egs5 sample user code (uccg_phantom.f)
Dose distribution calculation inside phantom
(Draft, July 28, 2004)

Hideo Hirayama and Yoshihito Namito

KEK, High Energy Accelerator Research Organization
1-1, Oho, Tsukuba, Ibaraki, 305-0801 Japan

Contents

1. Combinatorial geometry (cg)

1.1. Body Definition

Following bodies are supported in PRESTA-CG*.

1. Rectangular Parallel-piped (RPP)
   Specify the maximum and minimum values of x-, y-, and z-coordinates that bound a rectuangular parallel-piped whose six sides are perpendicular to the coordinate axis.

2. Sphere (SPH)
   Specify the components of the radius vector **V** to the center of sphere and the radius R of the sphere.

3. Right Circular Cylinder (RCC)
   Specify the components of a radius vector **V** to the center of one base, the components of a vector **H** from the center of that base to the other base, and the radius of the cylinder.

4. Truncated Right Angle Cone (TRC)
   Specify the components of a radius vector **V** to the center of one base, the components of a vector **H** from the center of that base to the center of the other base, and the radii R1 and R2 of the lower and upper bases, respectively.

5. Torus (TOR)
   Specify the components of a radius vector **V** to the center of the torus, and the torus is configured parallel to one of the axis. R1 is the length between the center of torus and the center of tube, and R2 is the radius of the tube. Also, input the direction number of torus (n: x/y/z = 1/2/3). Furthermore, input starting angle $\theta 1$ and ending angle $\theta 2$ of the sector for the calculation of a part of torus. For the calculation of "complete" torus, set $\theta 1$=0, and $\theta 2$=$2pi$, respectively.

Table 1 Data requied to described each bosy type.

Body Type	Inp. #	Real Data defining Paticular Body					
RPP	#	Xmin	Xmax	Ymin	Ymax	Zmin	Zmax
SPH	#	Vx	Vy	Vz	R		
RCC	#	Vx	Vy	Vz	Hx	Hy	Hz
		R					
TRC	#	Vx	Vy	Vz	Hx	Hy	Hz
		R1	R2				
TOR	#	Vx	Vy	Vz	R1	R2	
		$\theta 1$	$\theta 2$	n			

1.2. Region Definition

The basic technique for description of the geometry consists of defining the location and shape of the various zones in term of the intersections and unions of the geometric bodies. A special operator notations involving the symbols (+), (−), and (OR) is used to describe the intersections and unions. These symbols are used by the program to construct information relating material descriptions to the body definitions.

*Please see Appendix A of *JNC TN1410 2002-001* by T. Torii and T. Sugita[1].

If a body appears in a region description with a (+) operator, it means that the region being described is wholly contained in the body. If a body appears in a region description with a (−) operator, it means that the region being described is wholly outside the body. If body appears with an (OR) operator, it means that the region being described includes all points in the body. OR may be considered as a union operator. In some instances, a region may be described in terms of subregion lumped together by (OR) statements. Subregions are formed as intersects and then the region is formed by union of these subregions. When (OR) operators are used there are always two or more of them, and they refer to all body numbers following them, either (+) or (−). That is, all body numbers between "OR's" or until the end of the region cards for that region are intersected together before OR's are performed.

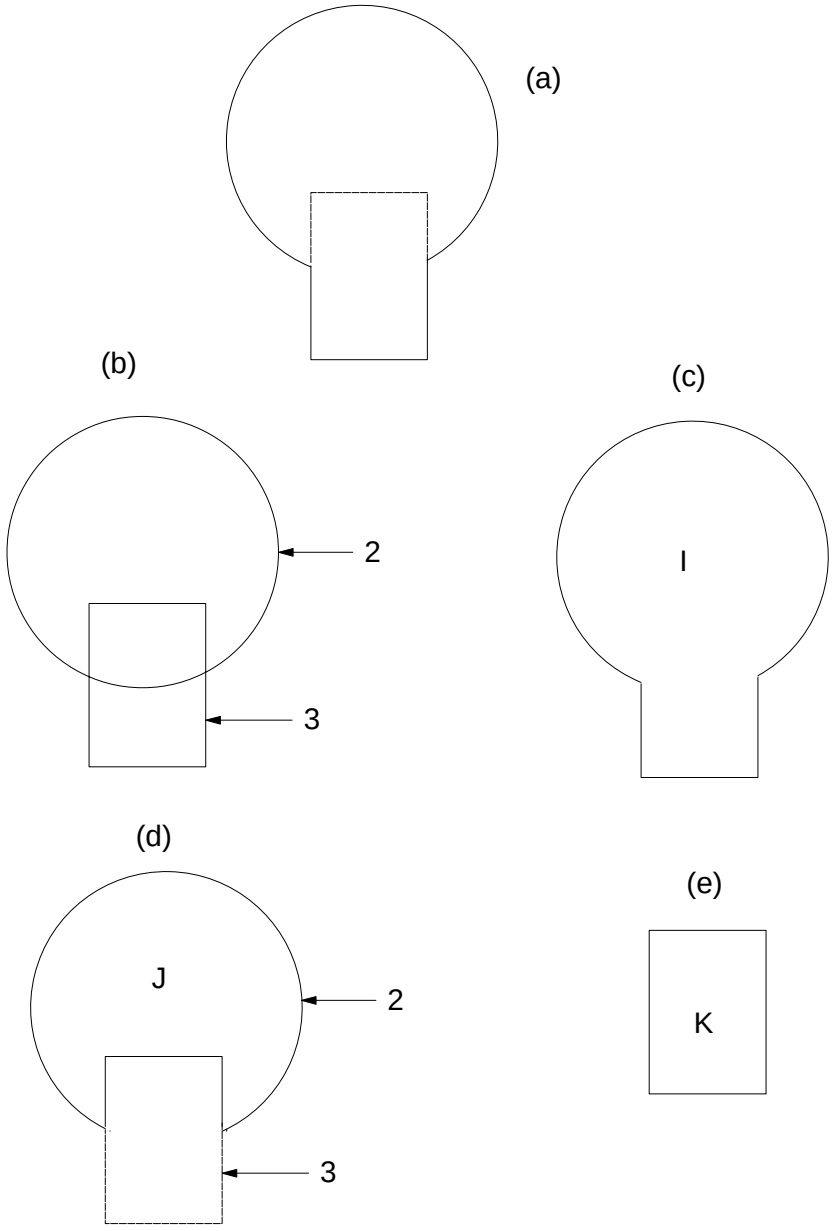

Figure 1: Examples of Combinatorial Geometry Method.

1.3. Example of Region Description

Consider an object composed of a sphere and a cylinder as shown in Fig. 1. To describe the object, one takes a spherical body (2) penetrated by a cylindrical body (3) (see Fig. 1). If the materials in the sphere and cylinder are the same, then they can be considered as one region, say region I (Fig. 1c). The description of region I would be

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

This means that a point is in region I if it is either body 2 or inside body 3.

If different material are used in the sphere and cylinder, then the sphere with a cylindrical hole in it would be given a different region number (say J) from one cylinder (K).

The description of region J would be (Fig. 1d):

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

This means that points in region J are all those points inside body 2 which are not inside body 3.

The description if region K is simply (Fig. 2e):

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

That is, all points in region K lie inside body 3.

Combination of more than two bodies and similar region descriptions could contain a long string of (+), (−), and (OR) operators. It is important however to remember that **every spatial point in the geometry must be located in one and only one region.**

As a more complicated example of the use of the (OR) operator, consider the system shown in Fig. 2 consisting of the shared region A and the unshared region B. These regions can be described by the two BOX's, bodies 1 and 3, and the RCC, body 2. The region description would be

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

and

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

Notice that OR operator refers to all following body numbers until the next OR operator is reached.

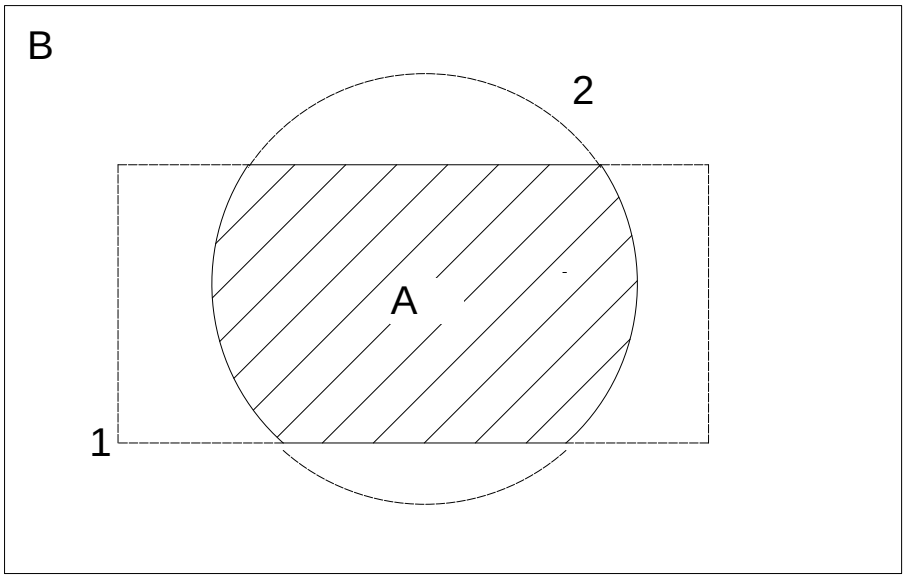

Figure 2: Use of OR operator.

2. Outlines of sample user code uccg_phantom.f

uccg_phantom.f is the egs5 user code to calculate the same problem with ucxyz_phantom.f using cg. Input data of cg are written at the top of the input data from unit 4.

2.1. Input data for cg

Each boxel region is defined by planes in ucxyz_phantom.f. On the other hand, in uccg_phantom.f, each region is defined by the combination of various rectangular parallel-pipes as shown in Fig. 3.

The input data for this geometry can be written as follows.

```
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
```

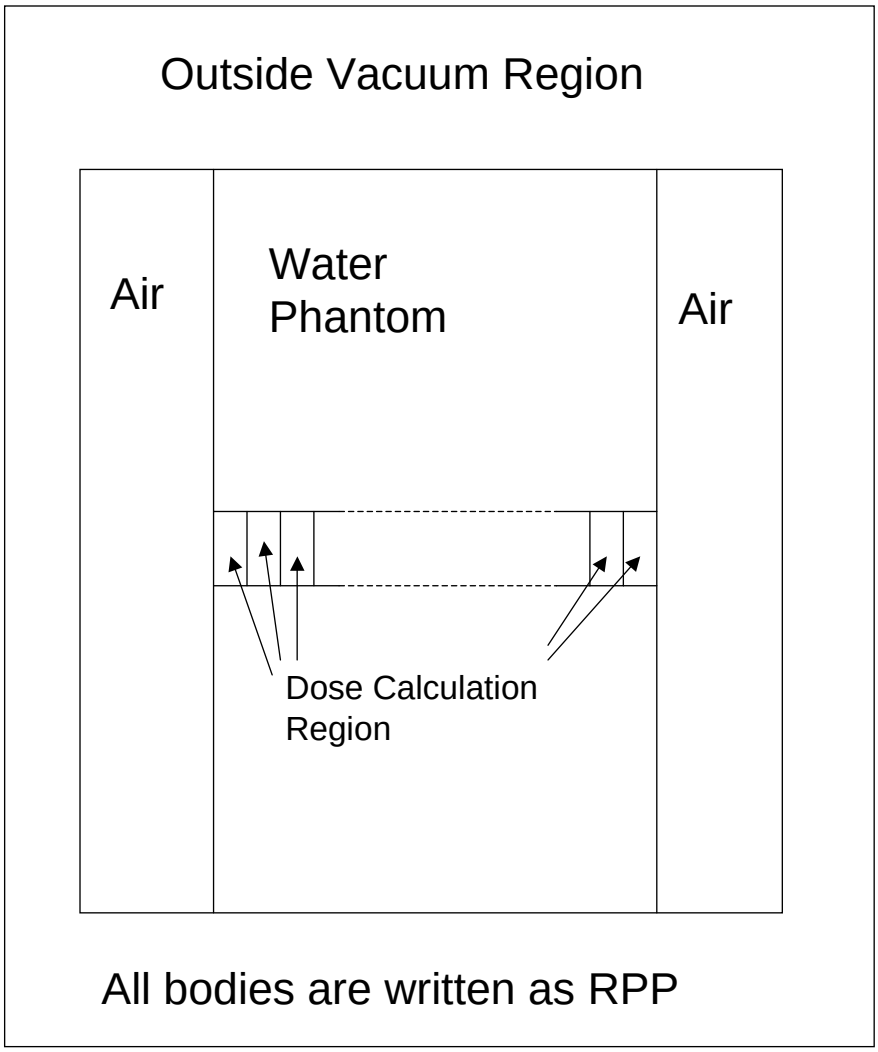

Figure 3: Geometry of uccg_phantom.f.

```
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. Geometry
   - Combination of rectangular parallel pipe (RPP)
   - Number of regions scoring dose is 20
   - phantom is modeled with water of 30cmx30cm area and 20cm depth
   - 5cm air region exits at before and after phantom
2. Source conditions
   - If `isemode=0`, source photon energy is sampled by using 100kV X-ray data (its spectrum information is read from xray.dat). If `isemode=1`, source photon energy is sampled by using data read from unit 4 at subroutine getcg.
   - Distance of point isotropic source (sposi) will be set from key-board.
   - Half-beam size at the phantom surface will be set both for x-direction (xhbeam) and y-direction (yhbeam) from key-board.
3. Calculation modes
   Following 2 modes are included. The mode is selected from key-board.
   - Trajectory mode. Make data to draw particle trajectories with the PICT32 system. (imode=0). Data will be written on egs5job.pic.
   - Dose calculation mode (imode=1). Result will be written on egs5job.out.
4. Results obtained
   (a) Trajectory display mode for CGview (imode=0)
   - Data of information of particle Trajectories (egs5job.pic)
   - Dose distributions and their uncertainties at central phantom (1cm $\times$ 1cm) area will be shown on console.
   - Back scattering factor at the phantom surface (1cm $\times$ 1cm area at the phantom center) will be shown on console. Exposure with or without the phantom is calculated from energy fluence and mass energy absorption coefficients of air.

   (b) Dose calculation mode (imode=1)
   - Information of material used
   - Material assignment to each region
   - Plane data defined
   - Comparison between sampled X-ray spectrum with data read from xray.dat
   - Number of histories and beam size at the phantom surface
   - Dose distributions and their uncertainties at central phantom (1cm $\times$ 1cm) area
   - Back scattering factor at the phantom surface (1cm $\times$ 1cm area at the phantom center)

3. Details of user code

3.1. Main program

3.1.1. Include lines and specification statements: egs5 is written in Fortran 77. The size of arguments is defined other files and included by using 'include line'. Various commons used inside egs5 are also included by the same way.

Include files related directory with egs5 are put on the sub-directory ('include' directory) of egs5 directory (currently egs5.0). Those for each user including geometry related are put on the subdirectory ('user_auxcommon' directory) of user directory (currently kek_sample). These files are linked by running egs5run script.

This is the most different feature with EGS4 at which the side of arguments can be modified inside an user code with Mortran macro. If it is necessary to modify the side of arguments used in

egs5, you must modify the related parameter in 'egs5.0/include/egs5_h.f'. The parameters related to each user are defined in 'kek_sampl/user_auxcommons/aux_h.f'.

First parts is include lines related egs5.

```
      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_elecin.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'` is always necessary. Other parts are only necessary when variables including at each common are used inside the main program.[†]

Neat is include lines not directly related to egas5 like geometry related.

```
!     ----------------------
!     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'
```

Next `etaly2.f` is the semi-egs5 `common` and put at the `egs5.0/auxcommons` directory. The last 7 `include` statements are related to cg.

`common` used inside the user code is defined next.

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

By `implicit none` at the top, it is required to declare all data by a type declaration statement.

[†]This is corresponding to COMIN macros in EGS4.

3.1.2. `open` statement: At the top of executable statement, it is necessary to open units used in the user code. Due to the new feature that pegs is called inside each user code, it must be careful to the unit number used. The unit number from 7 to 26 are used inside 'pegs' and close at the end of 'pegs'. These units, therefore, must be re-open after calling pegs. It is better not to use these unit in the user code. The unit used in the subroutine 'plotxyz' and 'geomout' used to keep and output trajectory information is changed from '9' to '39' for this reason.

```
!-----------------------------------------------------------------
!     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
!     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')
```

`Open` statement of `unit 2` is defined to read X-ray data from `xray.dat` file.

3.1.3. `call subroutine getcg`: Define the `npreci` which is used to define format for particle trajectories data and set 2 in this user code for CGview. After calling `subroutine region_init` which initialize some region, call `subroutine geomgt` to read cg input data and output cg information for CGview. `CSTA` and `CEND` are written before and after cg related data, respectively.

Next subroutine is called to clear various counter parameters.

`Subroutine getcg` which is called next is the new subroutine used to run pegs as a part of user code and call `subroutine hatch`.

In the `subroutine getcg`, material used, egs5 cut-off energy, various option flag, geometry related data etc. will be set by reading data from unit 4.

The material information of each region needed for CGview are output to `egs5job.pict`.

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

      itbody=0
      irppin=0
      isphin=0
      irccin=0
      itorin=0
      itrcin=0
      izonin=0
      izonad=0
      itverr=0
      igmmax=0
!      ifti = 90
      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
        write(39,1010)
1010    FORMAT('CEND')
      end if

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

```
      if (nreg.gt.mxreg) then
        write(6,1020) nreg,mxreg
1020    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
        write(39,1030)
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. Selection of calculation mode: As mentioned before, this user code has 2 calculation mode. The selection of mode is defined by the input data from key-board as follows.

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

3.1.5. Parameters of source particle: At first the distance between a point isotropic source and the phantom surface (`sposi`) is defined from key-board.

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

The number of dose calculation region is defined from key-board. The way of determining source energy is depending on the value of `isemode` as follows.

```
!-----------------------------
!     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 `imode=0`, a cumulative distribution function (cdf) calculated from a probability density function (pdf) which is read from `xray.dat`.

Minimum possible values Z-direction cosine is determined from the half beam width at the phantom surface both for x- and y-direction.

```
!-------------------------------------------------
!     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)
```

History number, `ncases`, is read from key-board. `ncases=0` means the end of execution.

3.1.6. How to increase the number of X-ray source: If you want use several type of X-rays and to select it from key-board, following modifications are necessary.

1. Change argument of `nofebin(1),deltae(1),sspec(1,201)` in the following `real*8` statement.

```
      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)
```

   '1' must be changed to the number of X-ray source and '201' to the maximum bin number within all sources used.

2. Add new data (number of bin `nofebin`, energy bin width (`deltae`:in MeV), X-ray number per bin (`sspec`)) to xray.dat.

3. Modify statements related to the selection of X-ray source. If 3 X-ray source (60kV, 80kV and 100kV) is used, this part is written as follows. Replace

```
\chgline
\begin{verbatim}
!-----------------------
!     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
```

   to

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

4. Modify write statement concerning the source (from 573 to 576 lines), from

```
        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')
```

to

```
        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')
```

5. Add `ixen` newly defined to `integer statement`.

3.1.7. Transport calculation: In this part, `subroutine shower` is called 'ncases' (history number). Before calling `shower`, various source parameters are sampled. In this used code, it is supposed that a point isotropic point source exits at `sposi` cm from the phantom surface. If `sposi` is larger than 5cm (air thickness in front of the phantom), starting source position at the surface of air region is determined considering the beam width at the phantom surface.

At each history, energy balance between the kinetic energy of source and absorbed energy in all region defined.

```
                                        ! -------------------------
      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.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
          if (isamp .eq. 0) then            ! Monoenergetic case
            ekin = ekein
            wtin = 1.0
          else if (isamp .eq. 1) then       ! Sample discrete energy from CDF
            call randomset(rnnow)
            i=0
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
            call randomset(rnnow)           ! interval and WEIGHT
            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 + epartd - 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 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+fexps
        fexps2s=fexps2s+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. Statistical uncertainty: The uncertainty of obtained, x, is estimated using the method used in MCNP in this user code.

- Assume that the calculation calls for N "incident" particle histories.
- Assume that x_i is the result at the i-th history.
- Calculate the mean value of x :

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

- Estimate the variance associated with the distribution of x_i:

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

- Estimate the variance associated with the distribution of $\overline{x}$:

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

- Report the statistical error as:

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

3.1.9. Output of results: Obtained results from `ncases` histories are analyzed and outputted in this part. In the dose calculation mode, the comparisons between sampled source spectrum and original data are printed.

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

The average absorbed dose and its uncertainty at each voxel are calculated. The depth distribution at the central area of the phantom and back scattering factor obtained from exposure at the phantom surface with and without phantom are printed.

3.2. Subroutine getcg

`Subroutine getcg` is used to read material data used, its density, egs5 cut-off energy, various optional flag applied to each region etc. for cg geometry problem and call `subroutine hatch`.

The data read from unit 4 are as follows.

1. Record 1 : Title (within 80 characters)
2. Record 2 : Number of media in problem (nmed)
3. Record 3 : Media names (j=1,24, i=1,nmed lines)

4. Record 4 : 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).
   If medium not 0, following option is set to the regions above. (0: off, 1:on)

5. Record 4a : irlinl,irlinu,,medtmp, rhotmp, ecutin, pcutin

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 : Incident X,Y,Z coordinates (cm) (xin, yin, zin)

7. Record 6 : Incident region

8. Record 7 : Incident direction cosines (uin,vin,win) If uin=vin=win=0, it means isotropic source.

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

10. Record 9 : Number of cases (ncases).

11. Record 10 : Kinetic energy (MeV), charge of incident beam, and sampling switch. If isamp=0, a monoenergetic beam (ekein) will be used. Otherwise, a spectrum input must follow (Records 14a through 14b), which will be sampled from discrete energy (isamp=1), directly (isamp=2) or uniformly over the energy range (isamp=3) with weighting factor.

12. Record 10a : Only required when $isamp > 1$ (see above).

13. Record 10b : Only required when usamp¿0(see above). ebin(i) is the 'top-edge' of each energy bin (MeV) and epdf(i) is the corresponding probability for the bin. For example, a cross section (mb) can be used for epdf (but do not divide it by dE). The last card is a delimiter and should be blank (or contain 0.0). The i-subscript runs from 1 to nebin (nebin calculated after the delimiter).

14. Record 11 : Switch for tracking events with swatch: (0=No, 1=each interaction, 2=each step)

15. Record 12 : Switches for bremsstrahlung and pair production ANGLE SAMPLING, and brems-strahlung SPLITTING:

ibrdst=0	No (use default: theta=m/E)
ibrdst=1	Yes (recommended)
iprdst=0	No (use default: theta=m/E)
iprdst=1	Yes (low-order distribution)
iprdst=2	Yes (recommended)
ibrspl=0	No splitting
ibrspl=1	Apply splitting (nbrspl=splitting factor)

16. Record 18 : Parameters used for charged particle transport (estepe,estepe2).

3.3. Subroutine ausgab

`Subroutine ausgab` is a subroutine to score variables that user want to score.

Include lines and specification statements are written at first by the same way used at the main program.

After the treatment related `iwatch` option, value of the stack number (np) is checked not to exceed the pre-set maximum value.

When $iarg < 5$, absorbed energy at the region nreg (outside the system) and other regions are summed separately to check energy balance at each history. If region is from 2 to nreg-3, score absorbed energy by setting a detector number to `idet=irl-1`.

If photon crosses the phantom surface at the central region, energy absorption of air is calculated from energy fluence of photon and mass attenuation coefficient of air. Energy absorption of air without phantom is corresponding those by photons never scattered backward. For this purpose, `latch(np))` is set to 1 if `w(np) < 0`.

If a trajectory display mode is selected, `subroutine plotxyz` which is record and output trajectory related information is called.

```
!     ------------------------------------------------------------------
!     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=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
```

3.4. Subroutine howfar

At subroutine howfar, a distance to the boundary of region is checked. If the distance to the boundary is shorter than the distance to the next point, the distance to the next point is replaced with the distance to the boundary and new region irnew is set to the region number to which particle will enter.

If idisc is set to 1 by user, the treatment to stop following will be done in this subroutine.

Calculation to a distance to the boundary is done by using the various subroutines related cg in uccg_phantom.f.

4. Comparison of speed between ucxyz_phantom.f and と uccg_phantom.f

Cg geometry is suitable to treat a complex geometry than the cylinder-plane geometry etc. On the other hanad, cg needs more cpu time. For example, uccg_phantom.f needs 2.5 times longer cpu time than ucxyz_phantom.f for the same problem.[‡]

[‡]Drastic sppedup for CG almost factor 5 in this case was provided by T. Sugita.

5. Exercise problems

5.1. Problem 1 : Change source energy

Change the source to 0.662 MeV photons from ^{137}Cs.

5.2. Problem 2 : Change source energy

Change source energy to 1.173 and 1.332 MeV photons from ^{60}Co.

5.3. Problem 3 : Change to lung model

Set surface 3 cm of phantom as the normal tissue (water), 3 to 13 cm as the lung (water with 0.3 g cm^{-3}) and 13-16cm as the normal tissue.
Source is the X-ray read from `xray.dat)`.

5.4. Problem 4 : Lung with tumor

Set tumor region at 3 to 5cm from the lung surface as the normal tissue.

5.5. Problem 5 : Inset iron inside phantom

Replace 5 to 6 cm region of the phantom with iron.

5.6. Other problems

In addition above, following problems are also useful as exercises.

- Use other X-ray sources
- Change incident particle to an electron
- Change thickness of iron
- Calculate for limited area of tumor

5.7. Answer for exercise

5.8. Problem 1

1. Change source energy selection mode isemode to 1 from 0.
2. Change value of ekinin at 68 lines in uccg_phantom.data to 0.667 from 1.332.
3. Save uccg_phantom.data as the different name and assign as the file name for unit 4.

5.9. Problem 2

1. Under isemode=1, change isamp to 1 from 0 at 68 lines in uccg_phantom.data.
2. Add following data after 68 lines.

```
1.173,    1.0              discrete energy 1
1.332,    1.0,             discrete energy 2
0.0,      0.0,             end of set energy
```

3. Save uccg_phantom.data as the different name and assign as the file name for unit 4.

5.10. Problem 3

1. Change source energy selection mode isemode to 0 from 1.
2. Change cg input data as follows.

```
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
```

3. Change following material assignment data (58 lines)

```
    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
```

to

```
    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
```

4. Save uccg_phantom.data as the different name and assign as the file name for unit 4.

5. Kei-in 16 as the number of region to calculate dose.

5.11. Problem 4

1. Change cg input data as follows.

```
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. Change following material assignment data (58 lines)

```
    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
```

to

```
    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       tussue
    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
```

3. Save uccg_phantom.data as the different name and assign as the file name for unit 4.

4. Key-in 16 as the number of region to calculate dose.

5.12. Ploblem 5

1. Add following data to uccg_phantom.inp and save as the different name.

```
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. Change number of material nmed at 53 line of uccg_phantom.data to '3' from '2'. Add following data after 4 lines.

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

3. Change data following data (58 to 59 lines)

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

to

```
     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
```

4. Save uccg_phantom.data as the different name and assign as the file name for unit 4.

5. Kei-in 20 as the number of region to calculate dose.

References

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

Appendix 1 Full listings of uccg_phantom.f

```
!*********************************************************************
!*****************************  KEK, High Energy Accelerator Research  *
!*****************************  Organization                           *
!*** u c c g _ p h a n t o m *                                         *
!*****************************     EGS5.0 USER CODE - 16 JUL 2004/1300 *
!*********************************************************************
* This is a general User Code based on the cg geometry scheme.        *
!*********************************************************************
!                                                                     *
!  PROGRAMMERS:  H. Hirayama                                          *
!                Radiation Science Center                             *
!                Applied Science Laboratory                           *
!                KEK, High Energy Accelerator Research Organization   *
!                1-1, Oho, Tsukuba, Ibaraki, 305-0801                 *
!                Japan                                                *
!                                                                     *
!                E-mail:      hideo.hirayama@kek.jp                   *
!                Telephone:   +81-29-864-5489                         *
!                Fax:         +81-29-864-1993                         *
!                Based on ucrtz_sampl4 by Nelson and James.           *
!                                                                     *
!*********************************************************************
!*********************************************************************
! The uccg_phantom.f User Code requires a data-input file             *
! (e.g., uccg_phantom.data) that is read by subroutine getcg (with    *
! instructions in its header).  The following shows the geometry for  *
! uccg_phantom.data.                                                  *
! Input data for CG geometry must be written at the top of data-input *
! file. This user code corresponds to ucphantomcgp.mor for egs4.      *
!*********************************************************************
!                                                                     *
!             ----------------------------------                      *
!             cg Geometry (uccg_phantom example)                      *
!             ----------------------------------                      *
!                                                                     *
!                                                                     *
!                X                                                    *
!                ^                                                    *
!                |                                                    *
!           +----+----+----+----   -+---+----+----+--- 20.0           *
!           |                                        |                *
!           |     Outer vacuum region                |                *
!           +    +----+----+----   -+---+----+    +--- 15.0           *
!           |    |    |                |    |    |                   *
!           |    +    +  Water (H2O)    |    |    |                   *
!           |    |    |                |    |    |                   *
!           +    +    +---+---+-   -+---+    +    +--   0.5          *
!           |    |    |   |   |      |   |    |    |                   *
!           |    |Vac.|H2O|H2O|      |H2O|Vac.|    |                   *
! X-ray     |    |    |   |   |      |   |    |    |                   *
!      ==========>+---+---+---+-   -+---+----+----+--> Z               *
! photons -20  -10.0  0  1.0 2.0  19.0 20.0     40.0                   *
!                                        30.0                         *
!                                                                     *
!                                                                     *
!                                                                     *
!*********************************************************************
!23456789|123456789|123456789|123456789|123456789|123456789|123456789|12
!
!---------------------------------------------------------------------
!------------------------------ main code ----------------------------
!---------------------------------------------------------------------

      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_elecin.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'

!     ----------------------
!     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
      itrcin=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
        write(39,1010)
1010    FORMAT('CEND')
      end if

!-------------------------------
!     Get nreg from cg input data
!-------------------------------
      nreg=izonin
      if (nreg.gt.mxreg) then
        write(6,1020) nreg,mxreg
1020    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
        write(39,1030)
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
        write(1,100)
100     FORMAT(' Quantities associated with each media:')
        do j=1,nmed
          write(1,110) (media(i,j),i=1,24)
```

```
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

!------------------------
!     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
                                              ! -------------------------
        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.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
          if (isamp .eq. 0) then            ! Monoenergetic case
            ekin = ekein
            wtin = 1.0
          else if (isamp .eq. 1) then       ! Sample discrete energy from CDF
            call randomset(rnnow)
            i=0
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
            call randomset(rnnow)           ! interval and WEIGHT
            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 + epartd - 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 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+fexps
        fexps2s=fexps2s+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
                                                  ! ------------------------

      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
      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(6,420) 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,
           incohrsw,            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  ixx,jxx             Starting random number seeding.
  ---------                     If ixx = 0, ixx is set to 123457.
                                If jxx = 0, jxx is set to 654321.
  ---------
  Record 9  ncases              Number of cases.
  ---------
  Record 10 ekein,iqin,isamp    Kinetic energy (MeV), charge of inci-
  ---------                     dent beam, and sampling switch.  If
                                isamp=0, a monoenergetic beam (ekein)
                                will be used.  Otherwise, a spectrum
                                input must follow (Records 10a through
                                10b), which will be sampled from discrete
                                energy (isamp=1), directly (isamp=2) or
                                uniformly over the energy range (isamp=3)
                                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 usamp>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 (NBRSPL=splitting factor)
! --------
! Record 13 estepe,estepe2
! ---------
! ------------------------------------------------------------------------

      subroutine getcg(nreg)

      implicit none

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

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

      include 'pegscommons/mscom.f'                          ! PEGS common

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

      include 'user_auxcommons/edata.f'
      include 'user_auxcommons/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

      write(6,100)
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
        write(6,130) nmed
130     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, meptmp, 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,
!            incohrsw,iprofrsw,impacrsw
! ---------------------------------------
        read(4,200) ipeangsw,iedgesw,iraysw,ipolarsw,incohrsw,
     * iprofrsw,impacrsw
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,incohrsw,iprofrsw,impacrsw
        write(1,230) ipolarsw,incohrsw,iprofrsw,impacrsw
230     FORMAT(11X,' lpolar=',I3,3X,'incohr=',I3,3X,'iprofr=',I3,
     *       3X,'impacr=',I3)
      else
        write(6,240) irlin
        write(1,240) irlin
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) = incohrsw
          iprofr(irlin) = iprofrsw
          impacr(irlin) = impacrsw
        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
```

```
      write(1,320) ncases
320   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,' (NBRSPL='
     *,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 NBRSPL=',
     *    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 efracl and efrach
! ------------------------------------
      write(6,*)
      write(6,*) '  EFRACL=',efracl
      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(6,520) (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,encoea,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.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=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: 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,tval00,tvalmn,udotau,udotav,
     *     udotaw,xiss,xl,yiss,yl,ziss,zl

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

      IRL=IR(NP)
      IF (IRL.LT.1.OR.IRL.GE.IZONIN) THEN
        IDISC=1
        RETURN
      END IF
      TVAL=1.E+30
      ITVALM=0
      DO I=1,NBBODY(IRL)
        DO J=1,IRPPIN
          IF (ABS(NBZONE(I,IRL)).EQ.NBRPP(J)) THEN
            UDOTAU=U(NP)
            UDOTAV=V(NP)
            UDOTAW=W(NP)
            XL=X(NP)
            YL=Y(NP)
            ZL=Z(NP)
            CALL RPPCG1(J,XL,YL,ZL,UDOTAU,UDOTAV,UDOTAW)
          END IF
        end do
        DO J=1,ISPHIN
          IF (ABS(NBZONE(I,IRL)).EQ.NBSPH(J)) THEN
            UDOTAU=U(NP)
            UDOTAV=V(NP)
            UDOTAW=W(NP)
            XL=X(NP)
            YL=Y(NP)
            ZL=Z(NP)
            CALL SPHCG1(J,XL,YL,ZL,UDOTAU,UDOTAV,UDOTAW)
          END IF
        end do
        DO J=1,IRCCIN
          IF (ABS(NBZONE(I,IRL)).EQ.NBRCC(J)) THEN
            UDOTAU=U(NP)
            UDOTAV=V(NP)
            UDOTAW=W(NP)
            XL=X(NP)
            YL=Y(NP)
            ZL=Z(NP)
            CALL RCCCG1(J,XL,YL,ZL,UDOTAU,UDOTAV,UDOTAW)
          END IF
        end do
        DO J=1,ITRCIN
          IF (ABS(NBZONE(I,IRL)).EQ.NBTRC(J)) THEN
            UDOTAU=U(NP)
            UDOTAV=V(NP)
            UDOTAW=W(NP)
            XL=X(NP)
            YL=Y(NP)
            ZL=Z(NP)
            CALL TRCCG1(J,XL,YL,ZL,UDOTAU,UDOTAV,UDOTAW)
          END IF
        end do
        DO J=1,ITORIN
          IF (ABS(NBZONE(I,IRL)).EQ.NBTOR(J)) THEN
            UDOTAU=U(NP)
            UDOTAV=V(NP)
            UDOTAW=W(NP)
            XL=X(NP)
            YL=Y(NP)
            ZL=Z(NP)
            CALL TORCG1(J,XL,YL,ZL,UDOTAU,UDOTAV,UDOTAW)
          END IF
        end do
      end do
      IRNEAR=IRL
      IF (ITVALM.EQ.0) THEN
        TVAL0=1.E-4
        XISS=X(NP)+TVAL0*U(NP)
        YISS=Y(NP)+TVAL0*V(NP)
        ZISS=Z(NP)+TVAL0*W(NP)
2291    IF(X(NP).NE.XISS.OR.Y(NP).NE.YISS.OR.Z(NP).NE.ZISS) GO TO 2292
          TVAL0=TVAL0*10.
          XISS=X(NP)+TVAL0*U(NP)
          YISS=Y(NP)+TVAL0*V(NP)
```

```
          ZISS=Z(NP)+TVAL0*W(NP)
        GO TO 2291
2292    CONTINUE
        XIDD=DBLE(X(NP))+DBLE(TVAL0)*DBLE(U(NP))
        YIDD=DBLE(Y(NP))+DBLE(TVAL0)*DBLE(V(NP))
        ZIDD=DBLE(Z(NP))+DBLE(TVAL0)*DBLE(W(NP))
        CALL SRZONE(XIDD,YIDD,ZIDD,IRNEXT)
        IF (IRNEXT.NE.IRL) THEN
          TVAL=0.0
          IRNEAR=IRNEXT
        ELSE
          TVAL00=0.0
          TVAL10=10.0*TVAL0
          IRLOLD=IRL
          IRLFG=0
2301      IF (IRLFG.EQ.1) GO TO 2302
            TVAL00=TVAL00+TVAL10
            IF (TVAL00.GT.1.0E+06) THEN
              WRITE(6,2310)IQ(NP),IR(NP),X(NP),Y(NP),Z(NP), U(NP),V(NP),
     *        W(NP),TVAL00
2310          FORMAT(' TVAL00 ERROR : IQ,IR,X,Y,Z,U,V,W,TVAL=', 2I3,
     *        1P7E12.5)
              STOP
            END IF
            XIDD=DBLE(X(NP))+DBLE(TVAL00)*DBLE(U(NP))
            YIDD=DBLE(Y(NP))+DBLE(TVAL00)*DBLE(V(NP))
            ZIDD=DBLE(Z(NP))+DBLE(TVAL00)*DBLE(W(NP))
            CALL SRZOLD(XIDD,YIDD,ZIDD,IRLOLD,IRLFG)
          GO TO 2301
2302      CONTINUE
          TVAL=TVAL00
          DO J=1,10
            XIDD=DBLE(X(NP))+DBLE(TVAL00)*DBLE(U(NP))
            YIDD=DBLE(Y(NP))+DBLE(TVAL00)*DBLE(V(NP))
            ZIDD=DBLE(Z(NP))+DBLE(TVAL00)*DBLE(W(NP))
            CALL SRZONE(XIDD,YIDD,ZIDD,IRNEXT)
            IF (IRNEXT.NE.IRLOLD) THEN
              TVAL=TVAL00
              IRNEAR=IRNEXT
            END IF
            TVAL00=TVAL00-TVAL0
          end do
          IF (IRL.EQ.IRNEAR) THEN
            WRITE(0,*) 'IRL,TVAL=',IRL,TVAL
          END IF
        END IF
      ELSE
        DO J=1,ITVALM-1
          DO I=J+1,ITVALM
            IF ((ATVAL(I).LT.ATVAL(J))) THEN
              ATVALTMP=ATVAL(I)
              ATVAL(I)=ATVAL(J)
              ATVAL(J)=ATVALTMP
            END IF
          end do
        end do
        ITVLFG=0
        TVALMN=TVAL
        DO JJJ=1,ITVALM
          IF (TVALMN.GT.ATVAL(JJJ)) THEN
            TVALMN=ATVAL(JJJ)
          END IF
          DELHOW=1.E-4
          TVAL0=ATVAL(JJJ)+DELHOW
          XISS=X(NP)+TVAL0*U(NP)
          YISS=Y(NP)+TVAL0*V(NP)
          ZISS=Z(NP)+TVAL0*W(NP)
2361      IF(X(NP).NE.XISS.OR.Y(NP).NE.YISS.OR.Z(NP).NE.ZISS) GO TO 2362
            DELHOW=DELHOW*10.
            TVAL0=ATVAL(JJJ)+DELHOW
            XISS=X(NP)+TVAL0*U(NP)
            YISS=Y(NP)+TVAL0*V(NP)
            ZISS=Z(NP)+TVAL0*W(NP)
          GO TO 2361
2362      CONTINUE
          XIDD=DBLE(X(NP))+DBLE(TVAL0)*DBLE(U(NP))
          YIDD=DBLE(Y(NP))+DBLE(TVAL0)*DBLE(V(NP))
          ZIDD=DBLE(Z(NP))+DBLE(TVAL0)*DBLE(W(NP))
          CALL SRZONE(XIDD,YIDD,ZIDD,IRNEXT)
          IF ((IRNEXT.NE.IRL.OR.ATVAL(JJJ).GE.1.).AND.TVAL.GT.
     *    ATVAL(JJJ)) THEN
            TVAL=ATVAL(JJJ)
```

```
            IRNEAR=IRNEXT
            ITVLFG=1
            GOTO 2370
          END IF
        end do
2370    IF (ITVLFG.EQ.0) THEN
          TVAL0=1.E-4
          XISS=X(NP)+TVAL0*U(NP)
          YISS=Y(NP)+TVAL0*V(NP)
          ZISS=Z(NP)+TVAL0*W(NP)
2381      IF(X(NP).NE.XISS.OR.Y(NP).NE.YISS.OR.Z(NP).NE.ZISS) GO TO 2382
            TVAL0=TVAL0*10.
            XISS=X(NP)+TVAL0*U(NP)
            YISS=Y(NP)+TVAL0*V(NP)
            ZISS=Z(NP)+TVAL0*W(NP)
          GO TO 2381
2382      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(6,2390)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----------------------
!------------------------------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=alog(enmu(i+1))
          enm0=alog(enmu(i))
          hnu1=alog(hnu(i+1))
          hnu0=alog(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=alog(enmu(i+1))
          enm0=alog(enmu(i))
          hnu1=alog(hnu(i+1))
```

```
          hnu0=alog(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-------------------------
!************************************************************************
!****************************  KEK, High Energy Accelerator Research  *
!****************************  Organization                           *
!*** u c c g _ p h a n t o m *                                         *
!****************************     EGS5.0 USER CODE - 28 JUL 2004/1300 *
!************************************************************************
!* This is a general User Code based on the cg geometry scheme.         *
!************************************************************************
!                                                                       *
!  PROGRAMMERS:  H. Hirayama                                            *
!                Radiation Science Center                               *
!                Applied Science Laboratory                             *
!                KEK, High Energy Accelerator Research Organization     *
!                1-1, Oho, Tsukuba, Ibaraki, 305-0801                   *
!                Japan                                                  *
!                                                                       *
!                E-mail:      hideo.hirayama@kek.jp                     *
!                Telephone:   +81-29-864-5489                           *
!                Fax:         +81-29-864-1993                           *
!                Based on ucrtz_sampl4 by Nelson and James.             *
!                                                                       *
!************************************************************************
!************************************************************************
! The uccg_phantom.f User Code requires a data-input file               *
! (e.g., uccg_phantom.data) that is read by subroutine getcg (with      *
! instructions in its header).  The following shows the geometry for    *
! uccg_phantom.data.                                                    *
! Input data for CG geometry must be written at the top of data-input   *
! file. This user code corresponds to ucphantomcgp.mor for egs4.        *
!************************************************************************
!                                                                       *
!              ----------------------------------                       *
!              cg Geometry (uccg_phantom example)                       *
!              ----------------------------------                       *
!                                                                       *
!                                                                       *
!                 X                                                     *
!                 ^                                                     *
!                 |                                                     *
!            +----+----+----+----   -+---+----+----+--- 20.0           *
!            |                                        |                 *
!            |     Outer vacuum region                |                 *
!            +    +----+----+----   -+---+----+    +--- 15.0           *
!            |    |    |                   |    |    |                 *
!            |    +    +  Water (H2O)      |    |    |                 *
!            |    |    |                   |    |    |                 *
!            +    |    +---+---+-   -+---+    +    +--   0.5          *
!            |    |    |   |   |     |   |    |    |                   *
!            |    |Vac.|H2O|H2O|     |H2O|Vac.|    |                   *
! X-ray      |    |    |   |   |     |   |    |    |                   *
!       ==========>+---+---+---+-   -+---+----+----+--> Z              *
! photons -20  -10.0  0  1.0 2.0  19.0 20.0     40.0                    *
!                                      30.0                             *
!                                                                       *
!                                                                       *
!                                                                       *
!************************************************************************
!23456789|123456789|123456789|123456789|123456789|123456789|123456789|12
!
!-----------------------------------------------------------------------
```

```
!------------------------------ main code ------------------------------
!-----------------------------------------------------------------------

      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_elecin.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'

!     ----------------------
!     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
      itrcin=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
        write(39,1010)
1010    FORMAT('CEND')
      end if

!-------------------------------
!     Get nreg from cg input data
!-------------------------------
      nreg=izonin
      if (nreg.gt.mxreg) then
        write(6,1020) nreg,mxreg
1020    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
        write(39,1030)
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
        write(1,100)
100     FORMAT(' Quantities associated with each media:')
        do j=1,nmed
          write(1,110) (media(i,j),i=1,24)
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

!------------------------
!     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
                                              ! -------------------------
        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.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
          if (isamp .eq. 0) then            ! Monoenergetic case
            ekin = ekein
            wtin = 1.0
          else if (isamp .eq. 1) then       ! Sample discrete energy from CDF
```

```
            call randomset(rnnow)
            i=0
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
            call randomset(rnnow)          ! interval and WEIGHT
            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 + epartd - 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 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+fexps
        fexps2s=fexps2s+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
                                              ! ------------------------

      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
       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(6,420) 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,
!           incohrsw,            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  ixx,jxx             Starting random number seeding.
!  ---------                     If ixx = 0, ixx is set to 123457.
!                                If jxx = 0, jxx is set to 654321.
!  ---------
!  Record 9  ncases              Number of cases.
!  ---------
!  Record 10 ekein,iqin,isamp    Kinetic energy (MeV), charge of inci-
!  ---------                     dent beam, and sampling switch.  If
!                                isamp=0, a monoenergetic beam (ekein)
!                                will be used.  Otherwise, a spectrum
!                                input must follow (Records 10a through
!                                10b), which will be sampled from discrete
!                                energy (isamp=1), directly (isamp=2) or
!                                uniformly over the energy range (isamp=3)
!                                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 usamp>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 (NBRSPL=splitting factor)
! --------
! Record 13 estepe,estepe2
! ---------
! ------------------------------------------------------------------------

      subroutine getcg(nreg)

      implicit none

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

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

      include 'pegscommons/mscom.f'                          ! PEGS common

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

      include 'user_auxcommons/edata.f'
      include 'user_auxcommons/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

      write(6,100)
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
        write(6,130) nmed
130     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, meptmp, 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,
!            incohrsw,iprofrsw,impacrsw
! ---------------------------------------
        read(4,200) ipeangsw,iedgesw,iraysw,ipolarsw,incohrsw,
     * iprofrsw,impacrsw
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,incohrsw,iprofrsw,impacrsw
        write(1,230) ipolarsw,incohrsw,iprofrsw,impacrsw
230     FORMAT(11X,' lpolar=',I3,3X,'incohr=',I3,3X,'iprofr=',I3,
     *       3X,'impacr=',I3)
       else
        write(6,240) irlin
        write(1,240) irlin
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) = incohrsw
           iprofr(irlin) = iprofrsw
           impacr(irlin) = impacrsw
         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
      write(1,320) ncases
320   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,' (NBRSPL='
     *,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 NBRSPL=',
     *    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 efracl and efrach
! ------------------------------------
       write(6,*)
       write(6,*) '  EFRACL=',efracl
       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(6,520) (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,encoea,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.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=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
! Incorprating 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,tval00,tvalmn,udotau,udotav,
     *     udotaw,xiss,xl,yiss,yl,ziss,zl

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

      IRL=IR(NP)
      IF (IRL.LT.1.OR.IRL.GE.IZONIN) THEN
        IDISC=1
        RETURN
      END IF
      TVAL=1.E+30
      ITVALM=0
      DO I=1,NBBODY(IRL)
        DO J=1,IRPPIN
          IF (ABS(NBZONE(I,IRL)).EQ.NBRPP(J)) THEN
            UDOTAU=U(NP)
            UDOTAV=V(NP)
            UDOTAW=W(NP)
            XL=X(NP)
            YL=Y(NP)
            ZL=Z(NP)
            CALL RPPCG1(J,XL,YL,ZL,UDOTAU,UDOTAV,UDOTAW)
          END IF
        end do
        DO J=1,ISPHIN
          IF (ABS(NBZONE(I,IRL)).EQ.NBSPH(J)) THEN
            UDOTAU=U(NP)
            UDOTAV=V(NP)
            UDOTAW=W(NP)
            XL=X(NP)
            YL=Y(NP)
            ZL=Z(NP)
            CALL SPHCG1(J,XL,YL,ZL,UDOTAU,UDOTAV,UDOTAW)
          END IF
        end do
        DO J=1,IRCCIN
          IF (ABS(NBZONE(I,IRL)).EQ.NBRCC(J)) THEN
            UDOTAU=U(NP)
            UDOTAV=V(NP)
            UDOTAW=W(NP)
            XL=X(NP)
            YL=Y(NP)
            ZL=Z(NP)
            CALL RCCCG1(J,XL,YL,ZL,UDOTAU,UDOTAV,UDOTAW)
          END IF
        end do
        DO J=1,ITRCIN
          IF (ABS(NBZONE(I,IRL)).EQ.NBTRC(J)) THEN
            UDOTAU=U(NP)
            UDOTAV=V(NP)
            UDOTAW=W(NP)
            XL=X(NP)
            YL=Y(NP)
```

```
            ZL=Z(NP)
            CALL TRCCG1(J,XL,YL,ZL,UDOTAU,UDOTAV,UDOTAW)
          END IF
        end do
        DO J=1,ITORIN
          IF (ABS(NBZONE(I,IRL)).EQ.NBTOR(J)) THEN
            UDOTAU=U(NP)
            UDOTAV=V(NP)
            UDOTAW=W(NP)
            XL=X(NP)
            YL=Y(NP)
            ZL=Z(NP)
            CALL TORCG1(J,XL,YL,ZL,UDOTAU,UDOTAV,UDOTAW)
          END IF
        end do
      end do
      IRNEAR=IRL
      IF (ITVALM.EQ.0) THEN
        TVAL0=1.E-4
        XISS=X(NP)+TVAL0*U(NP)
        YISS=Y(NP)+TVAL0*V(NP)
        ZISS=Z(NP)+TVAL0*W(NP)
2291    IF(X(NP).NE.XISS.OR.Y(NP).NE.YISS.OR.Z(NP).NE.ZISS) GO TO 2292
          TVAL0=TVAL0*10.
          XISS=X(NP)+TVAL0*U(NP)
          YISS=Y(NP)+TVAL0*V(NP)
          ZISS=Z(NP)+TVAL0*W(NP)
        GO TO 2291
2292    CONTINUE
        XIDD=DBLE(X(NP))+DBLE(TVAL0)*DBLE(U(NP))
        YIDD=DBLE(Y(NP))+DBLE(TVAL0)*DBLE(V(NP))
        ZIDD=DBLE(Z(NP))+DBLE(TVAL0)*DBLE(W(NP))
        CALL SRZONE(XIDD,YIDD,ZIDD,IRNEXT)
        IF (IRNEXT.NE.IRL) THEN
          TVAL=0.0
          IRNEAR=IRNEXT
        ELSE
          TVAL00=0.0
          TVAL10=10.0*TVAL0
          IRLOLD=IRL
          IRLFG=0
2301      IF (IRLFG.EQ.1) GO TO 2302
            TVAL00=TVAL00+TVAL10
            IF (TVAL00.GT.1.0E+06) THEN
              WRITE(6,2310)IQ(NP),IR(NP),X(NP),Y(NP),Z(NP), U(NP),V(NP),
     *        W(NP),TVAL00
2310          FORMAT(' TVAL00 ERROR : IQ,IR,X,Y,Z,U,V,W,TVAL=', 2I3,
     *        1P7E12.5)
              STOP
            END IF
            XIDD=DBLE(X(NP))+DBLE(TVAL00)*DBLE(U(NP))
            YIDD=DBLE(Y(NP))+DBLE(TVAL00)*DBLE(V(NP))
            ZIDD=DBLE(Z(NP))+DBLE(TVAL00)*DBLE(W(NP))
            CALL SRZOLD(XIDD,YIDD,ZIDD,IRLOLD,IRLFG)
          GO TO 2301
2302      CONTINUE
          TVAL=TVAL00
          DO J=1,10
            XIDD=DBLE(X(NP))+DBLE(TVAL00)*DBLE(U(NP))
            YIDD=DBLE(Y(NP))+DBLE(TVAL00)*DBLE(V(NP))
            ZIDD=DBLE(Z(NP))+DBLE(TVAL00)*DBLE(W(NP))
            CALL SRZONE(XIDD,YIDD,ZIDD,IRNEXT)
            IF (IRNEXT.NE.IRLOLD) THEN
              TVAL=TVAL00
              IRNEAR=IRNEXT
            END IF
            TVAL00=TVAL00-TVAL0
          end do
          IF (IRL.EQ.IRNEAR) THEN
            WRITE(0,*) 'IRL,TVAL=',IRL,TVAL
          END IF
        END IF
      ELSE
        DO J=1,ITVALM-1
          DO I=J+1,ITVALM
            IF ((ATVAL(I).LT.ATVAL(J))) THEN
              ATVALTMP=ATVAL(I)
              ATVAL(I)=ATVAL(J)
              ATVAL(J)=ATVALTMP
            END IF
          end do
        end do
```

```
        ITVLFG=0
        TVALMN=TVAL
        DO JJJ=1,ITVALM
          IF (TVALMN.GT.ATVAL(JJJ)) THEN
            TVALMN=ATVAL(JJJ)
          END IF
          DELHOW=1.E-4
          TVAL0=ATVAL(JJJ)+DELHOW
          XISS=X(NP)+TVAL0*U(NP)
          YISS=Y(NP)+TVAL0*V(NP)
          ZISS=Z(NP)+TVAL0*W(NP)
2361      IF(X(NP).NE.XISS.OR.Y(NP).NE.YISS.OR.Z(NP).NE.ZISS) GO TO 2362
            DELHOW=DELHOW*10.
            TVAL0=ATVAL(JJJ)+DELHOW
            XISS=X(NP)+TVAL0*U(NP)
            YISS=Y(NP)+TVAL0*V(NP)
            ZISS=Z(NP)+TVAL0*W(NP)
          GO TO 2361
2362      CONTINUE
          XIDD=DBLE(X(NP))+DBLE(TVAL0)*DBLE(U(NP))
          YIDD=DBLE(Y(NP))+DBLE(TVAL0)*DBLE(V(NP))
          ZIDD=DBLE(Z(NP))+DBLE(TVAL0)*DBLE(W(NP))
          CALL SRZONE(XIDD,YIDD,ZIDD,IRNEXT)
          IF ((IRNEXT.NE.IRL.OR.ATVAL(JJJ).GE.1.).AND.TVAL.GT.
     *    ATVAL(JJJ)) THEN
            TVAL=ATVAL(JJJ)
            IRNEAR=IRNEXT
            ITVLFG=1
            GOTO 2370
          END IF
        end do
2370    IF (ITVLFG.EQ.0) THEN
          TVAL0=1.E-4
          XISS=X(NP)+TVAL0*U(NP)
          YISS=Y(NP)+TVAL0*V(NP)
          ZISS=Z(NP)+TVAL0*W(NP)
2381      IF(X(NP).NE.XISS.OR.Y(NP).NE.YISS.OR.Z(NP).NE.ZISS) GO TO 2382
            TVAL0=TVAL0*10.
            XISS=X(NP)+TVAL0*U(NP)
            YISS=Y(NP)+TVAL0*V(NP)
            ZISS=Z(NP)+TVAL0*W(NP)
          GO TO 2381
2382      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(6,2390)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---------------------
!-----------------------------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=alog(enmu(i+1))
          enm0=alog(enmu(i))
          hnu1=alog(hnu(i+1))
          hnu0=alog(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=alog(enmu(i+1))
          enm0=alog(enmu(i))
          hnu1=alog(hnu(i+1))
          hnu0=alog(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-------------------------
```