

egs5 サンプルプログラム (ucphatomcgv.f)
人体中の線量分布計算 (cg Version)
(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
1.1. Body の定義	1
1.2. リージョンの定義	1
1.3. リージョン定義の例	2
2. サンプルプログラム ucphantomcgv.fの概要	4
2.1. CG 入力データ	4
3. ユーザーコードの内容	6
3.1. メインプログラム: Step 1	6
3.2. Step 2:pegs5-call	8
3.3. Step 3: Pre-hatch-call-initialization	8
3.4. メインプログラム: Step 4	10
3.4.1. X 線源の種類を増やす方法:	11
3.4.2. X 線源の種類を増やす方法:	13
3.5. メインプログラム: Step 5	14
3.6. メインプログラム: Step 6	15
3.7. メインプログラム: Step 7	15
3.8. メインプログラム: Step 8	16
3.8.1. 統計誤差:	18
3.9. メインプログラム: Step 9	19
3.10. Subroutine ausgab	20
3.11. subroutine howfar	21
4. ucphantom.f と ucphantomcgv.f の計算速度の比較	21
5. 実習課題	21
5.1. 実習課題 1 : 線源を Cs-137 に変更する	21
5.2. 実習課題 2 : 線源を Co-60 に変更する	21
5.3. 実習課題 3 : 肺のモデルに変更する	21
5.4. 実習課題 4 : 腫瘍を含む肺	22
5.5. 実習課題 5 : 金属の挿入	22
5.6. その他	22
6. 実習課題の解答例	23
6.1. 実習課題 1	23
6.2. 実習課題 2	24
6.3. 実習課題 3	24
6.4. 実習課題 4	26
6.5. 実習課題 5	28

1. Combinatorial Geometry (CG)

1.1. Body の定義

PRESTA-CG*では、以下のような Body を使用する事ができる。

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

Table 1 Data required to described each body type.

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

1.2. リージョンの定義

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

+ 記号の後に body 番号が書かれた場合には、body 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

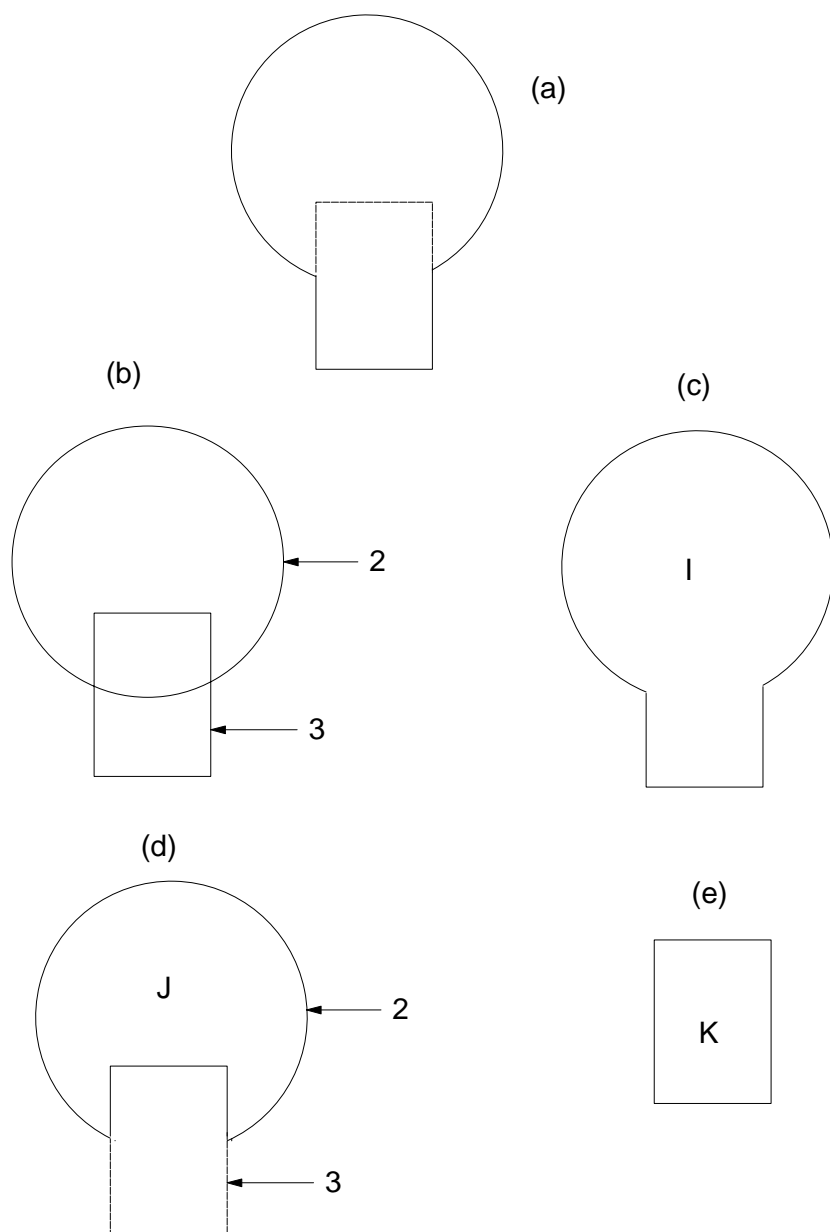


Figure 1: Examples of Combinatorial Geometry Method.

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、

$$I = +2OR + 3$$

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

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

リージョン J (図 1d) は、

$$J = +2 - 3$$

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

リージョン K (図 2e) は、単に

$$K = +3$$

と記述する。これは、body 3 の属する領域を意味する。

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

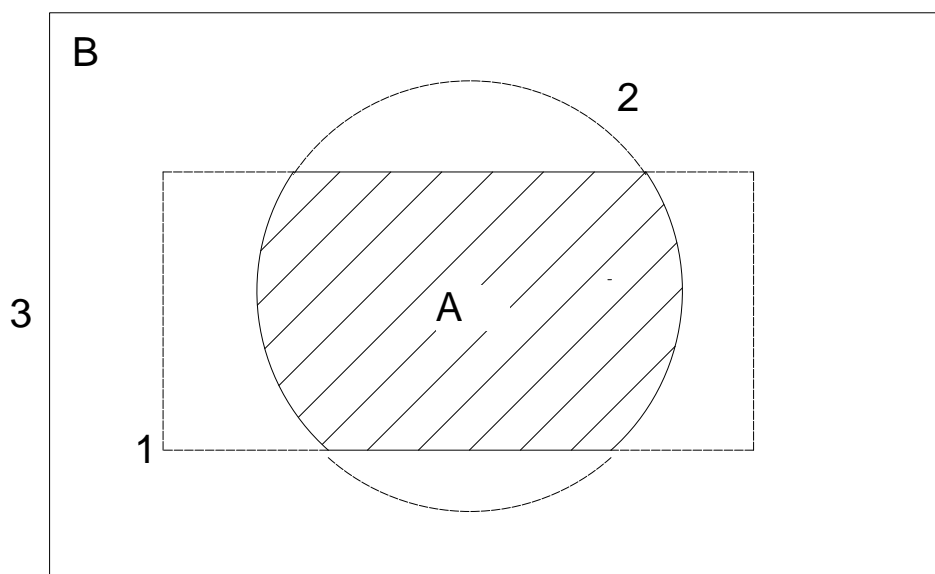


Figure 2: Use of OR operator.

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

$$A = +1 + 2$$

そして

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

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

2. サンプルプログラム ucphantomcgv.fの概要

ucphantomcgv.fは、CG を使ってファントム中の吸収線量を計算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載する。

2.1. CG 入力データ

ucphantomcgv.fでは、第3図に示すようにファントム前後の 5cm の空気層、厚さ 20cm ファントム、ファントム内の線量計算領域 (1cm x 1cm x 1cm) を直方体の組み合わせで形状を定義している。

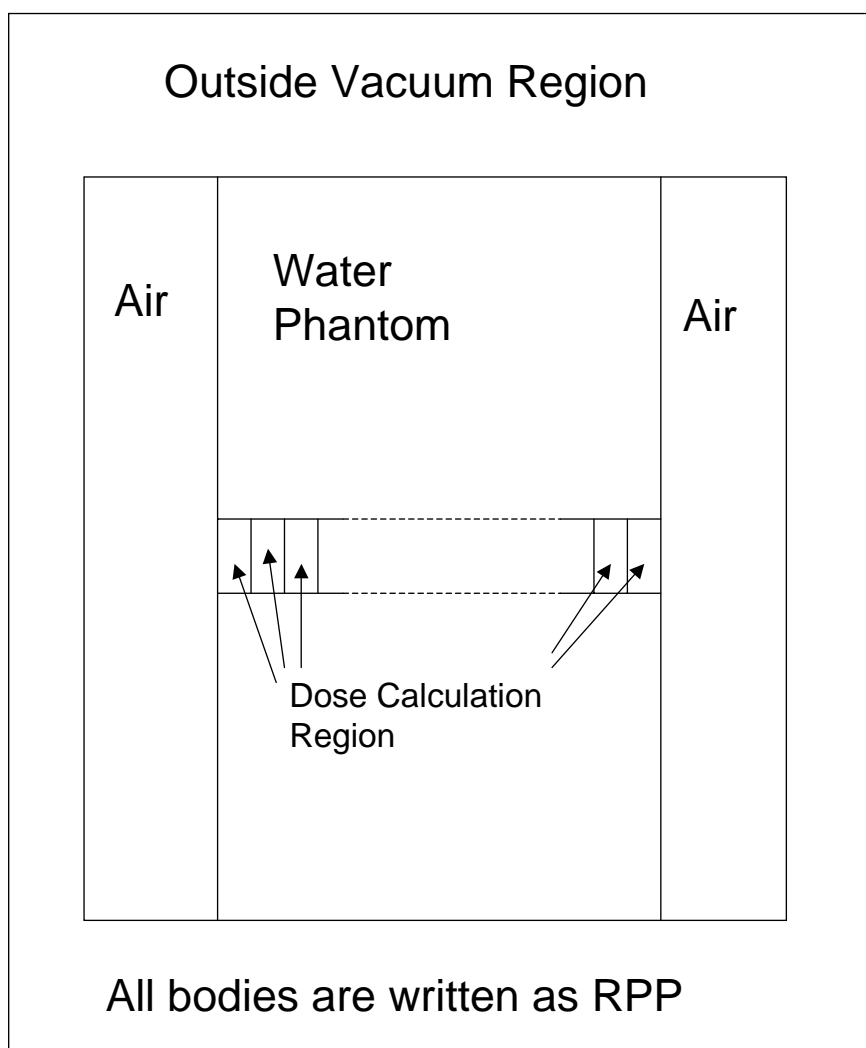


Figure 3: Geometry of ucphantomcgv.f.

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。

RPP	1	-15.0	15.0	-15.0	15.0	-5.0
		0.00				
RPP	2	-15.0	15.0	-15.0	15.0	0.0
		20.0				
RPP	3	-0.5	0.5	-0.5	0.5	0.0
		1.00				
RPP	4	-0.5	0.5	-0.5	0.5	1.0

RPP	5	2.00				
		-0.5	0.5	-0.5	0.5	2.0
RPP	6	3.00				
		-0.5	0.5	-0.5	0.5	3.0
RPP	7	4.00				
		-0.5	0.5	-0.5	0.5	4.0
RPP	8	5.00				
		-0.5	0.5	-0.5	0.5	5.0
RPP	9	6.00				
		-0.5	0.5	-0.5	0.5	6.0
RPP	10	7.00				
		-0.5	0.5	-0.5	0.5	7.0
RPP	11	8.00				
		-0.5	0.5	-0.5	0.5	8.0
RPP	12	9.00				
		-0.5	0.5	-0.5	0.5	9.0
RPP	13	10.00				
		-0.5	0.5	-0.5	0.5	10.0
RPP	14	11.00				
		-0.5	0.5	-0.5	0.5	11.0
RPP	15	12.00				
		-0.5	0.5	-0.5	0.5	12.0
RPP	16	13.00				
		-0.5	0.5	-0.5	0.5	13.0
RPP	17	14.00				
		-0.5	0.5	-0.5	0.5	14.0
RPP	18	15.00				
		-0.5	0.5	-0.5	0.5	15.0
RPP	19	16.00				
		-0.5	0.5	-0.5	0.5	16.0
RPP	20	17.00				
		-0.5	0.5	-0.5	0.5	17.0
RPP	21	18.00				
		-0.5	0.5	-0.5	0.5	18.0
RPP	22	19.00				
		-0.5	0.5	-0.5	0.5	19.0
RPP	23	20.00				
		-0.5	0.5	-0.5	0.5	0.0
RPP	24	20.00	15.0	-15.0	15.0	20.0
		-15.0				
RPP	25	25.00	20.0	-20.0	20.0	-20.0
		-20.0				
		40.00				
END						
Z1	+1					
Z2	+3					
Z3	+4					
Z4	+5					
Z5	+6					
Z6	+7					
Z7	+8					
Z8	+9					
Z9	+10					
Z10	+11					
Z11	+12					
Z12	+13					
Z13	+14					
Z14	+15					
Z15	+16					
Z16	+17					
Z17	+18					
Z18	+19					
Z19	+20					
Z20	+21					
Z21	+22					
Z22	+2	-23				
Z23	+24					
Z24	+25	-1	-2	-24		
END						

1. 体系

- 直方体 (RPP) の組み合わせ
- ファントム中で線量計算をする領域数 20
- 人体を一樣な水でモデル化 X-, Y-方向 30cm, 深さ 20cm
- ファントム前後に 5cm の空気層

2. 線源条件

- 粒子のエネルギーは、100kV の X 線 (スペクトルデータは、xray.dat から読み込み) データを用いてサンプリングする。
- 点等方線源:位置は、人体表面からの距離 (SPOSI) で指定
- ビームサイズ: 人体表面で XHBEAM*2 × YHBEAM*2 のビーム。

3. 計算モード

以下の 2 つのモードがある。目的によって切り替えて使用する。

- 飛跡表示システムを使って、飛跡を表示させるためのデータを作成するモード (imode=0) egs5job.pic に飛跡データを出力
- 多くのヒストリーを使用して線量分布を計算するモード (imode=1) egs5job.out に結果を出力

4. 得られる情報

(a) CGView を使った飛跡表示モード (imode=0)

- 飛跡情報 (egs5job.pic)
- コンソール上に、ファントム中心の 1cm × 1cm の領域の深度線量分布 (1cm 単位)
- コンソール上に、後方散乱係数 (ファントム中心の 1cm × 1cm の領域での、ファントムがない場合の照射線量に対するファントムがある場合の照射線量の比)。照射線量は、光子のエネルギー束と空気のエネルギー吸収係数から計算

(b) 線量分布計算モード

- 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
- 定義した平板に関するデータ
- サンプリングした X 線スペクトルと xray.dat から読み込んだスペクトルの比較
- ヒストリー数、ビームサイズ
- ファントム中心の 1cm × 1cm の領域の深度線量分布 (1cm 単位)
- 後方散乱係数
- 各リージョンの吸収エネルギー割合

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。配列の大きさを変更する場合には、egs5 に直接関係する場合は、egs/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。

最初の設定は、egs に直接関連する include 文である。

[†]これらの設定は、egs5run スクリプトで設定される。


```

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_elec.f'
include 'include/egs5_media.f'
include 'include/egs5_misc.f'
include 'include/egs5_switches.f'
include 'include/egs5_stack.f'
include 'include/egs5_thresh.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、ジオメトリ関係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/nfac.f'
include 'auxcommons/watch.f'

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

!
! -----
! cg related COMMONs
! -----
include 'auxcommons/cg/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/
* depe(20),faexp,fexps,imode,ndet,nreg      ! Variables to score
real*8 depe,faexp,fexps
integer imode,ndet,nreg

```

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

次に、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。pegs5 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。飛跡表示情報ファイルは、EGS4 では、ユニット 9 を使用していたが、egs5 では、39 に変更した。

```

!-----
! Units 7-26 are used in pegs and closed. It is better not
! to use as output file. If they are used must be re-open afeter

```

[‡]EGS4 の COMIN マクロに対応する扱いである。

```

!      getrz etc. Unit for pict must be 39.
!-----
      open(1,FILE='egs5job.out',STATUS='unknown')
      open(unit= 2,file='xray.dat',status='old') ! Data of source x-ray
      open(UNIT= 4,FILE='egs5job.inp',STATUS='old')
      open(39,FILE='egs5job.pic',STATUS='unknown')

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

```

unit 2 の open 文は、X 線データを xray.dat ファイルから読み込むために定義しているものである。その後、各種のカウンターを 0 にセットするサブルーティンを call する。

3.2. Step 2:pegs5-call

egs で使用している変数の初期化後、使用する物質の名前を定義する。ここで定義する物質の名前は、pegs5 の入力データで定義される物質名と一致している必要がある。pegs5 の入力データに含まれていれば、順番は一致している必要はない。定義した物質に関連するリージョンの最小大きさ（半径、厚さ、長さ等）に合い応する長さを chard として定義する。これらの設定の後、subroutine pegs5 を call する。

```

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

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

      nmed=2
      medarr(1)='WATER',
      medarr(2)='AIR-AT-NTP',

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

      chard(1) = 1.0d0      ! optional, but recommended to invoke
      chard(2) = 1.0d0      ! automatic step-size control

!-----
!      Run PEGS5 before calling HATCH
!-----
      write(6,*) ' PEGS5-call comes next'

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

```

3.3. Step 3: Pre-hatch-call-initialization

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ため、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。

```

!-----
!      Define pict data mode.
!-----
!      npreci 1: for PICT32

```

```

!           2: for CGview
nprec=2
!-----
! Initialize cg related parameter
!-----
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

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

```

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、全ての領域で光電子の角度オプション、特性 X 線の発生、レーレー散乱のオプション)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、通常はこの例の値が良い。

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

! Air region
med(1)=2
med(nreg-1)=2

! Water region
do i=1,nreg-1
    iphter(i) = 1      ! Switches for PE-angle sampling
    iedgfl(i) = 1      ! K & L-edge fluorescence
    iauger(i) = 0      ! K & L-Auger
    iraylr(i) = 1      ! 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
    med(i)=1           !Water phantom region
end do

!-----
! Set parameter estepe and estepe2
!-----
estepe=0.10
estepe2=0.20

```

```

write(6,130) estepe, estepe2
130  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(6,140) inseed
140  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

xray.dat からデータを読み込み、読み込んだ確率分布関数 (pdf) から、累積分布関数 (cdf) を計算する。読み込むデータは、ビン数 (nofebin)、ビンのエネルギー幅 (deltae:MeV 単位)、各ビンの X 線数 (sspec) である。

線源からファントム表面までの距離をキーボードから指定し、その後、線源パラメータを設定する。

```

!-----
! Read spectrum pdf
!-----
do i=1,1
  read(2,*) nofebin(i)
  read(2,*) deltae(i)
  read(2,*) (sspec(i,ie),ie=1,nofebin(i))
end do

!-----
! Select source type
!-----
150  write(6,160)
160  FORMAT(' Key in source type. 1:100kV')
  read(5,*) ixtype
  if (ixtype.eq.0.or.ixtype.gt.1) then
    write(6,170)
170  FORMAT(' IXTYPE must be >0 <= 1.')
    go to 150
  end if

!-----
! Calculate CDF for selected source
!-----
nsebin=nofebin(ixtype)
tnum=0.DO
do ie=1,nsebin
  tnum=tnum+sspec(ixtype,ie)
end do

nsebin=nsebin+1
ecdft(1)=0.0
do ie=2,nsebin
  ecdft(ie)=ecdft(ie-1)+sspec(ixtype,ie-1)/tnum
end do

!-----
! Make energy bin table
!-----
do ie=1,nsebin
  ebint(ie)=(ie-1)*deltae(ixtype)
end do

!-----

```

```

! Define source position from phantom surface.
!-----
write(6,180)
180 FORMAT(' Key in source position from phantom surface in cm')
read(5,*) sposi

!-----
! Source condition redefine
!-----
iqin=0 ! Incident charge - photons
ekein=ebint(nsebin) ! Maximum kinetic energy
etot=ekein + abs(iqin)*RM
xin=0.D0
yin=0.D0
zin=-sposi
uin=0.D0
vin=0.D0
win=1.D0
irin=0 ! Source region number is defined from xin and yin.

```

ファントム表面での照射野の半値幅は、X-方向とY-方向それぞれ別にキーボードから指定する。入力された値に伴い、Z-方向の方向余弦の最小値を計算する。

次に、モードをキーボードから指定する。飛跡表示モード (imode=0) か、計算モード (imode=1) の選択が出来る。

```

!-----
! Key in half width and height at phantom surface
!-----
write(6,190)
190 FORMAT(' Key in half width of beam at phantom surface in cm.')
read(5,*) xhbeam
write(6,200)
200 FORMAT(' Key in half height of beam at phantom surface in cm.')
read(5,*) yhbeam
radma2=xhbeam*xhbeam+yhbeam*yhbeam
wimin=sposi/dsqrt(sposi*sposi+radma2)

!-----
! Selection mode form Keyboard.
!-----
write(6,210)
210 FORMAT(' Key in mode. 0:trajectory display, 1:dose calculation')
read(5,*) imode
write(6,1090)

```

3.4.1. X線源の種類を増やす方法: ucphantomcgv.fでは、X線源スペクトルは、1個しかないが、複数の線源を用意したい場合には、以下のようにする。

1. X線源数の変更

```

real*8
* depeh(20),depeh2(20),dose(20),dose2(20),doseun(20),ebint(201),
* nofebin(1),deltae(1),sspec(1,201),ecdft(201),saspec(201)

```

で nofebin(1),deltae(1),sspec(1,201) 中の、1 を使用する X 線源の数に変更する。また、201 を、使用する X 線源中で、最も多いビン数の値に変更する。

2. xray.dat に、新たな線源に関するデータ (ビン数 (nofebin)、ビンのエネルギー幅 (deltae:MeV 単位)、各ビンの X 線数 (sspec)) を追加する。
3. データの読み込み及び X 線源を選択する部分を変更する。例えば、60kV, 80kV 及び 100kV から選択する場合 (スペクトルデータは、60kV, 80kV, 100kV の順に書かれているとする) には、

```

!-----
!   Read spectrum pdf
!-----
      do i=1,1
        read(2,*) nofebin(i)
        read(2,*) deltae(i)
        read(2,*) (sspec(i,ie),ie=1,nofebin(i))
      end do

!-----
!   Select source type
!-----
150  write(6,160)
160  FORMAT(' Key in source type. 1:100kV')
      read(5,*) ixtype
      if (ixtype.eq.0.or.ixtype.gt.1) then
        write(6,170)
170  FORMAT(' IXTYPE must be >0 <= 1.')
        go to 150
      end if
      end if

```

を、

```

!-----
!   Read spectrum pdf
!-----
      do i=1,3
        read(2,*) nofebin(i)
        read(2,*) deltae(i)
        read(2,*) (sspec(i,ie),ie=1,nofebin(i))
      end do

!-----
!   Select source type
!-----
150  write(6,160)
160  FORMAT(' Key in source type. 1:100kV, 2:80kV, 3:100kV')
      read(5,*) ixtype
      if (ixtype.eq.0.or.ixtype.gt.3) then
        write(6,170)
170  FORMAT(' IXTYPE must be >0 <= 3.')
        go to 150
      end if

```

に変更する。

4. 出力部で、線源に関する部分 (670-673 行目) を変更する。

```

      write(1,510) sposi
510  FORMAT(/' Absorbed energy inside phantom for 100 kV X-ray'/
*      ' Source position ',F10.1,' cm from phantom surface'/
*      ' Within 1cm x 1 cm area after 5 cm air')

```

を、

```

      if (ixtype.eq.1) then
        ixen=60
      elseif (ixtype.eq.2) then
        ixen=80
      else
        ixen=100
      end if
      write(1,510) ixen,sposi
510  FORMAT(/' Absorbed energy inside phantom for ',I4,'kV X-ray'/
*      ' Source position ',F10.1,' cm from phantom surface'/
*      ' Within 1cm x 1cm area after 5 cm air')

```

新たに使用する ixen を integer 宣言文に加える。

ヒストリー数 (ncases) は、コンソールから入力するようになっている。ncases の値として、0 を入力すると計算が終了する。0 以外の値を入力すると、新たなバッチとして処理される。

3.4.2. X 線源の種類を増やす方法: ucphantomcgv.f では、X 線源スペクトルは、1 個しかないが、複数の線源を用意したい場合には、以下のようにする。

1. X 線源数の変更

```
real*8
* depeh(LIMAX,LJMAX,LKMAX),depeh2(LIMAX,LJMAX,LKMAX),
* dose(LIMAX,LJMAX,LKMAX),doseun(LIMAX,LJMAX,LKMAX),
* ebint(201),nofebin(1),deltae(1),sspec(1,201),ecdft(201),
* saspec(201)
```

で nofebin(1),deltae(1),sspec(1,201) 中の、1 を使用する X 線源の数に変更する。また、201 を、使用する X 線源中で、最も多いピン数の値に変更する。

2. xray.dat に、新たな線源に関するデータ (ピン数 (nofebin)、ピンのエネルギー幅 (deltae:MeV 単位)、各ピンの X 線数 (sspec)) を追加する。
3. データの読み込み及び X 線源を選択する部分を変更する。例えば、60kV, 80kV 及び 100kV から選択する場合 (スペクトルデータは、60kV, 80kV, 100kV の順に書かれているとする) には、

```
!-----
!      Read spectrum pdf
!-----
      do i=1,1
        read(2,*) nofebin(i)
        read(2,*) deltae(i)
        read(2,*) (sspec(i,ie),ie=1,nofebin(i))
      end do

!-----
!      Select source type
!-----
180      write(6,190)
190      FORMAT(' Key in source type. 1:100kV')
        read(5,*) ixtype
        if (ixtype.eq.0.or.ixtype.gt.1) then
          write(6,200)
200      FORMAT(' IXTYPE must be >0 <= $NXTYPE.')
          go to 180
        end if
```

を、

```
!-----
!      Read spectrum pdf
!-----
      do i=1,3
        read(2,*) nofebin(i)
        read(2,*) deltae(i)
        read(2,*) (sspec(i,ie),ie=1,nofebin(i))
      end do

!-----
!      Select source type
!-----
180      write(6,190)
190      FORMAT(' Key in source type. 1:100kV, 2:80kV, 3:100kV')
        read(5,*) ixtype
        if (ixtype.eq.0.or.ixtype.gt.1) then
          write(6,200)
200      FORMAT(' IXTYPE must be >0 <= $NXTYPE.')
          go to 180
        end if
```

に変更する。

4. 出力部で、線源に関する部分 (573-576 行目) を変更する。

```
      write(1,390) sposi
390    FORMAT(/' Absorbed energy inside phantom for 100 kV X-ray'/ ' So
*urce position ',F10.1,' cm from phantom surface'/ ' Within 1cm x 1
*cm area after 5 cm air')
```

を、

```
      if (ixtype.eq.1) then
        ixen=60
      elseif (ixtype.eq.2) then
        ixen=80
      else
        ixen=100
      end if
      write(1,390) ixen,sposi
390    FORMAT(/' Absorbed energy inside phantom for ',I4,'kV X-ray'/
*      ' Source position ',F10.1,' cm from phantom surface'/
*      ' Within 1cm x 1cm area after 5 cm air')
```

新たに使用する ixen を integer 宣言文に加える。

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後、飛跡用のファイルに、各リージョンの物質番号を出力する。

```
! Define possible maximum total energy of electron before hatch
  emaxe = ekein + RM      ! photon

  write(6,220)
220  format(/' Start ucphantom'/
*      ' Call hatch to get cross-section data')

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

  write(6,230)
230  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(1,240)
240  FORMAT(/,' Quantities associated with each MEDIA:')
      do j=1,nmed
        write(1,250) (media(i,j),i=1,24)
250  FORMAT(/,1X,24A1)
        write(1,260) rhom(j),rlcm(j)
```



```

260     FORMAT(5X,' rho=',G15.7,' g/cu.cm      rlc=',G15.7,' cm')
       write(1,270) ae(j),ue(j)
270     FORMAT(5X,' ae=',G15.7,' MeV      ue=',G15.7,' MeV')
       write(1,280) ap(j),up(j)
280     FORMAT(5X,' ap=',G15.7,' MeV      up=',G15.7,' MeV',/)
       end do

       write(1,290)
290     FORMAT(/' Information of medium and cut-off for each region')
       do i=1,nreg
           if (med(i).eq.0) then
               write(1,300) i
300         FORMAT(' Medium(region:',I5,')= Vacuum')
           else
               write(1,310) i,(media(ii,med(i)),ii=1,24),
*                 ecut(i),pcut(i),rhor(i)
310         FORMAT(' Medium(region:',I5,
*                 ')=',24A1,/5X,'ECUT=',G10.5,' MeV, PCUT=',
*                 G10.5,' MeV, density=',F10.3)
           end if
       end do

       write(39,320)
320     FORMAT('MSTA')
       write(39,330) nreg
330     FORMAT(I4)
       write(39,340) (med(i),i=1,nreg)
340     FORMAT(15I4)
       write(39,350)
350     FORMAT('MEND')

```

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。

線量計算をする検出器数と計算したいヒストリー数をキーボードから入力する。飛跡データファイルに、データ番号を出力する。

```

       ncount = 0
       ilines = 0
       nwrite = 10
       nlines = 25
       idin = -1
       totke = 0.
       wtsum = 0.

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

!-----
!       Clear variables
!-----

       do nnn=1,20
           depe(nnn)=0.D0
           depeh(nnn)=0.D0
           depeh2(nnn)=0.D0
       end do

       faexp=0.D0
       faexps=0.D0

```

```

faexp2s=0.DO
fexp2s=0.DO
fexpss=0.DO
fexp2s=0.DO

do i=1,201
  saspec(i)=0.DO
end do

ibatch=0

!-----
!   Detector number to score
!-----
write(6,360) nreg-4
360  format(' Key in number of dose calculation region.(<=',I5,')')
     read(5,*) ndet

write(1,370)
370  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',/)

!-----
!   Key in history number
!-----
380  write(6,390)
390  FORMAT(' Key in number of cases (0 means end of calculation.)')
     read(5,*) ncases
     if (ncases.eq.0) go to 570

     ibatch=ibatch+1

     close(39,status='keep')
     open(39,file='egs5job.pic',access='append')
     write(39,400) ibatch
400  FORMAT('0',I5)

```

3.8. メインプログラム: Step 8

設定したヒストリー数 (ncases) だけ subroutine shower を call し、egs5 を使用する部分である。ucphantomcgv.f では、sposi の位置に、等方線源があり、そこから照射野内に、エネルギー分布を持った X 線が出るので、線源光子の方向、sposi が空気の厚さ (5cm) より長い場合の空気層の表面での位置及びエネルギーを決めるルーチンが加わっている。

各ヒストリー毎に、エネルギーバランス (入射運動エネルギーと、体系内外の吸収エネルギーの和が等しいこと) をチェックを行っている。

各ヒストリー終了後、平均値とその分散計算のために、計算対象量の値とその自乗をそれぞれ加算する。

```

do j=1,ncases
!-----
!   Start of CALL SHOWER loop
!-----
  icases=j

!-----
!   Determine direction (isotropic)
!-----
410  call randomset(w0)
     win=w0*(1.0-wimin)+wimin
     call randomset(phai0)
     phai=pi*(2.0*phai0-1.0)
     sinth=dsqrt(1.DO-win*win)
     uin=dcos(phai)*sinth
     vin=dsin(phai)*sinth
     dis=sposi/win
     xpf=dis*uin

```

```

ypf=dis*vin
if (dabs(xpf).gt.xhbeam.or.dabs(ypf).gt.yhbeam) go to 410
if (sposi.gt.5.0) then
  disair=(sposi-5.0)/win
  xin=disair*uin
  yin=disair*vin
  zin=-5.DO
else
  xin=0.DO
  yin=0.DO
  zin=-sposi
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

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

call randomset(ei0)
do ie=2,nsebin
  if (ei0.lt.ecdft(ie)) then
    go to 420
  end if
end do

420  if (ie.gt.nsebin) then
      ie=nsebin
    end if
    saspec(ie)=saspec(ie)+1.DO
    if (ecdft(ie).eq.ecdft(ie-1)) then
      ekin=ebint(ie-1)
    else
      * ekin=ebint(ie-1)+(ei0-ecdft(ie-1))*(ebint(ie)-ebint(ie-1))/
        (ecdft(ie)-ecdft(ie-1))
    end if
    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

    latchi=0

!
! -----
! Print first NWRITE or NLINES, whichever comes first
! -----
if (ncount .le. nwrite .and. ilines .le. nlines) then
  ilines = ilines + 1
  write(6,430) etot,xin,yin,zin,uin,vin,win,iqin,irinn,idin
430  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 + partd - ekin)/ekin .gt. 1.d-10) then

```

```

write(*,440) icesas, eparte, epartd
440   FORMAT('Error on # ',I6,' Escape = ',F9.5,' Deposit = ',F9.5)
endif

!-----
!   Sum variable and its squqre.
!-----

do kdet=1,ndet
  depeh(kdet)=depeh(kdet)+depe(kdet)
  depeh2(kdet)=depeh2(kdet)+depe(kdet)*depe(kdet)
  depe(kdet)=0.0
end do

faexps=faexps+faexp
faexp2s=faexp2s+faexp*faexp
faexp=0.0
fexpss=fexpss+fexps
fexpss2s=fexpss2s+fexps*fexps
fexps=0.0

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

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

end do                                     ! -----
                                           ! End of CALL SHOWER 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。ucphantomcgv.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、

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

を用いる。

先の計算すべき量とその自乗の和は、上記の処理を行うために行っている。

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。線量計算モードでは、最初に xray.dat からサンプリングした X 線のスペクトルと、元の X 線スペクトルとの比較、線源の条件 (線源のタイプ、位置)、ヒストリー数を出力する。その後、吸収線量を求めたい領域での平均吸収線量とその統計的な誤差を求めるを出力する。

```

!-----
!   Sampled source spectrum
!-----
      do ie=2,nsebin
        saspec(ie)=saspec(ie)/float(ncases)
      end do

      if (imode.ne.0) then
        write(1,480)
480      FORMAT(/' Comparison between sampled spectrum and original data'
* /23X,'   Sampled   Probability',25X,'   Sampled   Probability'
* )
        do ie=2,nsebin,2
          if(ie.eq.nsebin) then
            write(1,490) ebint(ie),saspec(ie),ecdft(ie)-ecdft(ie-1)
490          FORMAT(1X,G9.3,' MeV(upper)-- ',2G12.5)
          else
            write(1,500) ebint(ie),saspec(ie),ecdft(ie)-ecdft(ie-1),
* ebint(ie+1), saspec(ie+1),ecdft(ie+1)-ecdft(ie)
500          FORMAT(1X,G9.3,' MeV(upper)-- ',2G12.5,3X, '; ',G9.3,
* ' MeV(upper)-- ',2G12.5)
            end if
          end do

          write(1,510) sposi
510          FORMAT(/' Absorbed energy inside phantom for 100 kV X-ray'/
* ' Source position ',F10.1,' cm from phantom surface'/
* ' Within 1cm x 1 cm area after 5 cm air')

          write(1,520) ncases, xhbeam, yhbeam
520          FORMAT(1X,I8,' photons normally incident from front side'/ ' Hal
*f width of beam is ',G15.5,'cm for X and ',G15.5,'cm for Y')
          end if

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

      area=1.DO*1.DO
      do kdet=1,ndet
        vol=area*1.DO
        dose(kdet)=depeh(kdet)/ncases
        dose2(kdet)=depeh2(kdet)/ncases
        doseun(kdet)=dsqrt((dose2(kdet)-dose(kdet)*dose(kdet))/ncases)
        dose(kdet)=dose(kdet)*1.602E-10/vol
        doseun(kdet)=doseun(kdet)*1.602E-10/vol
        depths=kdet-1.0
        depthl=kdet
        write(6,530)depths,depthl,(media(ii,med(kdet+1)),ii=1,24),
* rhor(kdet+1),dose(kdet),doseun(kdet)
530      FORMAT(' At ',F4.1,'--',F4.1,'cm (' ,24A1,' rho:',F8.4,')=' ,
* G13.5,'+-',G13.5,'Gy/incident')
        write(1,530) depths,depthl,(media(ii,med(kdet+1)),ii=1,24),
* rhor(kdet+1),dose(kdet),doseun(kdet)
      end do

```

同様に入射粒子による照射線量、ファントム表面での照射線量及び後方散乱係数とそれぞれの誤差を求めて出力する。

その後、飛跡データファイルに、出力されないでメモリー上に残っているデータを出力し、その後データの終了を意味する '9' を書き込む。

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 nreg とそれ以外のリージョンでの吸収エネルギーを求める。線量計算を行う領域は、リージョン 2 から nreg-3 であるので、irl が該当するリージョンの場合にのみ、idet=irl-1; を検出器番号として、吸収線量を計算する。

更に、光子が、ファントム表面を横切った場合かどうかの判定を行い、横切ったと判断した場合には、面エネルギー束と空気のエネルギー吸収計数から、ファントム表面での空気吸収線量を計算する。光子が、Z-軸に対して逆に進んだことがない場合 (ファントムが無い場合のファントム表面位置) には、同様な方式で、ファントム無しの空気の吸収線量を計算する。この計算のため、 $w(np)$ が負になった場合には、latch(np) を 1 にセットし、ファントム無しの計算に加えないようにしている。

飛跡表示モードの場合は、粒子の情報を記録する subroutine plotxyz を呼ぶ。

```

!-----
! Print out particle transport information (if switch is turned on)
!-----
!
! if (iwatch .gt. 0) call swatch(iarg,iwatch)
!-----
!
! Keep track of how deep stack gets
!-----
100  if (np.gt.MXSTACK) then
      write(6,100) np,MXSTACK
      FORMAT(// ' In AUSGAB, np=',I3,' >= maximum stack',
*         ' allowed which is',I3/1X,79('?')//)
      stop
      end if
!
!-----
! 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
! added SJW for particle by particle energy balance
          if(irl.eq.nreg) then
              eparte = eparte + edepwt
          else
              epartd = epartd + edepwt
          endif
          end if
!
!-----
! Score data ate detector region (region 2-21)
!-----
      if (irl.ge.2.and.irl.le.nreg-3) then
          idet=irl-1
          if(idet.ge.1.and.idet.le.ndet) then
              depe(idet)=depe(idet)+edepwt/rhor(irl)
          end if
          end if
!
!-----
! Check cross phantom surface
!-----
      if (abs(irl-irold).eq.1.and.iq(np).eq.0) then
          if((w(np).gt.0.0.and.irl.eq.2).or.(w(np).le.0.0.and.irl.eq.1))

```

```

*   then
      if (dabs(w(np)).ge.0.0349) then
         cmod=dabs(w(np))
      else
         cmod=0.0175
      end if
      esing=e(np)
      dcon=encoea(esing)           ! PHOTX data
      fexps=fexps+e(np)*dcon*wt(np)/cmod
      if (w(np).lt.0.0) latch(np)=1
      if (w(np).gt.0.0.and.latch(np).eq.0) then
         faexp=faexp+e(np)*dcon*wt(np)/cmod
      end if
    end if
  end if

! -----
! 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),
*   w(np))
  end if

  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 ucphantomcgv.f の cg ルーチン用 howfar を使用する。

4. ucphantom.f と ucphantomcgv.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、ボクセル形状の howfar に比べ、計算時間が長いという問題がある。対象とする問題によって、違いは異なるが、ボクセル形状を使用している ucphantom.f と ucphantomcgv.f で全く同じ条件の計算を行うと、ucphantom.f の方が 1.4 倍速いという結果が得られている。[§]

5. 実習課題

5.1. 実習課題 1 : 線源を Cs-137 に変更する

線源を、Cs-137 の単一エネルギー光子 (0.662MeV) に変える。

5.2. 実習課題 2 : 線源を Co-60 に変更する

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。

5.3. 実習課題 3 : 肺のモデルに変更する

前面から 3cm を通常の人体組織、3-13cm を肺 (密度 0.3g/cm³) とし、その背後に 3cm の人体組織がある体系に変更する。線源は、元の X 線とする。

[§]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。

5.4. 実習課題4：腫瘍を含む肺

肺の前面から 3cm の位置に、厚さ 2cm の腫瘍を設定する。密度を通常の水とする。腫瘍は、X-, Y-方向全域に広がっていると仮定する。線源は、元の X 線とする。

5.5. 実習課題5：金属の挿入

ファントムから 5cm-6cm の領域を鉄に変える。線源は、元の X 線とする。

5.6. その他

上記に加えて、以下のような試みも考えられる。

- 線源として、他のエネルギーの X 線を使用する
- 光子だけでなく、電子入射の可能にする
- 挿入した金属の厚さを 1cm と異なる厚さにする
- 腫瘍の面積を限定する

6. 実習課題の解答例

比較のために、ucphantomcgv.fの計算モードで、ヒストリー数10000,計算領域20の場合の結果(egs5job.out)を別な名称のファイル名(例えば、phantom.out)で保存しておく。線源位置と、半値幅についてはこの比較で使用した条件と以下の実習課題で同じ値を使用する。

6.1. 実習課題1

1. cp ucphantomcgv.f ucphantomcgv1.f

2. ucphantomcgv1.fの変更

- 288-292行を以下のように変更する。

```
160  FORMAT(' Key in source type. 0:Cs,1:100kV')
      read(5,*) ixtype
      if (ixtype.lt.0.or.ixtype.gt.1) then
          write(6,170)
170  FORMAT(' IXTYPE must be >= 0 <= 1.')
```

- 390行を以下のように変更する。

```
      if(ixtype.eq.1) then
          ekein=ebint(nsebin)          ! Maximum kinetic energy
      else
          ekein=0.662
      end if
```

- 555-571行を以下のように変更する。

```
      if(ixtype.eq.1) then
          call randomset(ei0)
          do ie=2,nsebin
              if (ei0.lt.ecdft(ie)) then
                  go to 420
              end if
          end do
420  if (ie.gt.nsebin) then
          ie=nsebin
          end if
          saspec(ie)=saspec(ie)+1.D0
          if (ecdft(ie).eq.ecdft(ie-1)) then
              ekin=ebint(ie-1)
          else
              ekin=ebint(ie-1)+(ei0-ecdft(ie-1))*(ebint(ie)-ebint(ie-1))/
*              (ecdft(ie)-ecdft(ie-1))
          end if
          else
              ekin = ekein
          end if
```

- 659行をif(imode.ne.0.and.ixtype.eq.1) thenに修正する。

- 681行を以下のように変更する。

```
      write(1,510) sposi
510  FORMAT(/' Absorbed energy inside phantom for Cs gamma-ray'/
*      ' Source position ',F10.1,' cm from phantom surface'/
*      ' Within 1cm x 1 cm area after 5 cm air')
```

3. cp ucphantomcgv.data ucphantomcgv1.data

4. cp ucphantomcgv.inp ucphantomcgv1.inp

5. ucphantomcgv1.inpの7及び19行を以下のように変更する。

```
&INP AE=0.521,AP=0.010,UE=2.511,UP=2.0 /END
```

- ucphantomcgv1.f を egs5run で実行する。
ユニット 4 のファイル名及びユニット 25 のファイル名は、何も入力しないでリターンする。
- モードの選択で、1 の計算モードを選ぶ。
- 線量計算を行う領域数として 20 を入力する。
- 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、平均エネルギーが 0.662MeV となっていることを確認する。また、各値が X 線の場合と異なることを確認する。
- egs5job で再度実行し、モードの選択で 0 の飛跡表示を選び、CGView で飛跡の違いを見る。

6.2. 実習課題 2

- cp ucphantomcgv1.f ucphantomcgv2.f
- ucphantomcgv2.f を以下のように修正する。

- 288 行を以下のように変更する。

```
160  FORMAT(' Key in source type. 0:C0-60,1:100kV')
```

- 573 行を以下のように変更する。

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

- cp ucphantomcgv1.data ucphantomcgv2.data
- cp ucphantomcgv1.inp ucphantomcgv2.inp
- ucphantomcgv2.f を egs5run で実行する。
ユニット 4 のファイル名及びユニット 25 のファイル名は、何も入力しないでリターンする。
- モードの選択で、1 の計算モードを選ぶ。
- 線量計算を行う領域数として 20 を入力する。
- 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、平均エネルギーが 1.173 と 1.333 の平均値に近い値になっていることを確認する。また、各値が X 線の場合と異なることを確認する。

6.3. 実習課題 3

- cp ucphantomcgv.f ucphantomcgv3.f
- ucphantomcgv3.f の修正

- 246 行の後に以下を挿入する。

```
if((i.ge.5.and.i.le.14).or.i.eq.19) then ! Lung region
  rhor(i)=0.3
end if
```

- 476 行の

```
write(6,360) nreg-4
```

を

```
write(6,360) nreg-6
```

に変更する。

3. cp ucphantomcgv.data ucphantomcgv3.data

4. ucphantomcgv3.data を以下のように変更する。

RPP	1	-15.0	15.0	-15.0	15.0	-5.0
		0.00				
RPP	2	-15.0	15.0	-15.0	15.0	0.0
		16.0				
RPP	3	-0.5	0.5	-0.5	0.5	0.0
		1.00				
RPP	4	-0.5	0.5	-0.5	0.5	1.0
		2.00				
RPP	5	-0.5	0.5	-0.5	0.5	2.0
		3.00				
RPP	6	-0.5	0.5	-0.5	0.5	3.0
		4.00				
RPP	7	-0.5	0.5	-0.5	0.5	4.0
		5.00				
RPP	8	-0.5	0.5	-0.5	0.5	5.0
		6.00				
RPP	9	-0.5	0.5	-0.5	0.5	6.0
		7.00				
RPP	10	-0.5	0.5	-0.5	0.5	7.0
		8.00				
RPP	11	-0.5	0.5	-0.5	0.5	8.0
		9.00				
RPP	12	-0.5	0.5	-0.5	0.5	9.0
		10.00				
RPP	13	-0.5	0.5	-0.5	0.5	10.0
		11.00				
RPP	14	-0.5	0.5	-0.5	0.5	11.0
		12.00				
RPP	15	-0.5	0.5	-0.5	0.5	12.0
		13.00				
RPP	16	-0.5	0.5	-0.5	0.5	13.0
		14.00				
RPP	17	-0.5	0.5	-0.5	0.5	14.0
		15.00				
RPP	18	-0.5	0.5	-0.5	0.5	15.0
		16.00				
RPP	19	-15.0	15.0	-15.0	15.0	0.0
		3.00				
RPP	20	-15.0	15.0	-15.0	15.0	3.0
		13.00				
RPP	21	-15.0	15.0	-15.0	15.0	13.0
		16.00				
RPP	22	-15.0	15.0	-15.0	15.0	16.0
		21.00				
RPP	23	-0.5	0.5	-0.5	0.5	0.0
		16.00				
RPP	24	-20.0	20.0	-20.0	20.0	-20.0
		36.0				
END						
Z1	+1					
Z2	+3					
Z3	+4					
Z4	+5					
Z5	+6					
Z6	+7					
Z7	+8					
Z8	+9					
Z9	+10					
Z10	+11					
Z11	+12					
Z12	+13					
Z13	+14					

```

Z14      +15
Z15      +16
Z16      +17
Z17      +18
Z18      +19      -23
Z19      +20      -23
Z20      +21      -23
Z21      +22
Z22      +25          -1          -2          -22
END
  2      1      1      1      1      1      1      1      1      1      1      1
  1      1      1      1      1      1      1      1      2      0

```

5. cp ucphantomcgv.inp ucphantomcgv3.inp
6. ucphantomcgv3.f を egs5run で実行する。
 ユニット 4 のファイル名及びユニット 25 のファイル名は、何も入力しないでリターンする。
7. モードの選択で、1 の計算モードを選ぶ。
8. 線量計算を行う領域数として 16 を入力する。
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、肺の領域の密度が設定通りになっていることを確認する。また、線量分布が一様なファントムの場合と異なることを確認する。

6.4. 実習課題 4

1. cp ucphantomcgv3.f ucphantomcgv4.f
2. ucphantomcgv4.f の修正
 - 247 行を以下の様に変更する。
3. cp ucphantomcgv.data ucphantomcgv4.data
4. ucphantomcgv4.data を以下のように変更する。

```

          if((i.ge.5.and.i.le.7).or.(i.ge.10.and.i.le.14.).or.i.eq.19.
          * or.i.eq.21) then    ! Lung region

```

RPP	1	-15.0	15.0	-15.0	15.0	-5.0
		0.00				
RPP	2	-15.0	15.0	-15.0	15.0	0.0
		16.0				
RPP	3	-0.5	0.5	-0.5	0.5	0.0
		1.00				
RPP	4	-0.5	0.5	-0.5	0.5	1.0
		2.00				
RPP	5	-0.5	0.5	-0.5	0.5	2.0
		3.00				
RPP	6	-0.5	0.5	-0.5	0.5	3.0
		4.00				
RPP	7	-0.5	0.5	-0.5	0.5	4.0
		5.00				
RPP	8	-0.5	0.5	-0.5	0.5	5.0
		6.00				
RPP	9	-0.5	0.5	-0.5	0.5	6.0
		7.00				
RPP	10	-0.5	0.5	-0.5	0.5	7.0
		8.00				
RPP	11	-0.5	0.5	-0.5	0.5	8.0
		9.00				
RPP	12	-0.5	0.5	-0.5	0.5	9.0
		10.00				

```

RPP 13      -0.5      0.5      -0.5      0.5      10.0
            11.00
RPP 14      -0.5      0.5      -0.5      0.5      11.0
            12.00
RPP 15      -0.5      0.5      -0.5      0.5      12.0
            13.00
RPP 16      -0.5      0.5      -0.5      0.5      13.0
            14.00
RPP 17      -0.5      0.5      -0.5      0.5      14.0
            15.00
RPP 18      -0.5      0.5      -0.5      0.5      15.0
            16.00
RPP 19      -15.0     15.0     -15.0     15.0      0.0
            3.00
RPP 20      -15.0     15.0     -15.0     15.0      3.0
            6.00
RPP 21      -15.0     15.0     -15.0     15.0      6.0
            8.00
RPP 22      -15.0     15.0     -15.0     15.0      8.0
            13.00
RPP 23      -15.0     15.0     -15.0     15.0     13.0
            16.00
RPP 24      -15.0     15.0     -15.0     15.0     16.0
            21.00
RPP 25      -0.5      0.5      -0.5      0.5      0.0
            16.00
RPP 26      -20.0     20.0     -20.0     20.0     -20.0
            36.0
END
Z1          +1
Z2          +3
Z3          +4
Z4          +5
Z5          +6
Z6          +7
Z7          +8
Z8          +9
Z9          +10
Z10         +11
Z11         +12
Z12         +13
Z13         +14
Z14         +15
Z15         +16
Z16         +17
Z17         +18
Z18         +19      -25
Z19         +20      -25
Z20         +21      -25
Z21         +22      -25
Z22         +23      -25
Z23         +24
Z24         +26      -1      -2      -24
END
  2      1      1      1      1      1      1      1      1      1      1      1
  1      1      1      1      1      1      1      1      1      1      2      0

```

5. cp ucphantomcgv.inp ucphantomcgv4.inp
6. ucphantomcgv4.f を egs5run で実行する。
ユニット 4 のファイル名及びユニット 25 のファイル名は、何も入力しないでリターンする。
7. モードの選択で、1 の計算モードを選ぶ。
8. 線量計算を行う領域数として 16 を入力する。
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、腫瘍のヶ所の密度が設定通りになっていることを確認する。また、線量分布が一様なファントムの場合と異なることを確認する。

6.5. 実習課題5

1. cp ucphantomcgv.f ucphantomcgv5.f

2. ucphantomcgv5.f を以下のように修正する。

- 132 行を character*24 medarr(3) に変更する。
- 163-165 行を以下のように修正する。

```
nmed=3
medarr(1)='WATER'
medarr(2)='AIR-AT-NTP'
medarr(3)='FE'
```

- 247 行を以下のように変更する。

```
if(i.eq.7.or.i.eq.23) then ! Iron region
  med(i) = 3
else
  med(i)=1 !Water phantom region
end if
```

- 478 行を write(6,360) nreg-6 に変更する。

3. cp ucphantomcgv.data ucphantomcgv5.f

4. ucphantomcgv5.data を以下に変更する。

RPP	1	-15.0	15.0	-15.0	15.0	-5.0
		0.00				
RPP	2	-15.0	15.0	-15.0	15.0	0.0
		20.0				
RPP	3	-0.5	0.5	-0.5	0.5	0.0
		1.00				
RPP	4	-0.5	0.5	-0.5	0.5	1.0
		2.00				
RPP	5	-0.5	0.5	-0.5	0.5	2.0
		3.00				
RPP	6	-0.5	0.5	-0.5	0.5	3.0
		4.00				
RPP	7	-0.5	0.5	-0.5	0.5	4.0
		5.00				
RPP	8	-0.5	0.5	-0.5	0.5	5.0
		6.00				
RPP	9	-0.5	0.5	-0.5	0.5	6.0
		7.00				
RPP	10	-0.5	0.5	-0.5	0.5	7.0
		8.00				
RPP	11	-0.5	0.5	-0.5	0.5	8.0
		9.00				
RPP	12	-0.5	0.5	-0.5	0.5	9.0
		10.00				
RPP	13	-0.5	0.5	-0.5	0.5	10.0
		11.00				
RPP	14	-0.5	0.5	-0.5	0.5	11.0
		12.00				
RPP	15	-0.5	0.5	-0.5	0.5	12.0
		13.00				
RPP	16	-0.5	0.5	-0.5	0.5	13.0
		14.00				
RPP	17	-0.5	0.5	-0.5	0.5	14.0
		15.00				
RPP	18	-0.5	0.5	-0.5	0.5	15.0
		16.00				
RPP	19	-0.5	0.5	-0.5	0.5	16.0
		17.00				
RPP	20	-0.5	0.5	-0.5	0.5	17.0
		18.00				
RPP	21	-0.5	0.5	-0.5	0.5	18.0
		19.00				

```

RPP  22      -0.5      0.5      -0.5      0.5      19.0
      20.00
RPP  23      -0.5      0.5      -0.5      0.5      0.0
      20.00
RPP  24     -15.0     15.0     -15.0     15.0      0.0
      5.00
RPP  25     -15.0     15.0     -15.0     15.0      5.0
      6.00
RPP  26     -15.0     15.0     -15.0     15.0      6.0
      20.00
RPP  27     -15.0     15.0     -15.0     15.0     20.0
      25.00
RPP  28     -20.0     20.0     -20.0     20.0    -20.0
      40.00
END
Z1      +1
Z2      +3
Z3      +4
Z4      +5
Z5      +6
Z6      +7
Z7      +8
Z8      +9
Z9      +10
Z10     +11
Z11     +12
Z12     +13
Z13     +14
Z14     +15
Z15     +16
Z16     +17
Z17     +18
Z18     +19
Z19     +20
Z20     +21
Z21     +22
Z22     +24     -23
Z23     +25     -23
Z24     +26     -23
Z25     +27
Z26     +28     -1     -2     -27
END
  2      1      1      1      1      1      3      1      1      1      1      1
  1      1      1      1      1      1      1      1      1      1      3      1
  2      0

```

5. ucphantomcgv5.inp に次のデータを追加する。

```

ELEM
  &INP EFRACH=0.05,EFRACL=0.20,IRAYL=1,IBOUND=0,INCOH=0,
      ICPROF=0,IMPACT=0 /END
FE
      FE
ENER
  &INP AE=0.521,AP=0.010,UE=0.711,UP=0.3 /END
PWLF
  &INP /END
DECK
  &INP /END

```

6. ucphantomcgv5.f を egs5run で実行する。
 ユニット 4 のファイル名及びユニット 25 のファイル名は、何も入力しないでリターンする。
7. モードの選択で、1 の計算モードを選ぶ。
8. 線量計算を行う領域数として 16 を入力する。
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、鉄のリージョンが設定通りになっていることを確認する。また、線量分布が一様なファントムの場合と異なることを確認する。
11. egs5jobで再度実行し、飛跡表示モードを選択し、鉄の場所でほとんどの光子が止まっていることを確認する。

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).

Appendix 1 Full listings of ucphantomcgv.f

```

*****
***** KEK, High Energy Accelerator Research *
***** Organization *
** u c p h a n t o m c g v ** *
***** EGS5.0 USER CODE - 18 Jul 2005/1300 *
*****
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 *
*****
! The ucphantomcgv.f User Code requires a cg-input file only *
! (e.g., ucphantomcgv.data). *
! The following shows the geometry for ucphantomcgv.data. *
! Cg-data can be checked by CGview. *
! This user code corresponds to ucphantomcgp.mor for egs4. *
! Use Ranlux random number generator. *
*****
-----
cg Geometry (ucphantomcgv)
-----
      X
      |
      +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ 20.0
      |
      +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ 15.0
      |
      | Outer vacuum region
      |
      |
      +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
      |
      +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ 0.5
      |
      | Air | H2O | H2O | | | | H2O | Air |
      |-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
X-ray  =====>+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ Z
photons -20  -5.0  0  1.0  2.0  19.0  20.0  25.0  40.0
*****
|23456789|123456789|123456789|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_edge.f'
include 'include/egs5_elec.in.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'

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(20),faexp,fexps,imode,ndet
real*8 depe,faexp,fexps
integer imode,ndet

!**** real*8 ! Arguments
real*8 etot,totke
integer ins

!**** real*8 ! Local variables
real*8
* area,availke,depthl,depths,dis,disair,ei0,ekin,elow,eup,
* phai0,phai,radma2,sinth,sposi,tnum,vol,w0,wimin,wtin,wtsum,
* xhbeam,xpf,yhbeam,ypf

real*8 bsfa,bsferr,faexps,faexp2s,faexrr,fexpss,fexps2s,fexerr,
* faexpa,fexpsa

real*8
* depeh(20),depeh2(20),dose(20),dose2(20),doseun(20),ebint(201),
* nofebin(1),deltae(1),sspec(1,201),ecdft(201),saspec(201)

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

integer
* i,il,ibatch,icases,idin,ie,ifti,ifto,imed,ireg,isam,
* ixtype,j,k,kdet,nlist,nnn,nsebin

character*24 medarr(2)

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

open(unit= 1,file='egs5job.out',status='unknown')
open(unit= 2,file='xray.dat',status='old') ! Data of source x-ray
open(UNIT= 4,FILE='egs5job.inp',STATUS='old')
open(unit=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=2
medarr(1)='WATER'           '
medarr(2)='AIR-AT-NTP'      '

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

chard(1) = 1.0d0           ! optional, but recommended to invoke
chard(2) = 1.0d0           ! automatic step-size control

!   -----
!   Run PEGS5 before calling HATCH
!   -----
write(6,*) ' PEGS5-call comes next'

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

!-----
! Step 3: Pre-hatch-call-initialization
!-----
!-----
! Define pict data mode.
!-----
npreci 1: for PICT32
        2: for CGview
npreci=2
!-----
! Initialize cg related parameter
!-----
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

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

! Air region
med(1)=2
med(nreg-1)=2

! Water region

do i=2,nreg-2
  iphter(i) = 1      ! Switches for PE-angle sampling
  iedgfl(i) = 1      ! K & L-edge fluorescence

```

```

    iauger(i) = 0      ! K & L-Auger
    iraylr(i) = 1     ! 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
    med(i)=1         !Water phantom region
end do

! -----
! Set parameter estepe and estepe2
! -----
estepe=0.10
estepe2=0.20
write(6,130) estepe, estepe2
130  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(6,140) inseed
140  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
!-----
!-----
! Read spectrum pdf
!-----
do i=1,1
  read(2,*) nofebin(i)
  read(2,*) deltae(i)
  read(2,*) (sspec(i,ie),ie=1,nofebin(i))
end do

!-----
! Select source type
!-----
150  write(6,160)
160  FORMAT(' Key in source type. 1:100kV')
  read(5,*) ixtype
  if (ixtype.eq.0.or.ixtype.gt.1) then
    write(6,170)
170  FORMAT(' IXTYPE must be >0 <= 1.')
    go to 150
  end if

!-----
! Calculate CDF for selected source
!-----
nsebin=nofebin(ixtype)
tnum=0.D0
do ie=1,nsebin
  tnum=tnum+sspec(ixtype,ie)
end do

nsebin=nsebin+1
ecdft(1)=0.0
do ie=2,nsebin
  ecdft(ie)=ecdft(ie-1)+sspec(ixtype,ie-1)/tnum
end do

!-----
! Make energy bin table
!-----
do ie=1,nsebin
  ebint(ie)=(ie-1)*deltae(ixtype)
end do

!-----
! Define source position from phantom surface.

```

```

!-----
write(6,180)
180  FORMAT(' Key in source position from phantom surface in cm')
read(5,*) sposi

!-----
! Source condition redefine
!-----
iqin=0          ! Incident charge - photons
ekein=ebint(nsebin) ! Maximum kinetic energy
etot=ekein + abs(iqin)*RM
xin=0.DO
yin=0.DO
zin=-sposi
uin=0.DO
vin=0.DO
win=1.DO
irin=0          ! Source region number is defined from xin and yin.

!-----
! Key in half width and height at phantom surface
!-----
write(6,190)
190  FORMAT(' Key in half width of beam at phantom surface in cm.')
read(5,*) xhbeam
write(6,200)
200  FORMAT(' Key in half height of beam at phantom surface in cm.')
read(5,*) yhbeam
radma2=xhbeam*xhbeam+yhbeam*yhbeam
wimin=sposi/dsqrt(sposi*sposi+radma2)

!-----
! Selection mode form Keyboard.
!-----
write(6,210)
210  FORMAT(' Key in mode. 0:trajectory display, 1:dose calculation')
read(5,*) imode

!-----
! Step 5: hatch-call
!-----
! Define possible maximum total energy of electron before hatch
emaxe = ekein + RM          ! photon

write(6,220)
220  format(/' Start ucphantomcgv'/
*      ' Call hatch to get cross-section data')

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

write(6,230)
230  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(1,240)
240  FORMAT(/,' Quantities associated with each MEDIA:')
do j=1,nmed
write(1,250) (media(i,j),i=1,24)
250  FORMAT(/,1X,24A1)
write(1,260) rhom(j),rlcm(j)
260  FORMAT(5X,' rho=',G15.7,' g/cu.cm      rlc=',G15.7,' cm')
write(1,270) ae(j),ue(j)
270  FORMAT(5X,' ae=',G15.7,' MeV      ue=',G15.7,' MeV')

```

```

      write(1,280) ap(j),up(j)
280   FORMAT(5X,' ap=',G15.7,' MeV   up=',G15.7,' MeV',/)
      end do

      write(1,290)
290   FORMAT(/' Information of medium and cut-off for each region')
      do i=1,nreg
         if (med(i).eq.0) then
            write(1,300) i
300   FORMAT(' Medium(region:',I5,')= Vacuum')
         else
            write(1,310) i,(media(ii,med(i)),ii=1,24),
*             ecut(i),pcut(i),rhor(i)
310   FORMAT(' Medium(region:',I5,
*             ')=',24A1,/5X,' ECUT=',G10.5,' MeV, PCUT=',
*             G10.5,' MeV, density=',F10.3)
            end if
         end do

      write(39,320)
320   FORMAT('MSTA')
      write(39,330) nreg
330   FORMAT(I4)
      write(39,340) (med(i),i=1,nreg)
340   FORMAT(15I4)
      write(39,350)
350   FORMAT('MEND')

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

      ncount = 0
      ilines = 0
      nwrite = 10
      nlines = 25
      idin = -1
      totke = 0.
      wtsum = 0.

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

!-----
! Clear variables
!-----
      do nnn=1,20
         depe(nnn)=0.D0
         depeh(nnn)=0.D0
         depeh2(nnn)=0.D0
      end do

      faexp=0.D0
      faexp2s=0.D0
      faexp2s=0.D0
      fexp2s=0.D0
      fexp2s=0.D0
      fexp2s=0.D0

      do i=1,201
         saspec(i)=0.D0
      end do

      ibatch=0

!-----
! Detector number to score
!-----
      write(6,360) nreg-4
360   format(' Key in number of dose calculation region.(<=',I5,')')
      read(5,*) ndet

      write(1,370)
370   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',/)

!-----
!      Key in history number
!-----
380  write(6,390)
390  FORMAT(' Key in number of cases (0 means end of calculation.))'
      read(5,*) ncases
      if (ncases.eq.0) go to 570

      ibatch=ibatch+1

      close(39,status='keep')
      open(39,file='egs5job.pic',access='append')
      write(39,400) ibatch
400  FORMAT('0',I5)

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

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

      do j=1,ncases                                ! -----
                                                    ! Start of CALL SHOWER loop
      icases=j                                     ! -----

!-----
!      Determine direction (isotropic)
!-----
410  call randomset(w0)
      win=w0*(1.0-wimin)+wimin
      call randomset(phai0)
      phai=pi*(2.0*phai0-1.0)
      sinth=dsqrt(1.D0-win*win)
      uin=dcos(phai)*sinth
      vin=dsin(phai)*sinth
      dis=sposi/win
      xpf=dis*uin
      ypf=dis*vin
      if (dabs(xpf).gt.xhbeam.or.dabs(ypf).gt.yhbeam) go to 410
      if (sposi.gt.5.0) then
          disair=(sposi-5.0)/win
          xin=disair*uin
          yin=disair*vin
          zin=-5.D0
      else
          xin=0.D0
          yin=0.D0
          zin=-sposi
      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

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

      call randomset(ei0)
      do ie=2,nsebin
          if (ei0.lt.ecdft(ie)) then
              go to 420
          end if
      end do

```



```

420   if (ie.gt.nsebin) then
       ie=nsebin
     end if
     saspec(ie)=saspec(ie)+1.D0
     if (ecdft(ie).eq.ecdft(ie-1)) then
         ekin=ebint(ie-1)
     else
         ekin=ebint(ie-1)+(ei0-ecdft(ie-1))*(ebint(ie)-ebint(ie-1))/
*       (ecdft(ie)-ecdft(ie-1))
     end if
     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

     latchi=0

!-----
! Print first NWRITE or NLINES, whichever comes first
!-----
     if (ncount .le. nwrite .and. ilines .le. nlines) then
         ilines = ilines + 1
         write(6,430) etot,xin,yin,zin,uin,vin,win,iqin,irinn,idin
430     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(*,440) icases, eparte, epartd
440   FORMAT('Error on # ',I6,' Escape = ',F9.5,' Deposit = ',F9.5)
endif

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

     do kdet=1,ndet
         depeh(kdet)=depeh(kdet)+depe(kdet)
         depeh2(kdet)=depeh2(kdet)+depe(kdet)*depe(kdet)
         depe(kdet)=0.0
     end do

     faexps=faexps+faexp
     faexp2s=faexp2s+faexp*faexp
     faexp=0.0
     fexpss=fexpss+fexps
     fexp2s=fexp2s+fexps*fexps
     fexps=0.0

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

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

     tt=etime(tarray)
     tt1=tarray(1)
     cputime=tt1-tt0
     if (imode.ne.0) then
         write(1,450) cputime
450   format(' Elapsed Time (sec)=',G15.5)
     end if

!-----
! Step 9: Output-of-results
!-----
! Write out the results
!-----
     write(1,460) ncount,ncases,totke,totke/ncount
460   FORMAT('/', ' Ncount=',I10,' (actual cases run)',/,
*         ', Ncases=',I10,' (number of cases requested)',/,

```

```

*      ' TotKE =',G15.5,' (total KE (MeV) in run)'/
*      ' Average Kinetic energy =',G15.5,'MeV'/)

if (totke .le. 0.D0) then
write(6,470) totke,availke,ncount
470  FORMAT(/,' Stopped in MAIN with TotKE=',G15.5,/,
*      ' AvailKE=',G15.5, /,' Ncount=',I10)
  stop
end if

!-----
!      Sampled source spectrum
!-----
if (imode.ne.0) then
  do ie=2,nsebin
    saspec(ie)=saspec(ie)/float(ncases)
  end do

  write(1,480)
480  FORMAT(/' Comparison between sampled spectrum and original data'
*      /23X,' Sampled Probability',25X,' Sampled Probability'
*      )
  do ie=2,nsebin,2
    if(ie.eq.nsebin) then
      write(1,490) ebint(ie),saspec(ie),ecdf(ie)-ecdf(ie-1)
490  FORMAT(1X,G9.3,' MeV(upper)-- ',2G12.5)
    else
      write(1,500) ebint(ie),saspec(ie),ecdf(ie)-ecdf(ie-1),
*      ebint(ie+1),saspec(ie+1),ecdf(ie+1)-ecdf(ie)
500  FORMAT(1X,G9.3,' MeV(upper)-- ',2G12.5,3X,' ; ',G9.3,
*      ' MeV(upper)-- ',2G12.5)
    end if
  end do

  write(1,510) sposi
510  FORMAT(/' Absorbed energy inside phantom for 100 kV X-ray'/
*      ' Source position ',F10.1,' cm from phantom surface'/
*      ' Within 1cm x 1 cm area after 5 cm air')

  write(1,520) ncases, xhbeam, yhbeam
520  FORMAT(1X,I8,' photons normally incident from front side'/ ' Hal
*      f width of beam is ',G15.5,'cm for X and ',G15.5,'cm for Y')
  end if

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

area=1.D0*1.D0
do kdet=1,ndet
  vol=area*1.D0
  dose(kdet)=deph(kdet)/ncases
  dose2(kdet)=deph2(kdet)/ncases
  doseun(kdet)=dsqrt((dose2(kdet)-dose(kdet)*dose(kdet))/ncases)
  dose(kdet)=dose(kdet)*1.602E-10/vol
  doseun(kdet)=doseun(kdet)*1.602E-10/vol
  depths=kdet-1.0
  depthl=kdet
  write(6,530)depths,depthl,(media(ii,med(kdet+1)),ii=1,24),
*      rhor(kdet+1),dose(kdet),doseun(kdet)
530  FORMAT(' At ',F4.1,'--',F4.1,'cm (' ,24A1,' rho:',F8.4,')=',
*      G13.5,'+-',G13.5,'Gy/incident')
  write(1,530) depths,depthl,(media(ii,med(kdet+1)),ii=1,24),
*      rhor(kdet+1),dose(kdet),doseun(kdet)
  end do

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

faexpa=faexps/ncases
faexp2s=faexp2s/ncases
faexrr=dsqrt((faexp2s-faexpa*faexpa)/ncases)
faexpa=faexpa*1.6E-10/area
faexrr=faexrr*1.6E-10/area
fexpsa=fexps/ncases

```

```

fexps2s=fexps2s/ncases
fexerr=dsqrt((fexps2s-fexpsa*fexpsa)/ncases)
fexpsa=fexpsa*1.6E-10/area
fexerr=fexerr*1.6E-10/area
if (faexpa.gt.0.0) then
  bsfa=fexpsa/faexpa
  bsferr=bsfa*dsqrt((faexrr/faexpa)**2.+(fexerr/fexpsa)**2.)
  write(6,540) faexpa,faexrr,fexpsa,fexerr,bsfa,bsferr
  write(1,540) faexpa,faexrr,fexpsa,fexerr,bsfa,bsferr
540  FORMAT(/' Exposure in free air (using mu_en) =', G15.5,'+-',G15.
* 5,' Gy/incident'/' Exposure at phantom surface (using mu_en) ='
* , G15.5,'+-',G15.5,'Gy/incident'/' Backscattering factor =',G15
* .5,'+-',G15.5)
  else
  write(6,550) faexpa,faexrr,fexpsa,fexerr
  write(1,550) faexpa,faexrr,fexpsa,fexerr
550  FORMAT(/' Exposure in free air (using mu_en) =', G15.5,'+-',G15.
* 5,' Gy/incident'/' Exposure at phantom surface (using mu_en) ='
* , G15.5,'+-',G15.5,'Gy/incident')
  end if

!-----
! Write end of batch information
!-----
  call plotxyz(99,0,0,0.D0,0.D0,0.D0,0.D0,0,0.D0)
  write(39,560)
560  FORMAT('9')
  close(UNIT=39,status='keep')
  go to 380

570  if(imode.ne.0) then
! =====
! call ecnsv1(nlist,nreg,totke)
! =====
  end if

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

! -----
! Close files
! -----
  close(UNIT=1)
  close(UNIT=4)

  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 simple AUSGAB to:
!
! 1) Score energy deposition
! 2) Print out stack information
! 3) Print out particle transport information (if switch is turned on)
!-----

  subroutine ausgab(iarg)

  implicit none

  include 'include/egs5_h.f'           ! Main EGS "header" file
  include 'include/egs5_epcont.f'     ! COMMONs required by EGS5 code
  include 'include/egs5_media.f'

```

```

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

integer                                ! Arguments
* iarg

real*8                                ! Local variables
* cmod,dcon,edepwt,encoae,esing

integer idet,ie,iql,irl

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

!-----
! Keep track of how deep stack gets
!-----
if (np.gt.MXSTACK) then
  write(6,100) np,MXSTACK
100  FORMAT(// ' In AUSGAB, np=',I3,' >= maximum stack',
*      ' allowed which is',I3/1X,79('*')//)
  stop
end if

!-----
! 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

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

!-----
! Score data ate detector region (region 2-21)
!-----
if (irl.ge.2.and.irl.le.nreg-3) then
  idet=irl-1
  if(idet.ge.1.and.idet.le.ndet) then
    depe(idet)=depe(idet)+edepwt/rhor(irl)
  end if
end if

!-----
! Check cross phantom surface
!-----
if (abs(irl-iold).eq.1.and.iq(np).eq.0) then
  if((w(np).gt.0.0.and.irl.eq.2).or.(w(np).le.0.0.and.irl.eq.1))

```

```

* then
  if (dabs(w(np)).ge.0.0349) then
    cmod=dabs(w(np))
  else
    cmod=0.0175
  end if
  esing=e(np)
  dcon=encoea(esing)          ! PHOTX data
  fexps=fexps+e(np)*dcon*wt(np)/cmod
  if (w(np).lt.0.0) latch(np)=1
  if (w(np).gt.0.0.and.latch(np).eq.0) then
    faexp=faexp+e(np)*dcon*wt(np)/cmod
  end if
end if
end if

! -----
! 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,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 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
            write(*,*) 'srzone:1'
            call srzone(xidd,yidd,zidd,iq_np,ir_np,irnext)
c
            if(irnext.ne.ir_np) then
                tval=0.0d0
                irnear=irnext
            else
                tval00=0.0d0
                tval10=10.0d0*tval0
                irlold=ir_np
                irlfg=0
330    continue
                if(irlfg.eq.1) go to 340
                tval00=tval00+tval10
                if(tval00.gt.1.0d+06) then
                    write(6,9000) iq(np),ir(np),x(np),y(np),z(np),
&                                u(np),v(np),w(np),tval00
9000 format(' TVAL00 ERROR : iq,ir,x,y,z,u,v,w,tval=',
&          2I3,1P7E12.5)
                    stop
                    end if
                xidd=x_np+tval00*u_np

```

```

        yidd=y_np+tval00*v_np
        zidd=z_np+tval00*w_np
        call srzold(xidd,yidd,zidd,irlold,irlfg)
        go to 330
340    continue
c
        tval=tval00
        do j=1,10
            xidd=x_np+tval00*u_np
            yidd=y_np+tval00*v_np
            zidd=z_np+tval00*w_np
c
            write(*,*) 'srzone:2'
            call srzone(xidd,yidd,zidd,iq_np,irlold,irnext)
            if(irnext.ne.irlold) then
                tval=tval00
                irnear=irnext
            end if
            tval00=tval00-tval0
        end do
        if(ir_np.eq.irnear) then
            write(0,*) 'ir(np),tval=',ir_np,tval
        end if
    end if
else
    do j=1,itvalm-1
        do i=j+1,itvalm
            if(atval(i).lt.atval(j)) then
                atvaltmp=atval(i)
                atval(i)=atval(j)
                atval(j)=atvaltmp
            endif
        enddo
    enddo
    itvlf=0
    tvalmn=tval
    do jjj=1,itvalm
        if(tvalmn.gt.atval(jjj)) then
            tvalmn=atval(jjj)
        end if
        delhow=cgeps2
        tval0=atval(jjj)+delhow
        xidd=x_np+tval0*u_np
        yidd=y_np+tval0*v_np
        zidd=z_np+tval0*w_np
410    continue
        if(x_np.ne.xidd.or.y_np.ne.yidd.or.z_np.ne.zidd) go to 420
        delhow=delhow*10.d0
        tval0=atval(jjj)+delhow
        xidd=x_np+tval0*u_np
        yidd=y_np+tval0*v_np
        zidd=z_np+tval0*w_np
        go to 410
420    continue
c
        write(*,*) 'srzone:3'
        call srzone(xidd,yidd,zidd,iq_np,ir_np,irnext)
        if((irnext.ne.ir_np.or.atval(jjj).ge.1.).and.
&
            tval.gt.atval(jjj)) THEN
            tval=atval(jjj)
            irnear=irnext
            itvlf=1
            goto 425
        end if
    end do
425    continue
    if(itvlf.eq.0) then
        tval0=cgmnst
        xidd=x_np+tval0*u_np
        yidd=y_np+tval0*v_np
        zidd=z_np+tval0*w_np
430    continue
        if(x_np.ne.xidd.or.y_np.ne.yidd.or.z_np.ne.zidd) go to 440
        tval0=tval0*10.d0
        xidd=x_np+tval0*u_np
        yidd=y_np+tval0*v_np
        zidd=z_np+tval0*w_np
        go to 430
440    continue
        if(tvalmn.gt.tval0) then
            tval=tvalmn

```

```

        else
            tval=tval0
        end if
    end if
end if
ihitcg=0
if(tval.le.ustep) then
    ustep=tval
    ihitcg=1
end if
if(ihitcg.eq.1) THEN
    if(irnear.eq.0) THEN
        write(6,9200) iq(np),ir(np),x(np),y(np),z(np),
&          u(np),v(np),w(np),tval
9200 format(' TVAL ERROR : iq,ir,x,y,z,u,v,w,tval=',2I3,1P7E12.5)
        idisc=1
        itverr=itverr+1
        if(itverr.ge.100) then
            stop
        end if
        return
    end if
    irnew=irnear
    if(irnew.ne.ir_np) then
        call rstnxt(iq_np,ir_np,irnew)
    endif
end if
return
end
!-----last line of subroutine howfar-----
!-----encoea.f-----
! Version: 030831-1300
! Reference: SLAC-265 (p.19-20, Appendix 2)
!-----
!23456789|123456789|123456789|123456789|123456789|123456789|123456789|12
!
! real function encoea(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)
!-----
real function encoea(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
    encoea=enmu(38)
    return
end if
if (energy.lt.hnu(1)) then
    encoea=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)
        encoea=exp(enm0+slope*(ene0-hnu0))
        return
    end if
    if(energy.eq.hnu(i+1)) then
        encoea=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 encoea.f-----
!-----encoew.f-----
! Version: 030831-1300
! Reference: SLAC-265 (p.19-20, Appendix 2)
!-----
!23456789|123456789|123456789|123456789|123456789|123456789|123456789|12
!
! real function encoew(energy)
! Function to evaluate the energy absorption coefficient of water.
! (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)
!-----
real function encoew(energy)

real hnu(36)/0.001,0.0015,0.002,0.003,0.004,0.005,0.006,0.008,
* 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(36)/4065., 1372., 615.2, 191.7, 81.91, 41.88,
* 24.05, 9.915, 4.944, 1.374, 0.5503, 0.1557,
* 0.06947,0.04223,0.03190,0.02597,0.02546,0.02764,
* 0.02967,0.03192,0.03279,0.03299,0.03284,0.03206,
* 0.03103,0.02965,0.02833,0.02608,0.02281,0.02066,
* 0.01915,0.01806,0.01658,0.01566,0.01441,0.01382/

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

integer i

if (energy.gt.hnu(36)) then
    encoew=enmu(36)
    return
end if
if (energy.lt.hnu(1)) then
    encoew=enmu(1)
    return
end if

do i=1,36
    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=log(energy)
        slope=(enm1-enm0)/(hnu1-hnu0)
        encoew=exp(enm0+slope*(ene0-hnu0))
        return
    end if
    if(energy.eq.hnu(i+1)) then
        encoew=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 ENCOEW*****',/, ' E=',G15.5,///)
    return
    end
!-----last line of encoew.f-----
```