

egs5 サンプルプログラム (uccg_phantom.f)
人体中の線量分布計算 (cg Version)
(July 28, 2004, Draft)

平山 英夫、波戸 芳仁

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

Contents

1. Combinatorial geometry	1
1.1. Body の定義	1
1.2. リージョンの定義	1
1.3. リージョン定義の例	2
2. サンプルプログラム uccg_phantom.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	6
3.1.1. include 文及び型式宣言:	6
3.1.2. open 文:	7
3.1.3. subroutine getcg の call:	8
3.1.4. 計算モードの選択:	9
3.1.5. 入射粒子のパラメータ:	9
3.1.6. X 線源の種類を増やす方法:	10
3.1.7. 輸送計算:	11
3.1.8. 統計誤差:	13
3.1.9. 計算結果の出力:	14
3.2. subroutine getcg	14
3.3. subroutine ausgab	15
3.4. subroutine howfar	17
4. ucxyz_phantom.f と uccg_phantom.f の計算速度の比較	17
5. 実習課題	17
5.1. 実習課題 1 : 線源を Cs-137 に変更する	17
5.2. 実習課題 2 : 線源を Co-60 に変更する	17
5.3. 実習課題 3 : 肺のモデルに変更する	17
5.4. 実習課題 4 : 腫瘍を含む肺	17
5.5. 実習課題 5 : 金属の挿入	17
5.6. その他	18
5.7. 実習課題の解答例	19
5.8. 実習課題 1	19
5.9. 実習課題 2	19
5.10. 実習課題 3	19
5.11. 実習課題 4	20
5.12. 実習課題 5	21

1. Combinatorial geometry

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

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

1.3. リージョン定義の例

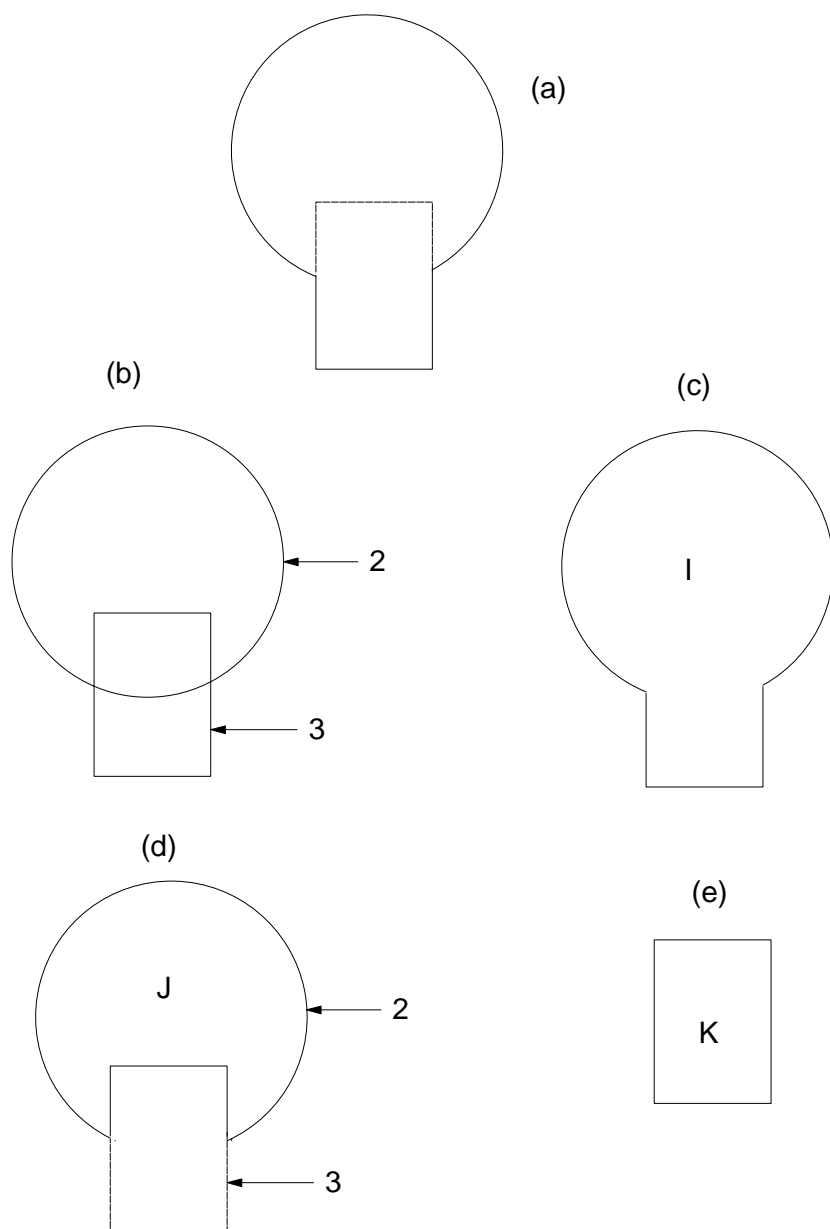


Figure 1: Examples of Combinatorial Geometry Method.

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

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

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

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

リージョン J (図 1d) は、

$$J = +2 - 3.$$

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

リージョン K (図 2e) は、単に

$$K = +3.$$

と記述する。これは、body 3 の属する領域を意味する。

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

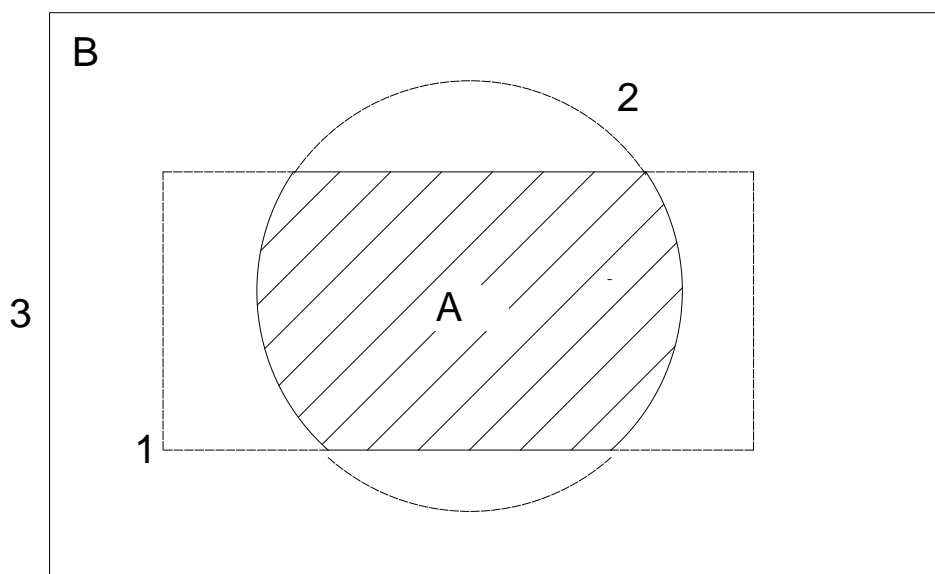


Figure 2: Use of OR operator.

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

$$A = +1 + 2$$

そして

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

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

2. サンプルプログラム uccg_phantom.fの概要

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

2.1. CG 入力データ

ucxyz_phantom.f が、平板によりボクセル形状のリージョンを定義しているのと異なり、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定義している。

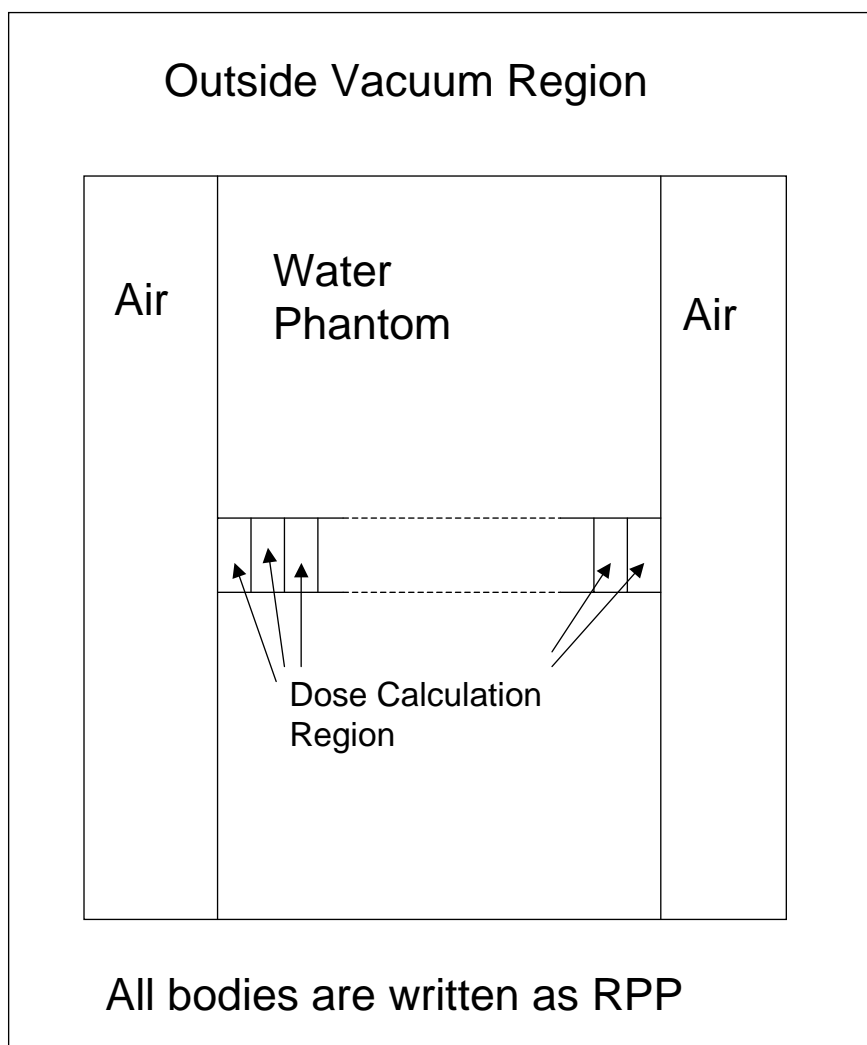


Figure 3: Geometry of uccg_phantom.f.

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	15.00	-5.0	0.00
RPP	2	-15.0	15.0	-15.0	15.00	0.0	20.00
RPP	3	-0.5	0.5	-0.5	0.50	0.0	1.00
RPP	4	-0.5	0.5	-0.5	0.50	1.0	2.00
RPP	5	-0.5	0.5	-0.5	0.50	2.0	3.00

RPP	6	-0.5	0.5	-0.5	0.50	3.0	4.00
RPP	7	-0.5	0.5	-0.5	0.50	4.0	5.00
RPP	8	-0.5	0.5	-0.5	0.50	5.0	6.00
RPP	9	-0.5	0.5	-0.5	0.50	6.0	7.00
RPP	10	-0.5	0.5	-0.5	0.50	7.0	8.00
RPP	11	-0.5	0.5	-0.5	0.50	8.0	9.00
RPP	12	-0.5	0.5	-0.5	0.50	9.0	10.00
RPP	13	-0.5	0.5	-0.5	0.50	10.0	11.00
RPP	14	-0.5	0.5	-0.5	0.50	11.0	12.00
RPP	15	-0.5	0.5	-0.5	0.50	12.0	13.00
RPP	16	-0.5	0.5	-0.5	0.50	13.0	14.00
RPP	17	-0.5	0.5	-0.5	0.50	14.0	15.00
RPP	18	-0.5	0.5	-0.5	0.50	15.0	16.00
RPP	19	-0.5	0.5	-0.5	0.50	16.0	17.00
RPP	20	-0.5	0.5	-0.5	0.50	17.0	18.00
RPP	21	-0.5	0.5	-0.5	0.50	18.0	19.00
RPP	22	-0.5	0.5	-0.5	0.50	19.0	20.00
RPP	23	-0.5	0.5	-0.5	0.50	0.0	20.00
RPP	24	-15.0	15.0	-15.0	15.00	20.0	25.00
RPP	25	-20.0	20.0	-20.0	20.00	-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
- 人体を一様な水でモデル化 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、isemode=0の時は、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、isemode=1の時は、ユニット4の入力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(5cm)×YHBEAM*2(5cm)のビーム。XHBEAM及びYHBEAMは、プログラム中で値を決めている。

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

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

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

この点が、Mortran のマクロ機能により、ユーザーコードで再設定できた EGS4 の場合と最も異なることである。従って、配列の大きさを変更する場合には、egs5 に直接関係する場合は、egs5.0/include/egs5_h.f 内の、その他の場合は、user_dir/user_auxcommons/aux_h.f の当該 parameter 文の値を変更することになる。

最初の設定は、egs に直接関連する include 文である。

```

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

include 'include/egs5_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_switches.f'
include 'include/egs5_stack.f'
include 'include/egs5_thresh.f'

```

[†]これらの設定は、egs5run スクリプトで設定される。


```

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

```

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

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

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

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

```

etaly2.f は、準 egs5 的な扱いになっている common である。最後の、7 つの include 文が、CG に関連したものである。

特定のユーザーコード内で使用する common を次に定義する。

```

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

```

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

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

```

[‡]EGS4 の COMIN マクロに対応する扱いである。

unit 2 の open 文は、X 線データを、xray.dat ファイルから読み込むことを定義しているものである。

3.1.3. subroutine getcg の call: 飛跡表示用データの型式定義 (この例では、CGview を使用することを前提にしている)、nprec=2) に設定してある。リージョンに関するパラメータをクリアするサブルーティンを call した後、CG の入力データを読み込み、処理をサブルーティン geomgt を call する。次に、各種の counter パラメータをクリアサブルーティンと、getcg を call する。getcg は、egs5 で新たに必要なサブルーティンである。このサブルーティンをどのように設定するかは、個々のユーザーコードにより異なるが、最低必要な機能は、pegs の実行と、subroutine hatch を call することである。このユーザーコードでは、物質の指定、カットオフエネルギー、オプション、入射粒子等の設定を全てこのサブルーティンにおいて、ユニット 4 から読み込むデータで設定する様にしている。

これらの処理の前後で、CGview で必要な情報を、egs5job.pict に出力する。

```

!-----
!      initialize cg related parameter
!-----
      nprec=2
!      =====
!      call region_init          ! Initialize some region variables
!      =====

      itbody=0
      irppin=0
      isphin=0
      irccin=0
      itorin=0
      itrclin=0
      izonin=0
      izonad=0
      itverr=0
      igmmax=0
!      ifti = 90
      ifti = 4
      ifto = 6

      if (nprec.eq.2) then
         ifto =39
         write(39,1000)
1000    FORMAT('CSTA')
      end if
      call geomgt(ifti,ifto,igmmax,itbody)
      if (nprec.eq.2) then
1010    write(39,1010)
         FORMAT('CEND')
      end if

!-----
!      Get nreg from cg input data
!-----
      nreg=izonin
      if (nreg.gt.mxreg) then
1020    write(6,1020) nreg,mxreg
         FORMAT(' NREG(=',I12,') must be less than MXREG(=',I12,')' /' Yo
         *u must chang MXREG in include/egs5_h.f.')
         stop
      end if

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

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

      if (nprec.eq.2 ) then

```

```

1030     write(39,1030)
        FORMAT('MSTA')
        write(39,1040) nreg
1040     FORMAT(I4)
        write(39,1050) (med(i),i=1,nreg)
1050     FORMAT(15I4)
        write(39,1060)
1060     FORMAT('MEND')
        end if

```

3.1.4. 計算モードの選択: このユーザーコードでは、最初に説明した様に、飛跡表示用データ作成モード (imode=0) と線量計算モードの両方対応している。モードの選択は、次により、キーボードやり入力したデータで行う。

```

        write(6,1090)
1090     FORMAT(' Key in mode. 0:trajectory display, 1:dose calculation')
        read(5,*) imode

```

線量計算モードにおいては、getcg で作成した物質データ及び、各リージョンの物質情報を出力するようにしている。

3.1.5. 入射粒子のパラメータ: 入射粒子のパラメータを設定する。線源の位置は、キーボードからの入力指定する。

```

        write(6,170)
170     FORMAT(' Key in source position from phantom surface in cm')
        read(5,*) sposi

```

線量計算を行うリージョンの数を入力する。粒子のエネルギーについては、isemode で、xray.dat から読み込んだ 100kV の X 線スペクトルを使用するか、egs5job.inp で設定した値を使用するかどうかを決める。

```

!-----
!   Detector number to score
!-----
        write(6,175) nreg-3
175     format(' Key in number of dose caluculation region.(<=','I5,')')
        read(5,*) ndet

```

```

!-----
!   Source energy sampling mode
!   isemode=0 use xray.dat
!   isemode=1 use egs5job.inp
!-----
        isemode=0

```

imode=0 の時は、xray.dat }からデータを読み込み、読み込んだ確立分布関数(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。

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。

```

!-----
!   Key in half width and height at phantom surface
!-----
        write(6,210)
210     FORMAT(' Key in half width of beam at phantom surface in cm.')
        read(5,*) xhbeam
        write(6,220)
220     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)

```

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。

3.1.6. X線源の種類を増やす方法: ucphantom_rec1k.mor 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.')
```

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')
```

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。

3.1.7. 輸送計算: 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。uccg_phantom.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の位置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。

```
do j=1,ncases                                     ! -----
                                                ! Start of CALL SHOWER loop
                                                ! -----
  icases=j
!-----
! Determine direction (isotropic)
!-----
270  call randomset(w0)
     win=w0*(1.0-wimin)+wimin
     call randomset(phai0)
     phai=pi*(2.0*phai0-1.0)
     sinth=dsqrt(1.00-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 270
     if (sposi.gt.5.0) then
       disair=(sposi-5.0)/win
       xin=disair*uin
       yin=disair*vin
       zin=-5.00
     else
       xin=0.00
       yin=0.00
       zin=-sposi
     end if
     irin=1
!-----
```

```

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

if (isemode.eq.0) then ! use xray.dat
  call randomset(ei0)
  do ie=2,nsebin
    if (ei0.lt.ecdft(ie)) then
      go to 280
    end if
  end do

280  if (ie.gt.nsebin) then
      ie=nsebin
    end if
    saspec(ie)=saspec(ie)+1.D0
    ekin=ebint(ie-1)+(ei0-ecdft(ie-1))*(ebint(ie)-ebint(ie-1))/
*   (ecdft(ie)-ecdft(ie-1))
    wtin = 1.0
  else
    ! use egs5job.inp
    ! Monoenergetic case
    if (isamp .eq. 0) then
      ekin = ekein
      wtin = 1.0
    else if (isamp .eq. 1) then ! Sample discrete energy from CDF
      call randomset(rnnow)
      i=0
290  continue
      i = i + 1
      if(ecdf(i) .le. rnnow) go to 290
      ekin = ebin(i)
      wtin = 1.0
    else if (isamp .eq. 2) then ! Sample DIRECTLY from CDF
      call edistr(ekin)
      wtin = 1.0
    else if (isamp .eq. 3) then ! Sample UNIFORMLY on energy
      ! interval and WEIGHT
      call randomset(rnnow)
      ekin = esam1 + rnnow*delsam
300  continue
      isam = isam + 1
      if (ekin .lt. ebin(isam)) go to 310
      go to 300
310  continue
      wtin = epdf(isam)
    end if
  end if

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

  latchi=0

!      -----
!      Print first NWRITE or NLINES, whichever comes first
!      -----
if (ncount .le. nwrite .and. ilines .le. nlines) then
  ilines = ilines + 1
  write(6,320) etot,xin,yin,zin,uin,vin,win,iqin,irin,idin
320  FORMAT(4G15.7/3G15.7,3I5)
end if

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

!      Added for energy balance tests (SJW)
if(DABS(eparte + partd - ekin)/ekin .gt. 1.d-10) then

```

```

        write(*,330)  icases, eparte, epartd
330      FORMAT('Error on # ',I6,' Escape = ',F9.5,' Deposit = ',F9.5)
        endif

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

        do kkk=1,ndet
            depeh(kkk)=depeh(kkk)+depe(kkk)
            depeh2(kkk)=depeh2(kkk)+depe(kkk)*depe(kkk)
            depe(kkk)=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
                                                ! -----

```

3.1.8. 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。uccg_phantom.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。

このために、ヒストリー毎に、計算すべき量とその自乗の値を保存している。

3.1.9. 計算結果の出力: 得られた結果を処理して打ち出す処理を行う。線量計算モードで、xray.dat から粒子のエネルギーをサンプリングする場合には、最初に、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を出力し、その後、線源の条件 (線源のタイプ、位置)、ヒストリー数を出力する。

```

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

      if (imode.ne.0) then
        write(1,370)
370      FORMAT(//' Comparison between sampled spectrum and original data
*'/ 23X,'   Sampled   Probability',25X,'   Sampled   Probability'
* )
        do ie=2,nsebin,2
          write(1,380) ebint(ie),saspec(ie),ecdft(ie)-ecdft(ie-1),
*   ebint(ie+1), saspec(ie+1),ecdft(ie+1)-ecdft(ie)
380      FORMAT(1X,G9.3,' MeV(upper)-- ',2G12.5,3X, '; ',G9.3,' MeV(upp
*er)-- ',2G12.5)
        end do

        if (isemode.eq.0) then
          write(1,390) sposi
390      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')
        else
          write(1,395) sposi
395      FORMAT(/' Absorbed energy inside phantom for source ',
*   'defined in egs5job.inp '/
*   ' Source position ',F10.1,' cm from phantom surface'/
*   ' Within 1cm x 1 cm area after 5 cm air')
        end if

        write(1,400) ncases, xhbeam, yhbeam
400      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

```

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。また、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。

3.2. subroutine getcg

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

ユニット 4 から読み込むデータは以下の様になっている。

1. Record 1 : タイトル情報 (80 文字)
2. Record 2 : 使用する物質数 (nmed)
3. Record 3 : 物質名 : 24 文字で指定する。pegs 入力データの名前と対応が必要。
4. Record 4 : irlinl から irlinu のリージョンの物質、密度、ecut, pcut の指定。
ecutin 及び pcutin は、カットオフエネルギーで、0 より大きい値の時に、ecut(i), pcut(i) を設定する。0 の時は、デフォルトの値を使用。
irlinll = 0 のデータは、指定モードの終了を意味する。
5. Record 4a : Record 4 で物質が真空でない (medtmp ≠ 0) 場合の各種オプションの設定 (0=off, 1=on)

ipeangsw Switches for PE-angle sampling
 iedgesw K & L-edge fluorescence
 iraysw Rayleigh scattering
 ipolarsw Linearly-polarized photon scattering
 incohrsw S /Z rejection
 iprofrsw Doppler broadening
 mpacrsw electron impact ionization

6. Record 5 : 入射粒子の位置 (xin, yin, zin)
7. Record 6 : 入射リージョン番号
8. Record 7 : 入射粒子の方向余弦 (uin, vin, win) $uin = vin = win = 0$ の時は、等方線源。
9. Record 8 : 最初の random number seed(ixx, jxx) を指定。ixx = 0, の時は、ixx= 123457 に、jxx = 0, の時は、jxx= 654321 に設定する。
10. Record 9 : ヒストリー数 (ncases) の指定。
11. Record 10 : 入射粒子の運動エネルギー (ekein:MeV), 電荷 (iqin) 及びサンプリング方法 (isamp) の指定。
isamp = 0 の時は、*isamp* = 0 の単一エネルギー、*isamp* = 1 の時は、 離散エネルギーからサンプリングを行い、*isamp* = 2 の時は、スペクトルからサンプリングを行い、*isamp* = 3 の時は、一様サンプリングを行いウエイトを使用する手法を用いることを意味する。*isamp* ≠ 0 の時は、以下のデータが必要である。
12. Record 10a : 最低エネルギーの指定。 (*isamp* > 1 の時)
13. Record 10b : 各エネルギービンの最大値 (ebin(i)) とビンに対応する確率 (epdf(i)) の指定。
 ブランク又は、0.0 は、指定の終了を意味する。
14. Record 11 : トラッキング状況を設定するフラグ (iwatch) の指定。
iwatch = 0: トラッキングなし。 *iwatch* = 1: 反応毎のトラッキング、 *iwatch* = 2: ステップ毎のトラッキング
15. Record 12 : 制動輻射 (ibrdst) 及び電子対生成 (iprdst) の際の角度分布オプションの設定及びスプリットングパラメータ (ibrspl, nbrspl) の設定。
 ibrdst=0 制動輻射で、デフォルト値 ($\theta = m/E$) を使用
 ibrdst=1 制動輻射で、サンプリング使用 (recommended)
 iprdst=0 電子対生成で、デフォルト値 ($\theta = m/E$) を使用
 iprdst=1 電子対生成で、low-order distribution を使用
 iprdst=2 電子対生成で、推奨のサンプリングを使用
 ibrspl=0 スプリットング使用せず
 ibrspl=1 nbrspl にスプリットング
16. Record 13 : 電子輸送に使用するパラメータ (estepe, estepe2) の指定

3.3. subroutine ausgab

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

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

iarg < 5 の場合には、リージョン 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
!-----
!
! 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 (irl.ne.irold.and.iq(np).eq.0) then
!   if((w(np).gt.0.0.and.irl.eq.2).or.(w(np).le.0.0.and.irold.eq.
* 2)) then
!     if (dabs(w(np)).ge.0.0349) then
!       cmod=dabs(w(np))
!     else
!       cmod=0.0175
!     end if
!     esing=e(np)
!     dcon=encoa(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

```

3.4. subroutine howfar

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

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

uccg_phantom.f では、cg ルーチンを使用している。

4. ucxyz_phantom.f と uccg_phantom.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 に較べ、計算時間が長いという問題がある。対象とする問題によって、違いは異なるが、ucxyz_phantom.f と uccg_phantom.f で全く同じ条件の計算を行うと、ucxyz_phantom.f の方が 2.5 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、1.117MeV と 1.332MeV 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、その背後の 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。

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。

[§]CG の場合の飛躍的なスピードアップ (この場合には、約 5 倍) は、杉田氏の改良によってなされたものである。

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

5.7. 実習課題の解答例

5.8. 実習課題 1

1. 線源エネルギー選択モード isemode=0 を isemode=1 に変更する。
2. uccg_phantom.data の 68 行目の ekinin の値を 0.662 に変更する。
3. uccg_phantom.data を別な名前で保存し、egs5run の実行時にユニット 4 のファイルとして、保存したファイル名を指定する。
4. 線量計算を行いリージョン数として 20 を入力する。

5.9. 実習課題 2

1. isemode=1 の状態で、uccg_phantom.data の 68 行目の ekinin の値を 1.332 に、isamp を 1 に変更する。
2. 上記の後に、

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

を挿入する。

3. uccg_phantom.data を別な名前で保存し、egs5run の実行時にユニット 4 のファイルとして、保存したファイル名を指定する。
4. 線量計算を行いリージョン数として 20 を入力する。

5.10. 実習課題 3

1. cg 入力データ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 -0.5     0.5     -0.5     0.5     0.0    16.00
RPP 20 -15.0    15.0    -15.0    15.0    16.0    21.00
RPP 21 -20.0    20.0    -20.0    20.0   -20.0    40.00
RPP 22 -15.0    15.0    -15.0    15.0     3.0    13.00
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          +22 -19
Z19          +2  -19  -22
Z20          +20
Z21          +21 -1  -2  -20

```

2. ファントムの物質指定に関する以下のデータ

```

2  22  1  0.  0.00  0.0  irlinl,irlinh,med,rho,ecut,pcut
1  1  0  0  0  0  0  peang,edge,ray,pola,incoh,prof,impac

```

を

```

2  4  1  0.  0.00  0.0  tissue
1  1  0  0  0  0  0  peang,edge,ray,pola,incoh,prof,impac
5  14  1  0.3  0.00  0.0  lung
1  1  0  0  0  0  0  peang,edge,ray,pola,incoh,prof,impac
15  17  1  0.  0.00  0.0  tissue
1  1  0  0  0  0  0  peang,edge,ray,pola,incoh,prof,impac
18  18  1  0.3  0.00  0.0  lung
1  1  0  0  0  0  0  peang,edge,ray,pola,incoh,prof,impac
19  19  1  0.  0.00  0.0  tiuue
1  1  0  0  0  0  0  peang,edge,ray,pola,incoh,prof,impac

```

に変更する。

3. uccg_phantom.data を別な名前で保存し、egs5run の実行時にユニット 4 のファイルとして、保存したファイル名を指定する。
4. 線量計算を行いリージョン数として 16 を入力する。

5.11. 実習課題 4

1. cg 入力データ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 -0.5  0.5  -0.5  0.5  0.0 16.00
RPP 20 -15.0  15.0  -15.0  15.0  16.0 21.00
RPP 21 -20.0  20.0  -20.0  20.0 -20.0 40.00
RPP 22 -15.0  15.0  -15.0  15.0  3.0 13.00
RPP 23 -15.0  15.0  -15.0  15.0  6.0  8.00
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         +22 -19 -23
Z19         +23 -19
Z20         +2 -19 -22
Z21         +20
Z22         +21 -1 -2 -20
END

```

2. ファントムの物質指定に関する以下のデータ

```

2  22  1  0.  0.00  0.0  irlinl,irlinh,med,rho,ecut,pcut
1  1  0  0  0  0  0  peang,edge,ray,pola,incoh,prof,impac

```

を

```

2  4  1  0.  0.00  0.0  tissue
1  1  0  0  0  0  0  peang,edge,ray,pola,incoh,prof,impac
5  7  1  0.3  0.00  0.0  lung
1  1  0  0  0  0  0  peang,edge,ray,pola,incoh,prof,impac
8  9  1  0.0  0.00  0.0  tumor
1  1  0  0  0  0  0  peang,edge,ray,pola,incoh,prof,impac
10 14 1  0.3  0.00  0.0  lung
1  1  0  0  0  0  0  peang,edge,ray,pola,incoh,prof,impac
15 17 1  0.  0.00  0.0  tissue
1  1  0  0  0  0  0  peang,edge,ray,pola,incoh,prof,impac
18 18 1  0.3  0.00  0.0  lung
1  1  0  0  0  0  0  peang,edge,ray,pola,incoh,prof,impac
19 19 1  0.  0.00  0.0  tumor
1  1  0  0  0  0  0  peang,edge,ray,pola,incoh,prof,impac
20 20 1  0.  0.00  0.0  irlinl,irlinh,med,rho,ecut,pcut
1  1  0  0  0  0  0  tissue

```

に変更する。

3. uccg_phantom.data を別な名前で作成し、egs5run の実行時にユニット 4 のファイルとして、保存したファイル名を指定する。
4. 線量計算を行いリージョン数として 16 を入力する。

5.12. 実習課題5

1. uccg_phantom.inp に次のデータを追加し、別な名前で作成する。

```

ELEM
&INP IAPRIM=1,EFRACH=0.05,EFRACL=0.20,IRAYL=1,IBOUND=0,INCOH=0,
ICPROF=0,IMPACT=0 /END
FE-IAPRIM FE
FE
ENER
&INP AE=0.521,AP=0.010,UE=2.511,UP=2.0 /END

```

```
PWLF
&INP /END
DECK
&INP /END
ELEM
```

2. 物質数を '2' から '3' に変え、3 番目の物質として

```
FE-IAPRIM media(j,3) (24A1)
```

を挿入する。

3. 下記の指定

```
2 22 1 0. 0.00 0.0 tissue
1 1 0 0 0 0 0 peang,edge,ray,pola,incoh,prof,impac
```

を

```
2 6 1 0. 0.00 0.0 tissue
1 1 0 0 0 0 0 peang,edge,ray,pola,incoh,prof,impac
7 8 3 0. 0.00 0.0 iron
1 1 0 0 0 0 0 peang,edge,ray,pola,incoh,prof,impac
9 22 1 0. 0.00 0.0 tissue
1 1 0 0 0 0 0 peang,edge,ray,pola,incoh,prof,impac
23 23 3 0. 0.00 0.0 iron
1 1 0 0 0 0 0 peang,edge,ray,pola,incoh,prof,impac
```

に変更する。

4. uccg_phantom.data を別な名前で作成し、egs5run の実行時にユニット 4 のファイルとして、保存したファイル名を指定する。
5. 線量計算を行いリージョン数として 20 を入力する。

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).


```

include 'include/egs5_uphiot.f'
include 'include/egs5_useful.f'
include 'include/randomm.f'

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

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

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

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

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

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

real*8                                ! Local variables
* 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,ii,iii,icases,idin,ie,ifti,ifto,igmmax,imed,ireg,isam,
* isemode,itbody,ixtype,izonad,j,k,kkk,nlist,nnn,nsebin

!
! -----
! 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 open after
! getcg 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')

!-----
! Initialize cg related parameter
!-----
npreci=2
=====
!

```

```

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

      itbody=0
      irppin=0
      isphin=0
      irccin=0
      itorin=0
      itrccin=0
      izonin=0
      izonad=0
      itverr=0
      igmmax=0
      ifti = 4
      ifto = 6

      if (npreci.eq.2) then
        ifto =39
        write(39,1000)
1000   FORMAT('CSTA')
      end if
      call geomgt(ifti,ifto,igmmax,itbody)
      if (npreci.eq.2) then
1010   write(39,1010)
        FORMAT('CEND')
      end if

!-----
!   Get nreg from cg input data
!-----
      nreg=izonin
      if (nreg.gt.mxreg) then
1020   write(6,1020) nreg,mxreg
        FORMAT(' NREG(=',I12,') must be less than MXREG(=',I12,')' /' Yo
        *u must chang MXREG in include/egs5_h.f.')
        stop
      end if

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

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

      if (npreci.eq.2 ) then
1030   write(39,1030)
        FORMAT('MSTA')
        write(39,1040) nreg
1040   FORMAT(I4)
        write(39,1050) (med(i),i=1,nreg)
1050   FORMAT(15I4)
        write(39,1060)
1060   FORMAT('MEND')
      end if

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

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

!-----
!   Output medium and region information to file for calculation mode.
!-----
      if (imode.ne.0) then
100   write(1,100)
        FORMAT(' Quantities associated with each media:')
        do j=1,nmed
          write(1,110) (media(i,j),i=1,24)
        end do
      end if

```

```

110     FORMAT(/,1X,24A1)
        write(1,120) rho(j),rlc(j)
120     FORMAT(5X,' Rho=',G15.7,' g/cm**3      RLC=',G15.7,' cm')
        write(1,130) ae(j),ue(j),ap(j),up(j)
130     FORMAT(5X,' AE=',G15.7,' MeV      UE=',G15.7,' MeV'/ 5X,' AP=',G
*      15.7,' MeV      UP=',G15.7,' MeV')
        end do

        write(1,140)
140     FORMAT(/,' Information of medium and cut-off for each region')
        do i=1,nreg
            if (med(i).eq.0) then
                write(1,150) i
150             FORMAT(' Medium(' ,I3,')= Vacuum')
            else
                write(1,160) i,(media(ii,med(i)),ii=1,24),ecut(i),pcut(i),
*                rhor(i)
160             FORMAT(' Medium(' ,I3,')=',24A1,' ECUT=',G10.5,' MeV, PCUT=',G
*                10.5,' MeV, density=',F10.3)
            end if
        end do

        end if

!-----
!      Define source from phantom surface.
!-----
        write(6,170)
170     FORMAT(' Key in source position from phantom surface in cm')
        read(5,*) sposi

!      =====
        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.D0
        fexps=0.D0
        fexpss=0.D0
        fexps2s=0.D0

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

        iii=0

!-----
!      Detector number to score
!-----
        write(6,175) nreg-3
175     format(' Key in number of dose calculation region.(<=',I5,')')
        read(5,*) ndet

!-----
!      Source energy sampling mode
!      isemode=0 use xray.dat
!      isemode=1 use egs5job.inp
!-----
        isemode=0

        if (isemode.eq.0) then      ! use xray.dat
!-----
!      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
        end if

```

```

        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

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

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

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

!-----
!   Source condition redefine
!-----
      xin=0.D0
      yin=0.D0
      zin=-sposi
      uin=0.D0
      vin=0.D0
      win=1.D0

!-----
!   Key in half width and height at phantom surface
!-----
      write(6,210)
210   FORMAT(' Key in half width of beam at phantom surface in cm. ')
      read(5,*) xhbeam
      write(6,220)
220   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)

      write(6,230)
230   FORMAT('/', ' ENERGY/COORDINATES/DIRECTION COSINES/ETC.', '/',
*          6X, 'E', 16X, 'X', 14X, 'Y', 14X, 'Z' /
*          1X, 'U', 14X, 'V', 14X, 'W', 9X, 'IQ', 4X, 'IR', 3X, 'IARG', /)

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

!-----
!   Key in history number
!-----
240   write(6,250)
250   FORMAT(' Key in number of cases (0 means end of calculation.)')
      read(5,*) ncases
      if (ncases.eq.0) go to 450

      iii=iii+1
      close(39,status='keep')
      open(39,file='egs5job.pic',access='append')
      write(39,260) iii
260   FORMAT('0',I5)

```

```

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

do j=1,ncases
! -----
! Start of CALL SHOWER loop
! -----
    icsases=j
!-----
! Determine direction (isotropic)
!-----
270  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 270
    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

    irin=1
!-----
! Select incident energy
!-----
    eparte = 0.d0          ! Initialize some energy-balance
    epartd = 0.d0          ! tallying parameters (SJW)

    if (isemode.eq.0) then ! use xray.dat
        call randomset(ei0)
        do ie=2,nsebin
            if (ei0.lt.ecdft(ie)) then
                go to 280
            end if
        end do

280  if (ie.gt.nsebin) then
        ie=nsebin
        end if
        saspec(ie)=saspec(ie)+1.D0
        ekin=ebint(ie-1)+(ei0-ecdft(ie-1))*(ebint(ie)-ebint(ie-1))/
*      (ecdft(ie)-ecdft(ie-1))
        wtin = 1.0
    else
        ! use egs5job.inp
        ! Monoenergetic case
        if (isamp .eq. 0) then
            ekin = ekein
            wtin = 1.0
        else if (isamp .eq. 1) then
            ! Sample discrete energy from CDF
            call randomset(rnnow)
            i=0
290  continue
            i = i + 1
            if(ecdft(i) .le. rnnow) go to 290
            ekin = ebin(i)
            wtin = 1.0
        else if (isamp .eq. 2) then
            ! Sample DIRECTLY from CDF
            call edistr(ekin)
            wtin = 1.0
        else if (isamp .eq. 3) then
            ! Sample UNIFORMLY on energy
            ! interval and WEIGHT
            call randomset(rnnow)
            ekin = esam1 + rnnow*delsam
            isam = 0
300  continue
            isam = isam + 1
            if (ekin .lt. ebin(isam)) go to 310
            go to 300
310  continue
            wtin = epdf(isam)

```

```

    end if
end if

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

latchi=0

! -----
! Print first NWRITE or NLINES, whichever comes first
! -----
if (ncount .le. nwrite .and. ilines .le. nlines) then
    ilines = ilines + 1
    write(6,320) etot,xin,yin,zin,uin,vin,win,iqin,irin,idin
320   FORMAT(4G15.7/3G15.7,3I5)
end if

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

! Added for energy balance tests (SJW)
if(DABS(eparte + epard - ekin)/ekin .gt. 1.d-10) then
    write(*,330) icases, eparte, epard
330   FORMAT('Error on # ',I6,' Escape = ',F9.5,' Deposit = ',F9.5)
endif

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

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

faexps=faexps+faexp
faexp2s=faexp2s+faexp*faexp
faexp=0.0
fexpss=fexpss+fexpss
fexpss2s=fexpss2s+fexpss*fexpss
fexpss=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
                                         ! -----

tt=etime(tarray)
tt1=tarray(1)
cputime=tt1-tt0
write(1,340) cputime
340   format(/' Elapsed Time (sec)=' ,G15.5)

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

! -----
! Write out the results
! -----
write(1,350) ncount,ncases,totke,iseed1,iseed2
350   FORMAT(/' Ncount=',I10,' (actual cases run)',/,
*         ' Ncases=',I10,' (number of cases requested)',/,
*         ' TotKE =',G15.5,' (total KE (MeV) in run)'/
*         ' Last iseed1 =',I12,', iseed2 =',I12)

if (totke .le. 0.D0) then
    write(6,360) totke,availke,ncount
360   FORMAT(/' Stopped in MAIN with TotKE=',G15.5,/,
*         ' AvailKE=',G15.5, /, ' Ncount=',I10)

```



```

        stop
    end if

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

    if (imode.ne.0) then
        write(1,370)
370    FORMAT(//' Comparison between sampled spectrum and original data
*'/ 23X,'   Sampled   Probability',25X,'   Sampled   Probability'
* )
        do ie=2,nsebin,2
            write(1,380) ebint(ie),saspec(ie),ecdft(ie)-ecdft(ie-1),
* ebint(ie+1), saspec(ie+1),ecdft(ie+1)-ecdft(ie)
380    FORMAT(1X,G9.3,' MeV(upper)-- ',2G12.5,3X,' ; ',G9.3,' MeV(upp
*er)-- ',2G12.5)
            end do

            if (isemode.eq.0) then
                write(1,390) sposi
390    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')
            else
                write(1,395) sposi
395    FORMAT(/' Absorbed energy inside phantom for source ',
* 'defined in egs5job.inp '/
* ' Source position ',F10.1,' cm from phantom surface'/
* ' Within 1cm x 1 cm area after 5 cm air')
            end if

            write(1,400) ncases, xhbeam, yhbeam
400    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 kkk=1,ndet
        vol=area*1.D0
        dose(kkk)=depeh(kkk)/ncases
        dose2(kkk)=depeh2(kkk)/ncases
        doseun(kkk)=dsqrt((dose2(kkk)-dose(kkk)*dose(kkk))/ncases)
        dose(kkk)=dose(kkk)*1.602E-10/vol
        doseun(kkk)=doseun(kkk)*1.602E-10/vol
        depths=kkk-1.0
        depthl=kkk
        write(6,410)depths,depthl,(media(ii,med(kkk+1)),ii=1,24),
* rhor(kkk+1),dose(kkk),doseun(kkk)
410    FORMAT(' At ',F4.1,'--',F4.1,'cm (' ,24A1,' ,rho:',F8.4,')=',
* G13.5,'+-',G13.5,'Gy/incident')
        if (imode.ne.0) then
            write(1,410) depths,depthl,(media(ii,med(kkk+1)),ii=1,24),
* rhor(kkk+1),dose(kkk),doseun(kkk)
        end if
    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=fexpss/ncases
    fexp2s=fexp2s/ncases
    fexerr=dsqrt((fexp2s-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(6,420) faexpa,faexrr,fexpsa,fexerr,bsfa,bsferr
  write(1,420) faexpa,faexrr,fexpsa,fexerr,bsfa,bsferr
420  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(6,430) faexpa,faexrr,fexpsa,fexerr
    write(1,430) faexpa,faexrr,fexpsa,fexerr
430  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
!-----
write(39,440)
440  FORMAT('9')
call plotxyz(99,0,0,0.D0,0.D0,0.D0,0.D0,0,0.D0)
close(UNIT=9,status='keep')
go to 240

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

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

!-----
! Close files
!-----
close(UNIT=4)
close(UNIT=6)
close(UNIT=7)

stop
end

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

!-----getcg.f-----
! Version: 040630-1300 KEK-LSCAT
! Reference: KEK Internal 2000-1
!23456789|123456789|123456789|123456789|123456789|123456789|123456789|12
!-----
! Auxiliary subroutine for use with the EGS5 Code System
!-----
! This is a data-entry subprogram for use with a cg geometry.
! The data input is similar to that in ucrz.
! However, this version is designed specifically to utilize
! cg geometry.
!-----
!-----
! SUBROUTINE ARGUMENT
!-----
! nreg      Number of regions in geometry (determined by data input).
!-----
! UNIT ASSIGNMENTS
!-----
! Unit 4    Input file.
! Unit 6    Output file.
! Unit 8    Echoes input cross-section data (assign a null file).
! Unit 12   Input cross-section file from PEGS5.
!-----

```

```

INPUT FILE
=====
CG geometry related data must be written before following data.
=====
Record 1  title (80A1)          Title line.
-----
Record 2  nmed                  Number of media in problem.
-----
Record 3  media(j,i) (24A1)    Media names (j=1,24, I=1,nmed lines).
-----
Record 4  irlinl,irlinu,medtmp, rhotmp, ecutin, pcutin
----- (3I5,3F10.3)              Set material for region from irlinl to ielinh.
                                medtmp : material number
                                rhotmp : If rhotmp=0.0, the default
                                value for that medium is used.
                                ecutin, pcutin : KINETIC energy cutoffs
                                for electrons and photons, respectively,
                                in MeV. If > 0, ecut(i) and pcut(i) are
                                set. Otherwise ae and ap are used (default).
                                irlinl =0 means end of define.

                                If medtmp not 0, following data follows.
-----
Record 4a  ipeangsw,           Switches for PE-angle sampling,
----- iedgesw,                K & L-edge fluorescence,
          iraysw,              Rayleigh scattering,
          ipolarsw,           Linearly-polarized photon scattering,
          incohsw,            S/Z rejection,
          iprofrsw,           Doppler broadening,
          impacrsw            electron impact ionization (0=off, 1=on).
          (7I5)
...+....1....+....2....+....3....+....4....+....5....+....6....+....7..
Record 5  xin,yin,zin          Incident X,Y,Z coordinates (cm).
-----
Record 6  irin                 Incident region.
-----
Record 7  uin,vin,win          Incident direction cosines (U,V,W).
----- If uin=vin=win=0, isotropic.
Record 8  ixj,jxx              Starting random number seeding.
----- If ixj = 0, ixj is set to 123457.
          If jxx = 0, jxx is set to 654321.
-----
Record 9  ncases               Number of cases.
-----
Record 10  ekein,iqin,isamp    Kinetic energy (MeV), charge of inci-
----- dent beam, and sampling switch. If
                                isamp=0, a monoenergetic beam (ekein)
                                will be used. Otherwise, a spectrum
                                input must follow (Records 10a through
                                10b), which will be sampled from discrete
                                energy (isamp=1), directly (isamp=2) or
                                uniformly over the energy range (isamp=3)
                                with weighting factor.
-----
Record 10a ebinmin             Only required when isamp>1(see above).
----- Lowest energy (MeV) in spectrum.
Record 10b ebin(i),epdf(i)    Only required when isamp>0(see above).
----- ebin(i) is 'discrete energy' with epdf(i)
                                for isamp=1. ebin (i) is 'top-edge' of
                                each energy bin (MeV) and epdf(i) is the
                                corresponding probability for the bin
                                for isamp > 1.
                                For example, a cross section (mb) can
                                be used for epdf (but do not divide it
                                by dE). The last card is a delimiter
                                and should be blank (or contain 0.0).
                                The i-subscript runs from 1 to nebin
                                (nebin calculated after the delimiter)
-----
Record 11  iwatch              Switch for tracking events with swatch:
----- (0=No, 1=each interaction,
          2=each step)
-----

```

```

! Record 12 ibrdst,iprdst,      Switches for bremsstrahlung and pair
! ----- ibrspl,nbrspl      production ANGLE SAMPLING, and brems-
!                               strahlung SPLITTING:
!                               ibrdst=0 No (use default: theta=m/E)
!                               1 Yes (recommended)
!                               iprdst=0 No (use default: theta=m/E)
!                               1 Yes (low-order distribution)
!                               2 Yes (recommended)
!                               ibrspl=0 No
!                               1 Yes (NBR SPL=splitting factor)
! -----
! Record 13 estepe,estepe2
! -----

```

```

subroutine getcg(nreg)
implicit none
include 'include/egs5_h.f'           ! Main EGS "header" file
include 'include/egs5_bounds.f'     ! COMMONs required by EGS5 code
include 'include/egs5_brempr.f'
include 'include/egs5_edge.f'
include 'include/egs5_eicom.f'
include 'include/egs5_elec.f'
include 'include/egs5_media.f'
include 'include/egs5_misc.f'
include 'include/egs5_switches.f'
include 'include/egs5_thresh.f'
include 'include/egs5_useful.f'
include 'include/egs5_userpr.f'
include 'include/egs5_usersc.f'
include 'include/egs5_uservr.f'
include 'include/egs5_userxt.f'

include 'pegscommons/mscom.f'       ! PEGS common
include 'user_auxcommons/aux_h.f'   ! Auxiliary-code "header" file
include 'user_auxcommons/edata.f'
include 'user_auxcommons/nfac.f'
include 'user_auxcommons/instuf.f'
include 'user_auxcommons/watch.f'

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

integer nreg                        ! Arguments

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

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

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

100 write(6,100)
   FORMAT(/,T25,'+-----+',
*        /,T25,'| EGS5 User Code using subroutine Getcg |',
*        /,T25,'+-----+',
*        /,T25,'| NOTE: cg geometry. |',
*        /,T25,'+-----+',
*        //)

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

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

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

```

```

! =====
! -----
! Record 1: title
! -----
      read(4,110) title
110   FORMAT(80A1)
      write(6,120) title
      write(1,120) title
120   FORMAT(' TITLE: '/1X,80A1/)

! -----
! Record 2: nmed
! -----
      read(4,*) nmed
      if (nmed.gt. MXMED) then
130   write(6,130) nmed
      FORMAT(' *** Stopped in Getcg with nmed=',I5,' > MXMED')
      stop
      end if
      write(6,140) nmed
      write(1,140) nmed
140   FORMAT(' nmed=',I5,/)

! -----
! Record 3: media
! -----
      do i=1,nmed
      read(4,150) (media(j,i),j=1,24)
150   FORMAT(24A1)
      write(6,160) i,(media(j,i),j=1,24)
      write(1,160) i,(media(j,i),j=1,24)
160   FORMAT(' MEDIUM=',I5,' ==> ',24A1)
      end do

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

! -----
! Record 4  irlinl, irlinu, medtmp, rhotmp, ecutin, pcutin
! -----
! Define to each region
! -----

170   continue
      read(4,180) irlinl,irlinu,medtmp,rhotmp,ecutin,pcutin
180   FORMAT(3I5,3F10.3)
      if (irlinl.eq. 0) go to 250

      if (medtmp.ne.0) then
! -----
! Record 4a:  ipeangsw,iedgesw,iraysw,ipolarsw,
!             incohrrsw,iprofrsw,impacrrsw
! -----
      read(4,200) ipeangsw,iedgesw,iraysw,ipolarsw,incohrrsw,
* iprofrsw,impacrrsw
200   FORMAT(7I5)

      write(6,210) irlinl,irlinu,medtmp,rhotmp,ecutin,pcutin
      write(1,210) irlinl,irlinu,medtmp,rhotmp,ecutin,pcutin
210   FORMAT(' Region from',I5,' to',I5,': medium =',I5,',' , rhoh=',
*          G15.5/11X,' ecut =',G15.5,',' , pcut =',G15.5)

      write(6,220) ipeangsw,iedgesw,iraysw
      write(1,220) ipeangsw,iedgesw,iraysw
220   FORMAT(11X,' iphter=',I3,3X,' iedgfl=',I3,3X,' iraylr=',I3)
      write(6,230) ipolarsw,incohrrsw,iprofrsw,impacrrsw
      write(1,230) ipolarsw,incohrrsw,iprofrsw,impacrrsw
230   FORMAT(11X,' lpolar=',I3,3X,' incohrr=',I3,3X,' iprofr=',I3,
*          3X,' impacrr=',I3)
      else
      write(6,240) irlinl
      write(1,240) irlinl
240   FORMAT(' Region =',I5,' is vacuum')
      end if

      do irlin=irlinl,irlinu

```

```

med(irlin)=medtmp
if (medtmp.ne.0) then
  if(rhotmp.gt.0.) then
    rhor(irlin) = rhotmp
  end if
  if(ecutin.gt.0.) then
    ecut(irlin) = pcutin
  end if
  if(pcutin.gt.0.) then
    pcut(irlin) = pcutin
  end if
  iphter(irlin) = ipeangsw
  iedgfl(irlin) = iedgesw
  iraylr(irlin) = iraysw
  lpolar(irlin) = ipolarsw
  incohr(irlin) = incohrrsw
  iprofr(irlin) = iprofrsw
  impac(irlin) = impacrrsw
end if
end do
go to 170

250  continue

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

      write(6,260) xin,yin,zin
      write(1,260) xin,yin,zin
260  FORMAT(/,' xin=',G15.7,5X,'yin=',G15.7,5X,'zin=',G15.7
*         /' (incident coordinates)')

! -----
! Record 5: irin
! -----
      read(4,*) irin
      write(6,270) irin
      write(1,270) irin
270  FORMAT(/,' irin=',I5,' (incident region)')

! -----
! Record 6: uin,vin,win
! -----
      read(4,*) uin,vin,win
      write(6,300) uin,vin,win
      write(1,300) uin,vin,win
300  FORMAT(/,' uin=',G15.7,5X,'vin=',G15.7,5X,'win=',G15.7,
*         /' (incident direction cosines)')

! SJW 02-May-2002 Not needed for EGS5
! -----
! Record 7: ixx,jxx
! -----
      read(4,*) ixx,jxx
      if (ixx .eq. 0) ixx = 123457           ! Default seed
      if (jxx .eq. 0) jxx = 654321           ! Default seed
      write(6,310) ixx,jxx
      write(1,310) ixx,jxx
310  FORMAT(/,' ixx=',I12,5X,'jxx=',I12,
*         /' (starting random-number seeds)')

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

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

! -----
! Record 8: ncases
! -----
      read(4,*) ncases
      write(6,320) ncases

```

```

320   write(1,320) ncases
      FORMAT(/,' ncases=',I12)
! -----
! Record 9: ekein,iqin,isamp
! -----
      read(4,*) ekein,iqin,isamp
      if (isamp .eq. 0) then
! -----
! Monoenergetic case
! -----
          write(6,330) iqin,ekein
          write(1,330) iqin,ekein
330   *   FORMAT(/,' MONOENERGETIC case has been selected with:',
*         //,' iqin=',I5,' (incident charge of beam)',
*         /,' ekein=',G15.5,' MeV (incident kinetic energy)')

      else if (isamp .gt. 0) then
! -----
! Energy spectrum case
! -----
! -----
! Record 9a: ebinmin
! -----
      if(isamp.ne.1) then
          read(4,*) ebinmin          ! Lowest energy in spectrum (MeV)
          write(6,340) iqin,ebinmin
          write(1,340) iqin,ebinmin
340   *   FORMAT(/,' Energy-SPECTRUM case has been selected with:',
*         //,' iqin=',I5,' (incident charge of beam)',
*         /,' ebinmin=',F10.3,' MeV (lowest energy bin)')

      end if

      if (isamp .eq. 1) then
          write(6,350) isamp
          write(1,350) isamp
350   *   FORMAT(' isamp =',I2,' (Sample from discrete energy)')
      elseif (isamp .eq. 2) then
          write(6,355) isamp
          write(1,355) isamp
355   *   FORMAT(' isamp =',I2,' (DIRECT-sampling over energy range)')
      else if (isamp .eq. 3) then
          write(6,360) isamp
          write(1,360) isamp
360   *   FORMAT(' isamp =',I2,
*         ' (UNIFORM-sampling over energy range) with WEIGHTING')
      end if

! -----
! Record 9b: ebin(i),epdf(i)
! -----
      i = 0
370   continue
! -----
! Start of energy-spectrum input loop
! -----
          i = i + 1
          if (i .gt. MXEBIN) then
              write(6,380) i
              write(1,380) i
380   *   FORMAT(/,' Stopped in getcg with I=',I6,' > MXEBIN')
              stop
          end if
          read(4,*) ebin(i),epdf(i)      ! ebin(i) is top-edge of bin
          if (i .gt. 1 .and. ebin(i) .le. ebin(i-1)) then
              go to 410
          else if (i .eq. 1 .and. ebin(i) .le. ebinmin) then
              go to 390
          end if
          go to 370

390   continue          ! Reach here when a read-error occurs
          write(6,400)
          write(1,400)
400   *   FORMAT(/,' Stopped in getcg with spectrum read-error')
          stop

410   continue          ! Reach here when delimiter card has been read

```

```

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

write(6,420) (i,ebin(i),epdf(i),ecdf(i),i=1,nebin)
write(1,420) (i,ebin(i),epdf(i),ecdf(i),i=1,nebin)
420  *  FORMAT(/,' BIN UPPER ENERGY PROBABILITY CUMULATIVE ',
*      /,' # (MeV) PROBABILITY',
*      /,'(I4,3X,F10.3,2F16.4)')

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

write(6,430) esam1,esam2
write(1,430) esam1,esam2
430  *  FORMAT(/,' Energy-sampling interval is:',/,'
*      ' esam1 =',G15.5,' MeV to esam2 =',G15.5,' MeV',/)
else
  write(6,440) isamp
  write(1,440) isamp
440  *  FORMAT(/,' Stopped in getcg with bad isamp=',I10)
stop
end if

! -----
! Record 10: iwatch
! -----
read(4,*) iwatch
write(6,450) iwatch
write(1,450) iwatch
450  *  FORMAT(/,' SWATCH tracking switch: iwatch=',I2,
*      ' (0=off, 1=each interaction, 2=each step)')

! -----
! Record 11: ibrdst,iprdst,ibrspl,nbrspl
! -----
read(4,*) ibrdst,iprdst,ibrspl,nbrspl

write(6,460) ibrdst,iprdst,ibrspl,nbrspl
write(1,460) ibrdst,iprdst,ibrspl,nbrspl
460  *  FORMAT(/,' IBRDST=',I2,/,' IPRDST=',I2,/,' IBRSPL=',I2,' (NBR SPL='
*,I5,')')

if (ibrspl .gt. 0) then
  if (nbrspl .gt. 0) then
    fbrspl = 1.0/float(nbrspl)
  else
    write(6,470) ibrspl,nbrspl
    write(1,470) ibrspl,nbrspl
470  *  FORMAT(/,' Stopped in Getcg with IBRSPL=',I5,' and NBR SPL=',
*      I5)
stop
end if
end if

! -----
! Run KEK version of PEGS5 before calling HATCH
! (method was developed by Y. Namito - 010306)
! -----
write(6,480)
write(1,480)
480  *  FORMAT(/,' PEGS5NB3-call comes next',/)

! =====
! call pegs5nb3
! =====
! -----

```



```

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

write(6,490)
490  FORMAT(/,' HATCH-call comes next',/)

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

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

! SJW 02-May-2002 replace reading of PRESTA switches with
! estepe and estepe2, and call to presta_inputs with calls
! to check_limits and rmsfit
! Set minimum (total) energy

ecutmn = 1.D10
do i = 1,nreg
  if (ecut(i).gt.0.0) ecutmn=min(ecutmn,ecut(i))
end do

ek0 = ekein                                ! Set maximum (kinetic) energy

!
!      =====
!      call presta_inputs(nreg,ecutmn,ek0)    ! Do PRESTA inputs/summary
!      =====

!
!      -----
!      Record 12: estepe,estepe2
!      -----
read(4,*) estepe, estepe2
write(6,500) estepe, estepe2
write(1,500) estepe, estepe2
500  FORMAT(/,1X,' ESTEPE at EKMAX: ',F10.0,' (estepe)',
*        /,1X,' ESTEPE at ECUT: ',F10.0,' (estepe2)')

!
!      -----
!      Print values used for efrac1 and efrac2
!      -----
write(6,*)
write(6,*) ' EFRACL=',efrac1
write(6,*) ' EFRACH=',efrach

!
!      =====
!      call check_limits(nreg,ecutmn,ek0)    ! Set energy step constants
!      =====

!
!      =====
!      call rmsfit                            ! read multiple scattering data
!      =====

!
!      -----
!      All of the input data should have been read in at this point,
!      but check to make sure that the incident kinetic energy is
!      below the limit set by PEGS (i.e., UE and UP) for all media.
!      -----
do j=1,nmed
  if (ekein+RM .gt. ue(j)) then
    write(6,*)
    *      'Stopped in SUBROUTINE getcg with ekein + RM > ue(j):'
    write(6,*) ' j = ',j
    write(6,*) ' ekein + RM = ',ekein+RM
    write(6,*) ' ue(j) = ',ue(j)
    write(1,*)
    *      'Stopped in SUBROUTINE getcg with ekein + RM > ue(j):'
    write(1,*) ' j = ',j
    write(1,*) ' ekein + RM = ',ekein+RM
    write(1,*) ' ue(j) = ',ue(j)
    stop
  end if
  if (ekein .gt. up(j)) then
    write(6,*)

```

```

*      'Stopped in SUBROUTINE getcg with ekein > up(j):'
      write(6,*) '   j = ',j
      write(6,*) '   ekein = ',ekein
      write(6,*) '   up(j) = ',up(j)
      write(1,*)
*      'Stopped in SUBROUTINE getcg with ekein > up(j):'
      write(1,*) '   j = ',j
      write(1,*) '   ekein = ',ekein
      write(1,*) '   up(j) = ',up(j)
      stop
      end if
    end do

!-----
! Print various data associated with each media (not region)
!-----
      write(6,510)
510    FORMAT(/,' Quantities associated with each MEDIA:')
      do j=1,nmed
        write(6,520) (media(i,j),i=1,24)
520    FORMAT(/,1X,24A1)
        write(6,530) rho(j),rlc(j)
530    FORMAT(5X,' rho=',G15.7,' g/cu.cm      rlc=',G15.7,' cm')
        write(6,540) ae(j),ue(j)
540    FORMAT(5X,' ae=',G15.7,' MeV      ue=',G15.7,' MeV')
        write(6,550) ap(j),up(j)
550    FORMAT(5X,' ap=',G15.7,' MeV      up=',G15.7,' MeV',/)
      end do

!-----
! Print media and cutoff energies assigned to each region
!-----
      if(moreOutput .eq.1) then
        do i=1,nreg
          if (med(i) .eq. 0) then
            write(6,560) i,ecut(i),pcut(i)
560          FORMAT(' medium(' ,I3,')=vacuum',18X,
*              'ecut=',G10.5,' MeV, pcut=',g10.5,' mev')
          else
            write(6,570) i,(media(ii,med(i)),ii=1,24),ecut(i),pcut(i)
570          FORMAT(' medium(' ,I3,')=',24A1,
*              'ecut=',G10.5,' MeV, pcut=',G10.5,' MeV')
!-----
! Print out energy information of K- and L-X-rays
!-----
            if (iedgfl(i) .ne. 0) then          ! Output X-ray energy
              ner = nne(med(i))
              do iiz=1,ner
                izn = zelem(med(i),iiz) ! Atomic number of this element
                write(6,580) izn
580              FORMAT('   X-ray information for Z=',I3)
                write(6,590) (ekx(ii,izn),ii=1,10)
590              FORMAT('   K-X-ray energy in keV',/,
*              4G15.5,/,4G15.5,/,2G15.5)
                write(6,600) (elx1(ii,izn),ii=1,8)
600              FORMAT('   L-1 X-ray in keV',/,4G15.5,/,4G15.5)
                write(6,610) (elx2(ii,izn),ii=1,5)
610              FORMAT('   L-2 X-ray in keV',/,5G15.5)
                write(6,620) (elx3(ii,izn),ii=1,7)
620              FORMAT('   L-3 X-ray in keV',/,4G15.5,/,3G15.5)
              end do
            end if
          end if
        end do
      end if

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

end

!-----last line of getcg.f-----

!-----ausgab.f-----
! Version: 030831-1300
! Reference: SLAC-265 (p.19-20, Appendix 2)
!-----
!23456789|123456789|123456789|123456789|123456789|123456789|123456789|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 'user_auxcommons/aux_h.f'   ! Auxiliary-code "header" file
include 'user_auxcommons/etaly1.f'  ! Auxiliary-code COMMONs
include 'user_auxcommons/lines.f'
include 'user_auxcommons/ntaly1.f'
include 'user_auxcommons/watch.f'

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

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

integer                               ! Arguments
* iarg

real*8                                 ! Local variables
* cmod,dcon,edepwt,encoaea,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 (irl.ne.iroid.and.iq(np).eq.0) then
        if((w(np).gt.0.0.and.irl.eq.2).or.(w(np).le.0.0.and.iroid.eq.
* 2)) 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

!-----last line of ausgab.f-----
!-----howfar.f-----
! Version: 040727-1300
! Reference: T. Torii and T. Sugita, "Development of PRESTA-CG
! Incorporating Combinatorial Geometry in EGS4/PRESTA", JNC TN1410 2002-201,
! Japan Nuclear Cycle Development Institute (2002).
! Improved version is provided by T. Sugita. 7/27/2004
! Reference: Provided by T. Sugita as improved Version
!-----
!23456789|123456789|123456789|123456789|123456789|123456789|123456789|12
!-----
! Required (geometry) subroutine for use with the EGS5 Code System
!-----
! This is a CG-HOWFAR.
!-----

subroutine howfar

implicit none

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

! include 'user_auxcommons/aux_h.f'
! include 'user_auxcommons/cg/tvalcg.f'
! include 'user_auxcommons/cg/zondta.f'
! include 'user_auxcommons/cg/rppdta.f'

```

```

include 'user_auxcommons/cg/sphdtac.f'
include 'user_auxcommons/cg/rccdta.f'
include 'user_auxcommons/cg/trcdta.f'
include 'user_auxcommons/cg/tordta.f'

real*8 atvaltmp,xidd,yidd,zidd                ! Local variables

real delhow,tval,tval0,tval10,tval100,tvalmn,udotau,udotav,
*   udotaw,xiss,xl,yiss,yl,ziss,zl

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

IRL=IR(NP)
IF (IRL.LT.1.OR.IRL.GE.IZONIN) THEN
  IDISC=1
  RETURN
END IF
TVAL=1.E+30
ITVALM=0
DO I=1,NBBODY(IRL)
  DO J=1,IRPPIN
    IF (ABS(NBZONE(I,IRL)).EQ.NBRPP(J)) THEN
      UDOTAU=U(NP)
      UDOTAV=V(NP)
      UDOTAW=W(NP)
      XL=X(NP)
      YL=Y(NP)
      ZL=Z(NP)
      CALL RPPCG1(J,XL,YL,ZL,UDOTAU,UDOTAV,UDOTAW)
    END IF
  end do
  DO J=1,ISPHIN
    IF (ABS(NBZONE(I,IRL)).EQ.NBSPH(J)) THEN
      UDOTAU=U(NP)
      UDOTAV=V(NP)
      UDOTAW=W(NP)
      XL=X(NP)
      YL=Y(NP)
      ZL=Z(NP)
      CALL SPHCG1(J,XL,YL,ZL,UDOTAU,UDOTAV,UDOTAW)
    END IF
  end do
  DO J=1,IRCCIN
    IF (ABS(NBZONE(I,IRL)).EQ.NBRCC(J)) THEN
      UDOTAU=U(NP)
      UDOTAV=V(NP)
      UDOTAW=W(NP)
      XL=X(NP)
      YL=Y(NP)
      ZL=Z(NP)
      CALL RCCG1(J,XL,YL,ZL,UDOTAU,UDOTAV,UDOTAW)
    END IF
  end do
  DO J=1,ITRCIN
    IF (ABS(NBZONE(I,IRL)).EQ.NBTRC(J)) THEN
      UDOTAU=U(NP)
      UDOTAV=V(NP)
      UDOTAW=W(NP)
      XL=X(NP)
      YL=Y(NP)
      ZL=Z(NP)
      CALL TRCCG1(J,XL,YL,ZL,UDOTAU,UDOTAV,UDOTAW)
    END IF
  end do
  DO J=1,ITORIN
    IF (ABS(NBZONE(I,IRL)).EQ.NBTOR(J)) THEN
      UDOTAU=U(NP)
      UDOTAV=V(NP)
      UDOTAW=W(NP)
      XL=X(NP)
      YL=Y(NP)
      ZL=Z(NP)
      CALL TORCG1(J,XL,YL,ZL,UDOTAU,UDOTAV,UDOTAW)
    END IF
  end do
end do
IRNEAR=IRL
IF (ITVALM.EQ.0) THEN
  TVALO=1.E-4
  XISS=X(NP)+TVALO*U(NP)
  YISS=Y(NP)+TVALO*V(NP)
  ZISS=Z(NP)+TVALO*W(NP)

```

```

2291 IF(X(NP).NE.XISS.OR.Y(NP).NE.YISS.OR.Z(NP).NE.ZISS) GO TO 2292
      TVALO=TVALO*10.
      XISS=X(NP)+TVALO*U(NP)
      YISS=Y(NP)+TVALO*V(NP)
      ZISS=Z(NP)+TVALO*W(NP)
      GO TO 2291
2292 CONTINUE
      XIDD=DBLE(X(NP))+DBLE(TVALO)*DBLE(U(NP))
      YIDD=DBLE(Y(NP))+DBLE(TVALO)*DBLE(V(NP))
      ZIDD=DBLE(Z(NP))+DBLE(TVALO)*DBLE(W(NP))
      CALL SRZONE(XIDD,YIDD,ZIDD,IRNEXT)
      IF (IRNEXT.NE.IRL) THEN
        TVAL=0.0
        IRNEAR=IRNEXT
      ELSE
        TVAL00=0.0
        TVAL10=10.0*TVALO
        IRL0LD=IRL
        IRLFG=0
2301 IF (IRLFG.EQ.1) GO TO 2302
        TVAL00=TVAL00+TVAL10
        IF (TVAL00.GT.1.0E+06) THEN
          WRITE(6,2310)IQ(NP),IR(NP),X(NP),Y(NP),Z(NP),U(NP),V(NP),
          * W(NP),TVAL00
2310 * FORMAT(' TVAL00 ERROR : IQ,IR,X,Y,Z,U,V,W,TVAL=',2I3,
          * 1P7E12.5)
          STOP
          END IF
          XIDD=DBLE(X(NP))+DBLE(TVAL00)*DBLE(U(NP))
          YIDD=DBLE(Y(NP))+DBLE(TVAL00)*DBLE(V(NP))
          ZIDD=DBLE(Z(NP))+DBLE(TVAL00)*DBLE(W(NP))
          CALL SRZOLD(XIDD,YIDD,ZIDD,IRL0LD,IRLFG)
          GO TO 2301
2302 CONTINUE
      TVAL=TVAL00
      DO J=1,10
        XIDD=DBLE(X(NP))+DBLE(TVAL00)*DBLE(U(NP))
        YIDD=DBLE(Y(NP))+DBLE(TVAL00)*DBLE(V(NP))
        ZIDD=DBLE(Z(NP))+DBLE(TVAL00)*DBLE(W(NP))
        CALL SRZONE(XIDD,YIDD,ZIDD,IRNEXT)
        IF (IRNEXT.NE.IRL0LD) THEN
          TVAL=TVAL00
          IRNEAR=IRNEXT
        END IF
        TVAL00=TVAL00-TVAL
      end do
      IF (IRL.EQ.IRNEAR) THEN
        WRITE(0,*) 'IRL,TVAL=',IRL,TVAL
      END IF
      END IF
      ELSE
        DO J=1,ITVALM-1
          DO I=J+1,ITVALM
            IF ((ATVAL(I).LT.ATVAL(J))) THEN
              ATVALTMP=ATVAL(I)
              ATVAL(I)=ATVAL(J)
              ATVAL(J)=ATVALTMP
            END IF
          end do
        end do
        ITVLF=0
        TVALMN=TVAL
        DO JJJ=1,ITVALM
          IF (TVALMN.GT.ATVAL(JJJ)) THEN
            TVALMN=ATVAL(JJJ)
          END IF
          DELHOW=1.E-4
          TVALO=ATVAL(JJJ)+DELHOW
          XISS=X(NP)+TVALO*U(NP)
          YISS=Y(NP)+TVALO*V(NP)
          ZISS=Z(NP)+TVALO*W(NP)
2361 IF(X(NP).NE.XISS.OR.Y(NP).NE.YISS.OR.Z(NP).NE.ZISS) GO TO 2362
          DELHOW=DELHOW*10.
          TVALO=ATVAL(JJJ)+DELHOW
          XISS=X(NP)+TVALO*U(NP)
          YISS=Y(NP)+TVALO*V(NP)
          ZISS=Z(NP)+TVALO*W(NP)
          GO TO 2361
2362 CONTINUE
      XIDD=DBLE(X(NP))+DBLE(TVALO)*DBLE(U(NP))
      YIDD=DBLE(Y(NP))+DBLE(TVALO)*DBLE(V(NP))
      ZIDD=DBLE(Z(NP))+DBLE(TVALO)*DBLE(W(NP))

```

```

CALL SRZONE(XIDD,YIDD,ZIDD,IRNEXT)
IF ((IRNEXT.NE.IRL.OR.ATVAL(JJJ).GE.1.).AND.TVAL.GT.
* ATVAL(JJJ)) THEN
    TVAL=ATVAL(JJJ)
    IRNEAR=IRNEXT
    ITVLF=1
    GOTO 2370
END IF
end do
2370 IF (ITVLF.EQ.0) THEN
    TVAL=1.E-4
    XISS=X(NP)+TVAL*U(NP)
    YISS=Y(NP)+TVAL*V(NP)
    ZISS=Z(NP)+TVAL*W(NP)
2381 IF (X(NP).NE.XISS.OR.Y(NP).NE.YISS.OR.Z(NP).NE.ZISS) GO TO 2382
    TVAL=TVAL*10.
    XISS=X(NP)+TVAL*U(NP)
    YISS=Y(NP)+TVAL*V(NP)
    ZISS=Z(NP)+TVAL*W(NP)
2382 GO TO 2381
CONTINUE
IF (TVALMN.GT.TVAL) THEN
    TVAL=TVALMN
ELSE
    TVAL=TVAL
END IF
END IF
END IF
IHITCG=0
IF (TVAL.LE.USTEP) THEN
    USTEP=TVAL
    IHITCG=1
END IF
IF (IHITCG.EQ.1) THEN
    IF (IRNEAR.EQ.0) THEN
        WRITE(6,2390) IQ(NP),IR(NP),X(NP),Y(NP),Z(NP),U(NP),V(NP),W(NP)
* ,TVAL
2390 FORMAT(' TVAL ERROR : IQ,IR,X,Y,Z,U,V,W,TVAL=' ,2I3,1P7E12.5)
        IDISC=1
        ITVERR=ITVERR+1
        IF (ITVERR.GE.100) THEN
            STOP
        END IF
        RETURN
    END IF
    IRNEW=IRNEAR
END IF
RETURN
END

```

```

!-----last line of subroutine howfar-----
!-----encoa.f-----

```

```

! Version: 030831-1300
! Reference: SLAC-265 (p.19-20, Appendix 2)

```

```

!23456789|123456789|123456789|123456789|123456789|123456789|123456789|12

```

```

!
! real function encoa(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 encoa(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=log(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=dlog(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-----

```