計を電離箱の体積で割ることにより、電離箱中での吸収エネルギー及び平均の光子スペクトルを求めることができる。

```

tdet=10.0
rdet=5.0
tcase=0.2
rcase=0.2
write(6,330) tdet,rdet,tcase,rcase
330  FORMAT(/' Detector length=',G15.5,' cm'/
*      ' Detector radius=',G15.5,' cm'/
*      ' Polyethylene case thickness=',G10.2,' cm'/
*      ' Polyethylene case side thickness=',G10.2,' cm'/)

write(6,340) ekein
340  FORMAT(' Results for ',G15.5,'MeV photon'/)

write(6,350) (media(i,2),i=1,24)
350  FORMAT(' Cover material is ',24A1/)

write(6,360) (media(i,1),i=1,24)
360  FORMAT(' Inside chamber is ',24A1/)

write(6,365) zin
365  FORMAT(' Cs-137 exists at z=',F8.2,' cm, x=y=0.0'/)

! -----
! Calculate average and its deviation
! -----
! Absorbed energy (MeV) devided by weight of air inside chamber (dweight)
! MeV/g
! 1MeV=1.602E-13J, 1kg=1000g
! 1MeV/g=1.602E-13(J/MeV)*1000(g/kg)=1.602E-10 Gy
tdet=10.0
rdet=5.0
vol=pi*rdet*rdet*tdet      ! volume of air cm3
weight=vol*rhom(med(1))   ! weight of air inside chamber g
                           ! rhom(1) is a density of air (medium 1)

write(6,370) vol,weight
370  FORMAT(' Volume of chamber : ',G15.5,' cm3'/
*      ' weight of chamber : ',G15.5,' g')

! -----
! Absorbed dose

```

```

! -----
380 write(6,380)
   FORMAT(/' Result from absorbed energy in air inside chamber')

   write(6,'(A)') ' Air absorbed dose from '

   do ie=1,5
     avdose = doses(ie)/ncount
     sigdose=dose2s(ie)/ncount
     sigdose=dsqrt((sigdose-avdose*avdose)/ncount)
     avdose = avdose/weight*1.602E-10
     sigdose = sigdose/weight*1.602E-10
     if(ie.eq.1) then
       write(6,'(A,G15.5,A,G15.4,A)')
*       ' Electron produced in air =' ,
*       avdose,'+-',sigdose,' Gy/incident photon'
     elseif(ie.eq.2) then
       write(6,'(A,G15.5,A,G15.4,A)')
*       ' Electron from front wall =' ,
*       avdose,'+-',sigdose,' Gy/incident photon'
     elseif(ie.eq.3) then
       write(6,'(A,G15.5,A,G15.4,A)')
*       ' Electron from bottom wall =' ,
*       avdose,'+-',sigdose,' Gy/incident photon'
     elseif(ie.eq.4) then
       write(6,'(A,G15.5,A,G15.4,A)')
*       ' Electron from side wall =' ,
*       avdose,'+-',sigdose,' Gy/incident photon'
     else
       write(6,'(A,G15.5,A,G15.4,A)')
*       ' Total =' ,
*       avdose,'+-',sigdose,' Gy/incident photon'
     end if
   end do

   write(6,400)
400  FORMAT(/' Result from absorbed energy in air'/
*      'using track length of photon inside chamber and',
*      ' mass energy absorption coefficient of air')

!   Track length (cm) gives photon fluence deviding by volume
!   Conversion from air absorbed dose in MeV cm2/g to that in Gy
!   Unit of mass energy absorption coefficient mu_en is cm2/g
!   Ausgab scores energy (MeV) times mu_en in unit of MeV cm2/g.
!   Main routine converts this into Gy (J/kg).
!   1MeV=1.602E-13J, 1kg=1000g
!   1MeV/g=1.602E-13(J/MeV)*1000(g/kg)=1.602E-10 Gy

   avdose = doseps/ncount
   dosep2s=dosep2s/ncount
   sigdose=dsqrt((dosep2s-avdose*avdose)/ncount)
   avdose = avdose/vol*1.602E-10
   sigdose = sigdose/vol*1.602E-10
   write(6,410) avdose,sigdose
410  FORMAT(' Air absorbed dose =' ,G15.4,'+-',G15.4,
*      ' Gy/incident photon')

! -----
!   Average photon spectrum inside chamber.
! -----
   write(6,420)
420  FORMAT(/' Average photon spectrum inside chamber'/)

   do ie=1,50
     elow=deltae*(ie-1)
     eup=deltae*ie

     avspe = specs(ie)/ncount
     spec2s(ie)=spec2s(ie)/ncount

```

```

        sigspe=dsqrt((spec2s(ie)-avspe*avspe)/ncount)
        avspe=avspe/vol/deltae
        sigspe= sigspe/vol/deltae

        write(6,430) eup,avspe,sigspe
430    FORMAT(' Upper energy', G15.5, ' MeV--',G12.5,'+-',G12.5,
*        ' photons/cm2/MeV')
    end do

```

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。

iarg < 5 の場合には、それぞれのリージョンでの吸収エネルギーを計算する。

電子が周囲の壁から空気中に入ってきた（現在のリージョン (irl) が前のリージョンと異なり、前のリージョン番号が周囲のケース 2) 場合には、粒子の z(np) により、前面 (1)、後面 (2) あるいは側面 (3) かを調べ、当該番号を latch(np) とする。電離箱内の空気中で発生した電子の場合は、latch(np) は 0 のままである。

```

        if(iql.eq.-1.and.irl.ne.iroid.and.iroid.eq.2) then
            if(z(np).le.0.0) then
                latch(np)=1
            elseif(z(np).ge.10.0) then
                latch(np)=2
            else
                latch(np)=3
            end if
        end if

```

リージョン番号が 1 (電離箱内の空気) の時は、ステップ中での吸収エネルギーを、latch(np) を使って、電子の生成場所毎に加算する。dose(5) は、合計なので、latch(np) に関係なく常に吸収エネルギーを加算する。

```

        if (irl .eq. 1) then ! inside ion chamber
            dose(5) = dose(5) + edepwt
            if(latch(np).ge.0.and.latch(np).le.3) then
                dose(latch(np)+1) = dose(latch(np)+1) + edepwt
            end if

```

リージョン 1 内で光子が移動する場合には、当該光子のエネルギーに対応する空気の質量エネルギー吸収係数 (dcon) を求め、エネルギーと光子の飛程長 (ustep) を使って、空気の吸収エネルギーを求め、dosep に加算する。また、電離箱内の空気中での平均光子スペクトルを得るために、光子のエネルギー区分毎の飛程長を加算する。

```

        if (iql .eq. 0) then ! photon
            dcon=encoa(e(np))
            dosep=dosep+wt(np)*e(np)*dcon*ustep
            ie = e(np)/deltae + 1
            if(ie .gt. 50) ie = 50
            spec(ie) = spec(ie) + wt(np)*e(np)*dcon*ustep
        end if

```

ヒストリー数が maxpict 以下の時は、飛積情報出力のために plotxyz を call する。

```

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

        if(iql.eq.-1.and.irl.ne.iroid.and.iroid.eq.2) then

```

```

        if(z(np).le.0.0) then
            latch(np)=1
        elseif(z(np).ge.10.0) then
            latch(np)=2
        else
            latch(np)=3
        end if
    end if

! -----
! 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
    end if

! -----
! Print out particle transport information (if switch is turned on)
! -----
    if (iwatch .gt. 0) call swatch(iarg,iwatch)
! -----
    if(iarg .ge. 5) return

    if (irl .eq. 1) then ! inside ion chamber
        dose(5) = dose(5) + edepwt
        if(latch(np).ge.0.and.latch(np).le.3) then
            dose(latch(np)+1) = dose(latch(np)+1) + edepwt
        end if

! -----
! Score particle information if it proceeds inside chamber
! -----
        if (iql .eq. 0) then ! photon
            dcon=encoea(e(np))
            dosep=dosep+wt(np)*e(np)*dcon*ustep
            ie = e(np)/deltae + 1
            if(ie .gt. 50) ie = 50
            spec(ie) = spec(ie) + wt(np)*e(np)*dcon*ustep
        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(6,100) e(np),x(np),y(np),z(np),u(np),v(np),w(np),
*           iql,irl,iarg
100    FORMAT(7G15.7,3I5)
    end if

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

    return

end

```

3.11 Subroutine howfar

CG を利用するかぎりユーザーが **howfar** を変更する必要は一切ない。

以下、参考のため howfar の機能を述べる。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。

3.12 function encoea

光子のエネルギーに対応する空気の質量エネルギー吸収係数を log-log 内挿で計算する function である。S. M. Seltzer と J. H. Hubbell のデータ [3] を用いている。このデータは、NIST のホームページ

<http://www.physics.nist.gov/PhysRefData/Xcom/html/xcom1-t.html>
で公開されているデータと同じものである。

4 計算結果

- `ucionch_cgv.f` を `egs5run` で実行する。
 - Linux 又は Cygwin の場合
ユーザーコード名として、`ucionch_cgv` を、ユニット 4 及びユニット 25 のファイル名には、何も入力しないでリターンする。
”Does this user code read from the terminal?” に対して 1 を入力する。
 - DOS の場合
`egs5run ucionch_cgv`
 - `ucionch_cgv` 等が、`egs5run.bat` を実行しているディレクトリーと別なディレクトリーにある場合は、ディレクトリー名からを記載する。DOS の場合、ディレクトリーの識別子は、/ ではなく \ であるので、間違わないように注意する。
- 計算が終了したら、`egs5job.out` を調べる。電離箱中の空気の吸収エネルギーから求めた空気の吸収線量と空気の質量エネルギー吸収係数と電離箱中での光子の飛程長から求めた空気の吸収線量を下の表に示す。

計算結果

空気吸収線量 (Gy per source photon)	
空気吸収エネルギー使用	
電離箱内の空気中で生成した電子	0.290E-14 ± 0.397E-16
前面のケースで生成した電子	0.757E-14 ± 0.686E-16
後面のケースで生成した電子	0.899E-15 ± 0.271E-16
側面のケースで生成した電子	0.894E-14 ± 0.373E-16
合計	0.203E-13 ± 0.120E-15
光子飛程長使用	0.392E-13 ± 0.143E-17

- 空気の吸収エネルギーから求めた結果が小さいことがわかる。
なお、計算機、コンパイラの違いにより、計算の状況がかわり、数値は統計誤差程度の変動を示すことがありうる。

5 実習課題

5.1 実習課題1：線源光子のエネルギーを変えた計算

次のように変更して、2つの空気吸収線量の関係を調べよ。

1. 線源を、0.2MeV) に変える。
2. 線源を、1.0 MeV に変える。

5.2 実習課題2：0.662MeV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、等方線源に変更した計算

線源位置を次のように変更して1 Bq 当たりの2つの空気吸収線量率の関係を調べよ。

1. -2.0 cm
2. -5.0 cm
3. -10.0 cm。

6 実習課題の解答例

比較のために、ucionch_cgv.f の計算モードでの結果 (egs5job.out) を別な名称のファイル名 (例えば、chamber_662.out) で保存しておく。

6.1 実習課題 1

1. ファイルの変更

- cp ucionch_cgv.f uciobch_cgv1.f
これは、UNIX 又は Cygwin の場合である。DOS の場合は、
copy ucionch_cgv.f uciobch_cgv1.f
又は、Windows 上でファイルのコピーを行う。以下の操作でも同様。
- cp ucionch_cgv.data uciobch_cgv1.data
- cp uciobch_cgv.inp uciobch_cgv1.inp
- uciobch_cgv1.f を以下のように変更する。
 - uciobch_cgv1.f 中の

ekein=0.662	! Incident particle kinetic energy
を	
ekein=0.2	! Incident particle kinetic energy
 - 又は、

ekein=1.0	! Incident particle kinetic energy
-----------	------------------------------------

 に変更する。

2. uciobch_cgv1.f を egs5run で実行する。

- Linux 又は Cygwin の場合
ユーザーコード名として、uciobch_cgv1 を、ユニット 4 及びユニット 25 のファイル名には、何も入力しないでリターンする。
”Does this user code read from the terminal?” に対して 1 を入力する。
- DOS の場合
egs5run uciobch_cgv1
- uciobch_cgv1 等が、egs5run.bat を実行しているディレクトリーと別なディレクトリーにある場合は、ディレクトリー名から記載する。DOS の場合、ディレクトリーの識別子は、/ ではなく ¥ であるので、間違わないように注意する。

3. 計算が終了したら、egs5job.out を調べ、0.662 MeV の結果と比較する。

空気吸収エネルギーの比較

	空気吸収線量 (Gy per source photon)		
	0.662MeV	0.200 MeV	1.00 MeV
空気吸収エネルギー使用			
電離箱内の空気中で生成した電子	0.290E-14 ± 0.397E-16	0.775E-14 ± 0.801E-16	0.194E-14 ± 0.303E-16
前面のケースで生成した電子	0.757E-14 ± 0.686E-16	0.110E-14 ± 0.318E-16	0.107E-13 ± 0.727E-16
後面のケースで生成した電子	0.899E-15 ± 0.271E-16	0.125E-15 ± 0.834E-17	0.114E-14 ± 0.291E-16
側面のケースで生成した電子	0.894E-14 ± 0.373E-16	0.117E-14 ± 0.328E-16	0.128E-13 ± 0.872E-16
合計	0.203E-13 ± 0.120E-15	0.101E-13 ± 0.397E-16	0.266E-13 ± 0.127E-15
光子飛程長使用	0.392E-13 ± 0.541E-17	0.107E-13 ± 0.124E-17	0.565E-13 ± 0.181E-17

6.2 実習課題 2

1. ファイルの変更

- cp uciobch_cgv.f uciobch_cgv2.f
- cp uciobch_cgv.data uciobch_cgv2.data
- cp uciobch_cgv.inp uciobch_cgv2.inp
- uciobch_cgv2.f を以下の様に変更する。

- 崩壊当たりの γ 線放出率 erate を変数に追加する。

```
real*8                                     ! Local variables
* availke,doses,dose2s,doseps,dosep2s,avdose,sigdose,avspe,sigspe,
* rr0,wtin,wtsun, xi0,yi0,zi0,vol,weight
```

を

```
real*8                                     ! Local variables
* availke,doses,dose2s,doseps,dosep2s,avdose,sigdose,avspe,sigspe,
* rr0,wtin,wtsun, xi0,yi0,zi0,vol,weight,erate
```

に変更する。

- erate の値を指定する data 文の挿入

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

を

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

```
data erate/0.851/
```

に変更する。

- 線源位置の変更

-5.0 cm 以外の場合は、

```
zin=-5.0
```

を

```
zin=-2.0
```

又は、

```
zin=-10.0
```

に変更する。

- 4π での等方角度分布をサンプリングする文を追加する。

```
! -----
! Select incident angle
! -----
```

を

```
! -----
! Select incident angle
! -----
275 call randomset(rnnow)
    zi0=2.0*rnnow-1.0
    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 275
    win = zi0/rr0
    uin = xi0/rr0
    vin = yi0/rr0
```

に変更。

- 出力部の変更

```

do ie=1,5
  avdose = doses(ie)/ncount
  sigdose=dose2s(ie)/ncount
  sigdose=dsqrt((sigdose-avdose*avdose)/ncount)
  avdose = avdose/weight*1.602E-10
  sigdose = sigdose/weight*1.602E-10
  if(ie.eq.1) then
    write(6,'(A,G15.5,A,G15.4,A)')
  *   ' Electron produced in air  =',
  *   avdose,'+-',sigdose,' Gy/incident photon'
  elseif(ie.eq.2) then
    write(6,'(A,G15.5,A,G15.4,A)')
  *   ' Electron from front wall  =',
  *   avdose,'+-',sigdose,' Gy/incident photon'
  elseif(ie.eq.3) then
    write(6,'(A,G15.5,A,G15.4,A)')
  *   ' Electron from bottom wall =',
  *   avdose,'+-',sigdose,' Gy/incident photon'
  elseif(ie.eq.4) then
    write(6,'(A,G15.5,A,G15.4,A)')
  *   ' Electron from side wall  =',
  *   avdose,'+-',sigdose,' Gy/incident photon'
  else
    write(6,'(A,G15.5,A,G15.4,A)')
  *   ' Total                      =',
  *   avdose,'+-',sigdose,' Gy/incident photon'
  end if
end do

write(6,400)
400  FORMAT(/' Result from absorbed energy in air'/
*     'using track length of photon inside chamber and',
*     ' mass energy absorption coefficient of air')

!   Track length (cm) gives photon fluence deviding by volume
!   Conversion from air absorbed dose in MeV cm2/g to that in Gy
!   Unit of mass energy absorption coefficient mu_en is cm2/g
!   Ausgab scores energy (MeV) times mu_en in unit of MeV cm2/g.
!   Main routine converts this into Gy (J/kg).
!   1MeV=1.602E-13J,  1kg=1000g
!   1MeV/g=1.602E-13(J/MeV)*1000(g/kg)=1.602E-10 Gy

  avdose = doseps/ncount
  dosep2s=dosep2s/ncount
  sigdose=dsqrt((dosep2s-avdose*avdose)/ncount)
  avdose = avdose/vol*1.602E-10
  sigdose = sigdose/vol*1.602E-10
  write(6,410) avdose,sigdose
410  FORMAT(' Air absorbed dose =',G15.4,'+-',G15.4,
*         ' Gy/source photon')

を

do ie=1,5
  avdose = doses(ie)/ncount
  sigdose=dose2s(ie)/ncount
  sigdose=dsqrt((sigdose-avdose*avdose)/ncount)
  avdose = avdose*erate/weight*1.602E-10*3600.0
  sigdose = sigdose*erate/weight*1.602E-10*3600.0
  if(ie.eq.1) then
    write(6,'(A,G15.5,A,G15.4,A)')
  *   ' Electron produced in air  =',
  *   avdose,'+-',sigdose,' Gy/h per Bq'
  elseif(ie.eq.2) then
    write(6,'(A,G15.5,A,G15.4,A)')
  *   ' Electron from front wall  =',
  *   avdose,'+-',sigdose,' Gy/h per Bq'
  elseif(ie.eq.3) then
    write(6,'(A,G15.5,A,G15.4,A)')

```

```

*      ' Electron from bottom wall =',
*      avdose,'+-',sigdose,' Gy/h per Bq'
      elseif(ie.eq.4) then
        write(6,'(A,G15.5,A,G15.4,A)')
*      ' Electron from side wall =',
*      avdose,'+-',sigdose,' Gy/h per Bq'
      else
        write(6,'(A,G15.5,A,G15.4,A)')
*      ' Total =',
*      avdose,'+-',sigdose,' Gy/h per Bq'
      end if
    end do

    write(6,400)
400  FORMAT(/' Result from absorbed energy in air'/
*      'using track length of photon inside chamber and',
*      ' mass energy absorption coefficient of air')

!      Track length (cm) gives photon fluence deviding by volume
!      Conversion from air absorbed dose in MeV cm2/g to that in Gy
!      Unit of mass energy absorption coefficient mu_en is cm2/g
!      Ausgab scores energy (MeV) times mu_en in unit of MeV cm2/g.
!      Main routine converts this into Gy (J/kg).
!      1MeV=1.602E-13J, 1kg=1000g
!      1MeV/g=1.602E-13(J/MeV)*1000(g/kg)=1.602E-10 Gy

      avdose = doseps/ncount
      sigdose=dosep2s/ncount
      sigdose=dsqrt((sigdose-avdose*avdose)/ncount)
      avdose = avdose*erate/vol*1.602E-10*3600.0
      sigdose = sigdose*erate/vol*1.602E-10*3600.0
      write(6,410) avdose,sigdose
410  FORMAT(' Air absorbed dose rate =',G15.4,'+-',G15.4,
*      ' Gy/h per Bq')

!      -----
!      Average photon spectrum inside chamber.
!      -----
      write(6,420)
420  FORMAT(/' Average photon spectrum inside chamber'/)

      do ie=1,50
        elow=deltae*(ie-1)
        eup=deltae*ie

        avspe = specs(ie)/ncount
        spec2s(ie)=spec2s(ie)/ncount
        sigspe=dsqrt((spec2s(ie)-avspe*avspe)/ncount)
        avspe=avspe*erate/vol/deltae
        sigspe= sigspe*erate/vol/deltae
        write(6,430) eup,avspe,sigspe
430  FORMAT(' Upper energy', G15.5,' MeV--',G12.5,'+-',G12.5,
*      ' photons/cm2/MeV/s per Bq')
      end do

```

に変更する。

2. uciobch_cgv2.f を egs5run で実行する。

3. 計算が終了したら、egs5job.out を調べ、各宣言位置で較する。

空気吸収エネルギーの比較

	空気吸収線量率 (Gy/ per Bq)		
	-2 cm	- 5 cm	-10 cm
空気吸収エネルギー使用			
電離箱内の空気中で生成した電子	0.100E-11 ± 0.400E-13	0.554E-12 ± 0.309E-13	0.209E-12 ± 0.184E-13
前面のケースで生成した電子	0.612E-11 ± 0.108E-12	0.258E-11 ± 0.700E-13	0.923E-12 ± 0.402E-13
後面のケースで生成した電子	0.352E-12 ± 0.295E-13	0.167E-12 ± 0.191E-13	0.942E-13 ± 0.141E-13
側面のケースで生成した電子	0.101E-10 ± 0.151E-12	0.487E-11 ± 0.106E-12	0.228E-11 ± 0.740E-13
合計	0.175E-10 ± 0.202E-12	0.817E-11 ± 0.137E-12	0.351E-11 ± 0.911E-13
光子飛程長使用	0.178E-10 ± 0.111E-13	0.831E-11 ± 0.832E-14	0.352E-11 ± 0.586E-14

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).
- [2] T. Sugita, T. Torii, A. Takamura, “Incorporating Combinatorial Geometry to the EGS5 Code and Its Speed-Up”, Twelfth EGS User’s Meeting in Japan, KEK Proc. **2005-10**, 7-21, (KEK, Tsukuba, 9 - 11 Aug. 2005).
- [3] S. M. Seltzer and J. H. Hubbell, “Tables and Graphs of photon mass attenuation coefficients and energy-absorption coefficients for photon energies 1 keV to 20 MeV for elements Z=1 to 92 and some dosimetric materials”, 1995 Japanese Society of Radiological Technology.


```

include 'include/egs5_brempr.f'
include 'include/egs5_edge.f'
include 'include/egs5_media.f'
include 'include/egs5_misc.f'
include 'include/egs5_stack.f'
include 'include/egs5_thresh.f'
include 'include/egs5_uphiot.f'
include 'include/egs5_useful.f'
include 'include/egs5_usersc.f'
include 'include/egs5_userxt.f'
include 'include/randomm.f'

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

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

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

common/totals/ ! Variables to score
* dose(5),dosep,deltae,spec(50),maxpict
real*8 dose,dosep,deltae,spec
integer maxpict

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

real*8 ! Local variables
* availke,doseps,dosep2s,avdose,sigdose,avspe,sigspe,
* rr0,wtin,wsum, xi0,yi0,zi0,vol,weight

real*8
* specs(50),spec2s(50),doses(5),dose2s(5)

real ! Local variables
* elow,eup,rdet,rcase,tcase,tdet

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

integer
* i,icases,idin,ie,ifti,ifto,ii,iiz,imed,ireg,isam,
* izn,nlist,j,k,n,ner
character*24 medarr(MXMED)

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

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

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

! -----
! Step 2: pegs5-call
! -----
! -----

```

```

! Define media before calling PEGS5
! -----
nmed=2
if(nmed.gt.MXMED) then
  write(6,'(A,I4,A,I4,A/A)')
*   ' nmed (' ,nmed,') larger than MXMED (' ,MXMED,')',
*   ' MXMED in iclude/egs5_h.f must be increased.'
  stop
end if

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

medarr(1)='20 degree 1 atm air      '
medarr(2)='POLYETHYLENE            '

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

chard(1) = 5.0d0      ! automatic step-size control
chard(2) = 0.2d0

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

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

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

! -----
! Step 3: Pre-hatch-call-initialization
! -----
write(6,*) 'Read cg-related data'

! -----
! Initialize CG related parameters
! -----
npreci=3      ! PICT data mode for CGView in free format

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

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

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

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

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

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

! Set option except vacuum region
do i=1,nreg
  if(med(i).ne.0) then
    iphter(i) = 1      ! Switches for PE-angle sampling
    iedgfl(i) = 1      ! K & L-edge fluorescence
    iauger(i) = 0      ! K & L-Auger
    iraylr(i) = 0      ! Rayleigh scattering
    lpolar(i) = 0      ! Linearly-polarized photon scattering
  end if
end do

```

```

        incohr(i) = 0      ! S/Z rejection
        iprofr(i) = 0     ! Doppler broadening
        impacr(i) = 0     ! Electron impact ionization
    end if
end do

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

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

! -----
! Step 4: Determination-of-incident-particle-parameters
! -----
! Define initial variables for incident particle normally incident
! on the slab
    iqin=0          ! Incident particle charge - photons
    ekein=0.662    ! Incident particle kinetic energy
    xin=0.0        ! Source position
    yin=0.0
    zin=-5.0
    uin=0.0        ! Moving along z axis
    vin=0.0
    win=1.0
    irin=0         ! Starting region (0: Automatic search in CG)
    wtin=1.0      ! Weight = 1 since no variance reduction used

! pdf data for many source
deltae=0.05      ! Energy bin of response

! -----
! Get source region from cg input data
! -----
!
    if(irin.le.0.or.irin.gt.nreg) then
        call srzone(xin,yin,zin,iqin+2,0,irin)
        if(irin.le.0.or.irin.ge.nreg) then
            write(6,fmt="( ' Stopped in MAIN. irin = ',i5)")irin
            stop
        end if
        call rstnxt(iqin+2,0,irin)
    end if

! -----
! Step 5: hatch-call
! -----
    emaxe = 0.D0 ! dummy value to extract min(UE,UP+RM).

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

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

140  write(6,140)
    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)
! -----

```



```

150 write(6,150)
   FORMAT(/,' Quantities associated with each MEDIA:')
   do j=1,nmed
     write(6,160) (media(i,j),i=1,24)
160   FORMAT(/,1X,24A1)
     write(6,170) rhom(j),rlcm(j)
170   FORMAT(5X,' rho=',G15.7,' g/cu.cm      rlc=',G15.7,' cm')
     write(6,180) ae(j),ue(j)
180   FORMAT(5X,' ae=',G15.7,' MeV      ue=',G15.7,' MeV')
     write(6,190) ap(j),up(j)
190   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(6,200) i
200      FORMAT(' medium(',I3,')=vacuum')
     else
       write(6,210) i,(media(ii,med(i)),ii=1,24),ecut(i),pcut(i)
210      FORMAT(' medium(',I3,')=',24A1,
*         'ecut=',G12.5,' MeV, pcut=',G12.5,' 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(6,220) izn
220          FORMAT(' X-ray information for Z=',I3)
           write(6,230) (ekx(ii,izn),ii=1,10)
230          FORMAT(' K-X-ray energy in keV',/,
*            4G15.5,/,4G15.5,/,2G15.5)
           write(6,240) (elx1(ii,izn),ii=1,8)
240          FORMAT(' L-1 X-ray in keV',/,4G15.5,/,4G15.5)
           write(6,250) (elx2(ii,izn),ii=1,5)
250          FORMAT(' L-2 X-ray in keV',/,5G15.5)
           write(6,260) (elx3(ii,izn),ii=1,7)
260          FORMAT(' L-3 X-ray in keV',/,4G15.5,/,3G15.5)
         end do
       end if
     end if
   end do

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

! -----
! Step 6: Initialization-for-howfar
! -----
! -----
! Step 7: Initialization-for-ausgab
! -----

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

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

   write(6,270)
270  FORMAT(/,' Energy/Coordinates/Direction cosines/etc.',/,
*      6X,'e',14X,'x',14X,'y',14X,'z',
*      14X,'u',14X,'v',14X,'w',11X,'iq',3X,'ir',1X,'iarg',/)

! Energy bin width

```

```

deltae=ekein / 50

! Zero the variables
do i=1,5
  dose(i)=0.D0
  doses(i)=0.D0
  dose2s(i)=0.D0
end do
dosep=0.D0
doseps=0.D0
dosep2s=0.D0
do j=1,50
  spec(j)=0.D0
end do

! Set histories
ncases=10000000
! Set maximum number for pict
maxpict=50

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

-----
! Step 8: Shower-call
-----
! Write batch number
write(39,fmt="( '0 1' )")

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

do i=1,ncases
! -----
! Start of shower call-loop
! -----
  latchi=0

! -----
! Select incident energy
! -----

  wtin = 1.0

  wtsum = wtsum + wtin
  etot = ekein + iabs(iqin)*RM
  if(iqin.eq.1) then
    availke = ekein + 2.0*RM
  else
    availke = ekein
  end if

  totke = totke + availke

! -----
! Select incident angle
! -----

! -----
! Print first NWRITE or N_LINES, whichever comes first
! -----
  if (ncount .le. nwrite .and. ilines .le. nlines) then
    ilines = ilines + 1
    write(6,280) etot,xin,yin,zin,uin,vin,win,iqin,irin,idin
280   FORMAT(7G15.7,3I5)
  end if

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

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

```

```

1      ' normalized.',3e12.5)")uin,vin,win
      stop
      end if

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

! -----
! Sum variable and its square.
! -----

      do ie=1,5
        doses(ie)=doses(ie)+dose(ie)
        dose2s(ie)=dose2s(ie)+dose(ie)*dose(ie)
        dose(ie)=0.d0
      end do
      doseps=doseps+dosep
      dosep2s=dosep2s+dosep*dosep
      dosep=0.d0
      do ie=1,50
        specs(ie)=specs(ie)+spec(ie)
        spec2s(ie)=spec2s(ie)+spec(ie)*spec(ie)
        spec(ie)=0.D0
      end do

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

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

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

!
! =====
! if(iwatch.gt.0) call swatch(-88,iwatch)
! =====

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

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

      tt=etime(tarray)
      tt1=tarray(1)
      cputime=tt1-tt0
      write(6,300) cputime
300   format(' Elapsed Time (sec)=',G15.5)

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

      write(6,310) ncount,ncases,totke
310   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(6,320) totke,availke,ncount
320   FORMAT(/,' Stopped in MAIN with TotKE=',G15.5,/,
*         ' AvailKE=',G15.5, /, ' Ncount=',I10)
        stop
      end if

      tdet=10.0
      rdet=5.0
      tcase=0.2
      rcase=0.2
      write(6,330) tdet,rdet,tcase,rcase
330   FORMAT(/' Detector length=',G15.5,' cm'/
*         ' Detector radius=',G15.5,' cm'/
*         ' Polyethylene case thickness=',G10.2,' cm'/
*         ' Polyethylene case side thickness=',G10.2,' cm'/)

      write(6,340) ekein
340   FORMAT(' Results for ',G15.5,'MeV photon'/)

      write(6,350) (media(i,2),i=1,24)
350   FORMAT(' Cover material is ',24A1/)

```

```

360 write(6,360) (media(i,1),i=1,24)
   FORMAT(' Inside chamber is ',24A1/)

365 write(6,365) zin
   FORMAT(' Cs-137 exists at z=',F8.2,' cm, x=y=0.0'//)

! -----
! Calculate average and its deviation
! -----
! Absorbed energy (MeV) devided by weight of air inside chamber (dweight)
! MeV/g
! 1MeV=1.602E-13J, 1kg=1000g
! 1MeV/g=1.602E-13(J/MeV)*1000(g/kg)=1.602E-10 Gy
! tdet=10.0
! rdet=5.0
! vol=pi*rdet*rdet*tdet ! volume of air cm3
! weight=vol*rhom(med(1)) ! weight of air inside chamber g
! ! rhom(1) is a density of air (medium 1)

370 write(6,370) vol,weight
   FORMAT(' Volume of chamber : ',G15.5,' cm3'/
*         ' weight of chamber : ',G15.5,' g')

! -----
! Absorbed dose
! -----
380 write(6,380)
   FORMAT(/' Result from absorbed energy in air inside chamber')

   write(6,'(A)') ' Air absorbed dose from '

   do ie=1,5
     avdose = doses(ie)/ncount
     sigdose=dose2s(ie)/ncount
     sigdose=dsqrt((sigdose-avdose*avdose)/ncount)
     avdose = avdose/weight*1.602E-10
     sigdose = sigdose/weight*1.602E-10
     if(ie.eq.1) then
       write(6,'(A,G15.5,A,G15.4,A)')
*       ' Electron produced in air =',
*       avdose,'+',sigdose,' Gy/incident photon'
     elseif(ie.eq.2) then
       write(6,'(A,G15.5,A,G15.4,A)')
*       ' Electron from front wall =',
*       avdose,'+',sigdose,' Gy/incident photon'
     elseif(ie.eq.3) then
       write(6,'(A,G15.5,A,G15.4,A)')
*       ' Electron from bottom wall =',
*       avdose,'+',sigdose,' Gy/incident photon'
     elseif(ie.eq.4) then
       write(6,'(A,G15.5,A,G15.4,A)')
*       ' Electron from side wall =',
*       avdose,'+',sigdose,' Gy/incident photon'
     else
       write(6,'(A,G15.5,A,G15.4,A)')
*       ' Total =',
*       avdose,'+',sigdose,' Gy/incident photon'
     end if
   end do

400 write(6,400)
   FORMAT(/' Result from absorbed energy in air'/
*         'using track length of photon inside chamber and',
*         ' mass energy absorption coefficient of air')

! Track lenght (cm) gives photon fluence deviding by volume
! Conversion from air absorbed dose in MeV cm2/g to that in Gy
! Unit of mass energy absorption coefficient mu_en is cm2/g
! Ausgab scores energy (MeV) times mu_en in unit of MeV cm2/g.
! Main routine converts this into Gy (J/kg).
! 1MeV=1.602E-13J, 1kg=1000g
! 1MeV/g=1.602E-13(J/MeV)*1000(g/kg)=1.602E-10 Gy

   avdose = doseps/ncount
   dosep2s=dosep2s/ncount
   sigdose=dsqrt((dosep2s-avdose*avdose)/ncount)
   avdose = avdose/vol*1.602E-10
   sigdose = sigdose/vol*1.602E-10

```

```

write(6,410) avdose,sigdose
410  FORMAT(' Air absorbed dose =',G15.4,'+-',G15.4,
*      ' Gy/source photon')

! -----
! Average photon spectrum inside chamber.
! -----
write(6,420)
420  FORMAT(/' Average photon spectrum inside chamber'/)

do ie=1,50
  elow=deltae*(ie-1)
  eup=deltae*ie

  avspe = specs(ie)/ncount
  spec2s(ie)=spec2s(ie)/ncount
  sigspe=dsqrt((spec2s(ie)-avspe*avspe)/ncount)
  avspe=avspe/vol/deltae
  sigspe= sigspe/vol/deltae

  write(6,430) eup,avspe,sigspe
430  FORMAT(' Upper energy', G15.5,' MeV--',G12.5,'+-',G12.5,
*      ' photons/cm2/MeV per source photon')
end do

nlist=1

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

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

stop
end

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

!-----ausgab.f-----
! Version: 080708-1600
! 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'

common/totals/                     ! Variables to score

```

```

* dose(5),dosep,deltae,spec(50),maxpict
real*8 dose,dosep,deltae,spec
integer maxpict

integer                                ! Arguments
* iarg

real*8                                  ! Local variables
* edepwt,dcon,decon,encocea

integer
* ie,iql,irl

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

if(iql.eq.-1.and.irl.ne.irold.and.irold.eq.2) then
  if(z(np).le.0.0) then
    latch(np)=1
  elseif(z(np).ge.10.0) then
    latch(np)=2
  else
    latch(np)=3
  end if
end if

! -----
! 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
end if

! -----
! Print out particle transport information (if switch is turned on)
! -----
if (iwatch .gt. 0) call swatch(iarg,iwatch)
if(iarg .ge. 5) return

if (irl .eq. 1) then ! inside ion chamber
  dose(5) = dose(5) + edepwt
  if(latch(np).ge.0.and.latch(np).le.3) then
    dose(latch(np)+1) = dose(latch(np)+1) + edepwt
  end if
end if

! -----
! Score particle information if it proceeds inside chamber
! -----
if (iql .eq. 0) then ! photon
  dcon=encocea(e(np))
  dosep=dosep+wt(np)*e(np)*dcon*ustep
  ie = e(np)/deltae + 1
  if(ie .gt. 50) ie = 50
  spec(ie) = spec(ie) + wt(np)*e(np)*dcon*ustep
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(6,100) e(np),x(np),y(np),z(np),u(np),v(np),w(np),
* iql,irl,iarg
100  FORMAT(7G15.7,3I5)
end if

! -----
! Output particle information for plot
! -----
if (ncount.le.maxpict) then
  call plotxyz(iarg,np,iq(np),x(np),y(np),z(np),e(np),ir(np),

```

```

*      wt(np),time(np))
end if

return

end

!-----last line of ausgab.f-----
!-----encoaea.f-----
! Version: 030831-1300
! Reference: SLAC-265 (p.19-20, Appendix 2)
!-----
!23456789|123456789|123456789|123456789|123456789|123456789|123456789|12
!
!      double precision function encoaea(energy)
!      Function to evaluate the energy absorption coefficient of air.
!      (Tables and Graphs oh photon mass attenuation coefficients and
!      energy-absorption coefficients for photon energies 1 keV to
!      20 MeV for elements Z=1 to 92 and some dosimetric materials,
!      S. M. Seltzer and J. H. Hubbell 1995, Japanese Society of
!      Radiological Technology)
!-----
      double precision function encoaea(energy)

      real hnu(38)/0.001,0.0015,0.002,0.003,0.0032029,0.0032029,
*          0.004,0.005,0.006,0.008,0.01,0.015,0.02,0.03,0.04,
*          0.05,0.06,0.08,0.10,0.15,0.2,0.3,0.4,0.5,0.6,0.8,1.0,
*          1.25,1.5,2.0,3.0,4.0,5.0,6.0,8.0,10.0,15.0,20.0/

      real enmu(38)/3599., 1188., 526.2, 161.4, 133.0, 146.0,
*          76.36, 39.31, 22.70, 9.446, 4.742, 1.334, 0.5389,
*          0.1537,0.06833,0.04098,0.03041,0.02407,0.02325,0.02496,
*          0.02672,0.02872,0.02949,0.02966,0.02953,0.02882,0.02789,
*          0.02666,0.02547,0.02345,0.02057,0.01870,0.01740,0.01647,
*          0.01525,0.01450,0.01353,0.01311/;

      real*8 energy,enm1,hnu1,ene0,slope;

      integer i

      if (energy.gt.hnu(38)) then
          encoaea=enmu(38)
          return
      end if
      if (energy.lt.hnu(1)) then
          encoaea=enmu(1)
          return
      end if

      do i=1,38
          if(energy.ge.hnu(i).and.energy.lt.hnu(i+1)) then
              enm1=log(enmu(i+1))
              enm0=log(enmu(i))
              hnu1=log(hnu(i+1))
              hnu0=log(hnu(i))

              ene0=dlog(energy)
              slope=(enm1-enm0)/(hnu1-hnu0)
              encoaea=exp(enm0+slope*(ene0-hnu0))
              return
          end if
          if(energy.eq.hnu(i+1)) then
              encoaea=enmu(i+1)
              return
          end if
      end do

! If sort/interpolation cannot be made, indicate so by writing
! a comment and stopping here.
      write(6,100) energy
100  FORMAT(///,' *****STOPPED IN ENCOEA*****',/, ' E=',G15.5,///)
      return
      end

!-----last line of encoaea.f-----
!-----howfar.f-----
! Version: 070627-1600
! Reference: T. Torii and T. Sugita, "Development of PRESTA-CG
! Incorporating Combinatorial Geometry in EGS4/PRESTA", JNC TN1410 2002-201,
! Japan Nuclear Cycle Development Institute (2002).

```

```
! Improved version is provided by T. Sugita. 7/27/2004
!-----
!23456789|123456789|123456789|123456789|123456789|123456789|123456789|12
```

```
!-----
! Required (geometry) subroutine for use with the EGS5 Code System
!-----
! This is a CG-HOWFAR.
!-----
```

```

subroutine howfar
implicit none
c
include 'include/egs5_h.f'      ! Main EGS "header" file
include 'include/egs5_epcont.f' ! COMMONs required by EGS5 code
include 'include/egs5_stack.f'
include 'auxcommons/geom_common.f' ! geom-common file
c
c
integer i,j,jjj,ir_np,nozone,jty,kno
integer irnear,irnext,irlold,irlfg,itvlf,ihitcg
double precision xidd,yidd,zidd,x_np,y_np,z_np,u_np,v_np,w_np
double precision tval,tval0,tval00,tval10,tvalmn,delhow
double precision atvalttmp
integer iq_np
c
ir_np = ir(np)
iq_np = iq(np) + 2
c
if(ir_np.le.0) then
write(6,*) 'Stopped in howfar with ir(np) <=0'
stop
end if
c
if(ir_np.gt.izonin) then
write(6,*) 'Stopped in howfar with ir(np) > izonin'
stop
end if
c
if(ir_np.EQ.izonin) then
idisc=1
return
end if
c
tval=1.d+30
itvalm=0
c
body check
u_np=u(np)
v_np=v(np)
w_np=w(np)
x_np=x(np)
y_np=y(np)
z_np=z(np)
c
do i=1,nbody(ir_np)
nozone=ABS(nbzone(i,ir_np))
jty=itblty(nozone)
kno=itblno(nozone)
c
rpp check
if(jty.eq.ityknd(1)) then
if(kno.le.0.or.kno.gt.irppin) go to 190
call rppcg1(kno,x_np,y_np,z_np,u_np,v_np,w_np)
c
sph check
elseif(jty.eq.ityknd(2)) then
if(kno.le.0.or.kno.gt.isphin) go to 190
call sphcg1(kno,x_np,y_np,z_np,u_np,v_np,w_np)
c
rcc check
elseif(jty.eq.ityknd(3)) then
if(kno.le.0.or.kno.gt.irccin) go to 190
call rcccg1(kno,x_np,y_np,z_np,u_np,v_np,w_np)
c
trc check
elseif(jty.eq.ityknd(4)) then
if(kno.le.0.or.kno.gt.itrcin) go to 190
call trccg1(kno,x_np,y_np,z_np,u_np,v_np,w_np)
c
tor check
elseif(jty.eq.ityknd(5)) then

```



```

        if(kno.le.0.or.kno.gt.itorin) go to 190
        call torcgl(kno,x_np,y_np,z_np,u_np,v_np,w_np)
c      rec check
        elseif(jty.eq.ityknd(6)) then
            if(kno.le.0.or.kno.gt.irecin) go to 190
            call reccgl(kno,x_np,y_np,z_np,u_np,v_np,w_np)
c      ell check
        elseif(jty.eq.ityknd(7)) then
            if(kno.le.0.or.kno.gt.iellin) go to 190
            call ellcgl(kno,x_np,y_np,z_np,u_np,v_np,w_np)
c      wed check
        elseif(jty.eq.ityknd(8)) then
            if(kno.le.0.or.kno.gt.iwedin) go to 190
            call wedcgl(kno,x_np,y_np,z_np,u_np,v_np,w_np)
c      box check
        elseif(jty.eq.ityknd(9)) then
            if(kno.le.0.or.kno.gt.iboxin) go to 190
            call boxcgl(kno,x_np,y_np,z_np,u_np,v_np,w_np)
c      arb check
        elseif(jty.eq.ityknd(10)) then
            if(kno.le.0.or.kno.gt.iarbin) go to 190
            call arbcgl(kno,x_np,y_np,z_np,u_np,v_np,w_np)
c      hex check
        elseif(jty.eq.ityknd(11)) then
            if(kno.le.0.or.kno.gt.ihexin) go to 190
            call hexcgl(kno,x_np,y_np,z_np,u_np,v_np,w_np)
c      haf check
        elseif(jty.eq.ityknd(12)) then
            if(kno.le.0.or.kno.gt.ihafin) go to 190
            call hafcgl(kno,x_np,y_np,z_np,u_np,v_np,w_np)
c      tec check
        elseif(jty.eq.ityknd(13)) then
            if(kno.le.0.or.kno.gt.itecin) go to 190
            call teccgl(kno,x_np,y_np,z_np,u_np,v_np,w_np)
c      gel check
        elseif(jty.eq.ityknd(14)) then
            if(kno.le.0.or.kno.gt.igelin) go to 190
            call gelcgl(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
        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-----

```