SLAC-TN-06-002

iSHIELD11
A Line Source Application of SHIELD11

W. R. Nelson* and S. H. Rokni

Radiation Physics Group
Stanford Linear Accelerator Center
Stanford University
Stanford, CA 94309

February 15, 2006

Prepared for the Department of Energy under contract DE-AC02-76SF00515

*Department Associate in the Radiation Physics Group (retired)

Summary

iSHIELD11 performs a line-source numerical integration of radiation source terms that are defined by the SHIELD11 computer code[1] . An example is provided to demonstrate how one can use iSHIELD11 to perform a shielding analysis for a 250 GeV electron linear accelerator.

1 Introduction

SHIELD11 is a computer code for performing shielding analyses around a high-energy electron accelerator[1]. It makes use of simple analytic expressions for the production and attenuation of photons and neutrons resulting from electron beams striking thick targets, such as dumps, stoppers, collimators, and other beam devices. iSHIELD11 is an auxiliary code that performs an integration over the SHIELD11 kernal in order to take into account a line source that might be typical of an electron accelerator (*e.g.*, linear collider) losing power uniformly along its length.

2 iSHIELD11: Line integration of SHIELD11 source terms

Figure 1 depicts a line source along z of radiation from a uniform power loss, Q(W / m), caused by an electron beam of energy E(GeV). The primary shield is a cylindrical shell of earth having a length $L = L_2 - L1$.

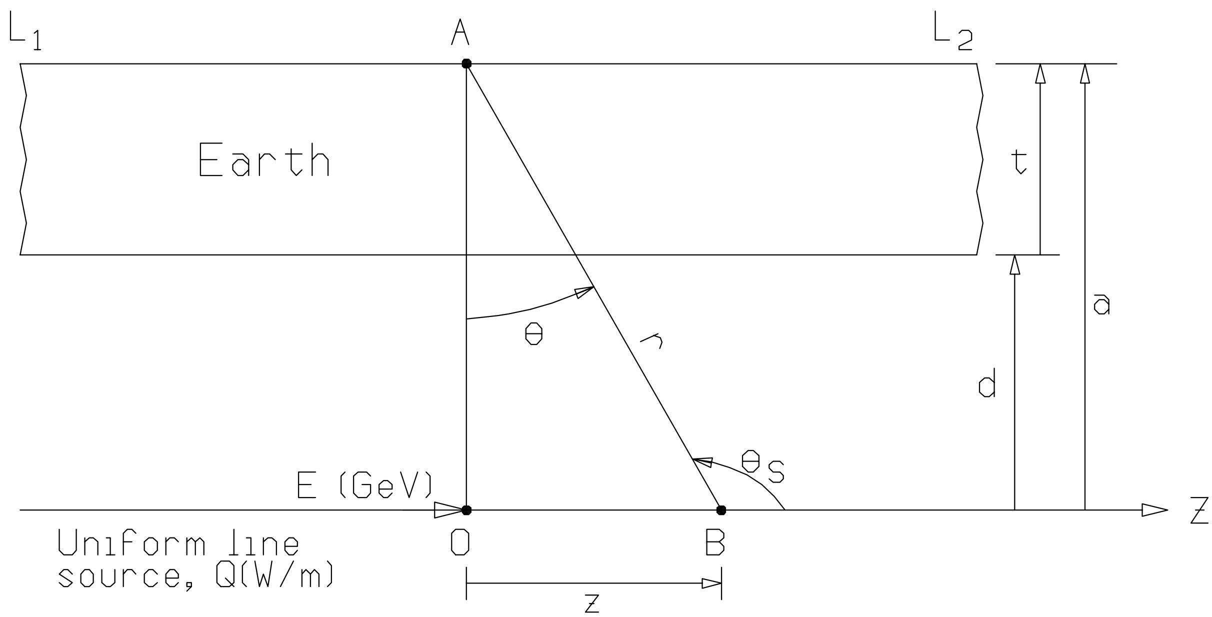

Fig. 1 Geometry for iSHIELD11 integration

The dose at a detector located at point A, from a beam loss over a differential length dz at B, is given by*

$$dD(\text{mrem/h}) = Q\frac{S(\theta_S)}{r^2}e^{-t\sec\theta/\lambda}\,dz, \tag{1}$$

*For example, see Chapter 5 of Kase and Nelson[2].

where

$$\begin{aligned} S(\theta_S) &= \text{SHIELD11 (point) source term (mrem/h/W at one meter)}, \\ \theta_S &= \text{detector angle in SHIELD11 coordinates}, \\ &= \theta + \pi/2, \\ t &= \text{thickness of primary (earth) shield}, \\ \lambda &= \text{attenuation mean free path of the primary shield}. \end{aligned}$$

The integrated dose at point A for a line source of length $L = L_2 - L1$ is then

$$D(\text{mrem/h}) = Q \int_{L_1}^{L_2} \frac{S(\theta_S)}{r^2} e^{-t \sec\theta/\lambda} dz. \tag{2}$$

From Fig. 1 we see that $z = (d+t)\tan\theta$ and $r = (d+t)/\cos\theta$, where d is the radius of the primary shield, so Eqn.2 becomes

$$D(\text{mrem/h}) = \frac{Q}{(d+t)} \int_{\theta_1}^{\theta_2} S(\theta) e^{-t\sec\theta/\lambda}\, d\theta, \tag{3}$$

where

$$\begin{aligned} \theta_1 &= \tan^{-1}(L_1/(d+t)), \\ \theta_2 &= \tan^{-1}(L_2/(d+t)). \end{aligned}$$

There are five components of radiation in the SHIELD11 model, so the *total* dose at the detector (point A) is given by

$$D_{tot} = \sum_{i=1}^{5} D_i\ , \tag{4}$$

where each D_i is given by Eqn. (3) using $S_i(\theta)$ along with the appropriate λ_i. The iSHIELD11 program therefore consists of coding for Eqns. (3) and (4) with copies of `FUNCTION Sorc` and `BLOCK DATA` from SHIELD11 attached in order to provide the S_i and the λ_i, respectively.

Only four materials are defined in `BLOCK DATA`; namely, concrete (`CONC`), iron (`FE`), lead (`PB`) and a *miscellaneous* material (`MISC`) that is left up to the discretion of the user to redefine. As recommended in the `BLOCK DATA` commentary, we have chosen `MISC` to represent "earth" and have assigned to it all of the parameters of `CONC`, with the exception of the density (`RHOmat`), which we have selected to be 1.70 g/cm^3. Also, we can apply additional attenuation—*e.g.*, in the form of inner shells of concrete, iron and/or lead—as demonstrated in the next section.

3 An example of running iSHIELD11

Our example of running iSHIELD11 is based on a previously published shielding study[3] for a 250-GeV electron linac that looses 0.25% of 4.2 MW uniformly and continuously over a distance of 10 km, resulting in a Q value of 1 W/m.

The geometry is an earth shield located immediately outside a 2-ft thick concrete shell (not depicted in Fig. 1). The inner radius of the concrete is 9-ft, so that d =11 ft, with the concrete handled by the code as a separate, fixed attenution factor. The integrated dose at point A on the surface of the shield will be obtained using $L_1 = -$16,404 ft and L_2 =16,404 ft, corresponding to the middle of the 10-km linac. Essentially, this results in an integration over θ between $-\pi/2$ and $\pi/2$.

The target model is chosen as follows. First of all, we know from experience[4, 5, 6] that the photon and low-energy neutron components will be significantly attenuated by the concrete and the first four or five feet of earth. As a result, the penetrating component of radiation will be high-energy neutrons created in, and radially escaping from, the core of the electromagnetic shower. Accompanying the high-energy neutrons will be the progeny created in the final foot or so of earth. Second, experience also tells us that by using a so-called *standard target—i.e.*, a 12-inch cylinder of iron having a radius of 2 inches—we will <u>maximize</u> the production of these high-energy neutrons. Furthermore, the high-energy neutron component scales directly with beam power, not with incident beam energy, so locating point A at the middle of the 10-km linac is only important in terms of providing reasonable integration limits† .

Therefore, in this example study, although we will assume that the beam is lost uniformly over 10 km, the target-input parameters are those for the *standard target*. Furthermore, we will account for the 2-ft of concrete by means of the `AddCON` parameter. Table 1 shows the iSHIELD11 input file based on these considerations.

Table 1
iSHIELD11 input file (`ishield11.in5`)

```
1                                IOsw (material output: 0=no, 1=yes) (I1)
     250.0       1.0             Ebeam (GeV), Qval (2F10.0)
FE                               NamTar (A4)
      12.0       2.0             TarLen (inch), TarRad (inch) (2F10.0)
0                                IattSW (n-atten. by target: 0=no, 1=yes) (I1)
MISC                             NamShl (A4)
                11.0             DisShl (ft) (10X,F10.0)
      25.0                       ThkShlMax (ft) (F10.0)
            -16404.0   16404.0   L1, L2 (ft) (10X,2F10.0)
      24.0                       AddCON (inch) (F10.0)
       0.0                       AddFE (inch) (F10.0)
       0.0                       AddPB (inch) (F10.0)
       0.0                       AddMIS (inch) (F10.0)
```

To be conservative, one should always select `IattSW=0` in order to <u>turn off</u> neutron attenuation by the target itself (otherwise the choice of `TarLen` and `TarRad` will become important!). Also, we have selected `IOsw=1` in order to produce a header output showing the *material* parameters (in `BLOCK DATA`) currently being used by iSHIELD11.

The output from running iSHIELD11 is provided in Table 2 below. The total dose rate is given as a function of the earth shield thickness along with each of the five components in the SHIELD11 model.

†It is instructive for the user of iSHIELD11 to increase the length and radius of the target, as well as to select different energies, to prove these points.

Table 2
iSHIELD11 output file (`ishield11.out6`)

```
*********************************************************************
                             MATERIAL DATA
*********************************************************************
MatID                    CONC      FE        PB       MISC

Zmat                    13.00    26.00    82.00    13.00
Amat                    26.98    55.85   207.19    26.98
RHOmat  (g/cu.cm)        2.35     7.87    11.35     1.70

RLmat   (g/sq.cm)       26.70    13.84     6.37    26.70
XMUmat  (g/sq.cm)       11.10    10.70    14.20    11.10
                                                            Isorc
Xmfp    (g/sq.cm)       30.00    47.00    97.00    30.00   1 (GRNs)
                        55.00   145.00   200.00    55.00   2 (MIDs)
                       120.00   145.00   200.00   120.00   3 (HENs)
                        42.00    33.60    24.00    42.00   4 (GamD)
                       120.00   145.00   200.00   120.00   5 (GamI)
*********************************************************************

Ebeam  = 250.00 GeV         -----------------       NamTar =  FE
Qval   =   1.00 W/m         iSHIELD11 results       TarLen =  12.00 in
NamShl =   MISC             -----------------       TarRad =   2.00 in
DisShl =  11.00 ft          AddCON = 24.00 in       IattSW =  0
L1     =  -16404.00 ft      AddFE  =  0.00 in
L2     =   16404.00 ft      AddPB  =  0.00 in
                            AddMIs =  0.00 in

----------------------------------------------------------------------
                             Dose Rate (mrem/h)
----------------------------------------------------------------------
T(ft)     GRN        MID        HEN        GamD       GamI       TOTAL
----------------------------------------------------------------------
 1.0   3.561E-01  2.262E+00  2.968E+00  6.977E+00  7.923E-01  1.336E+01
 2.0   5.275E-02  7.296E-01  1.557E+00  1.602E+00  4.157E-01  4.357E+00
 3.0   8.000E-03  2.413E-01  8.410E-01  3.835E-01  2.246E-01  1.698E+00
 4.0   1.235E-03  8.134E-02  4.643E-01  9.440E-02  1.240E-01  7.652E-01
 5.0   1.932E-04  2.782E-02  2.607E-01  2.371E-02  6.960E-02  3.820E-01
 6.0   3.056E-05  9.624E-03  1.484E-01  6.050E-03  3.962E-02  2.037E-01
 7.0   4.880E-06  3.362E-03  8.540E-02  1.563E-03  2.280E-02  1.131E-01
 8.0   7.851E-07  1.184E-03  4.962E-02  4.076E-04  1.325E-02  6.446E-02
 9.0   1.272E-07  4.197E-04  2.906E-02  1.072E-04  7.758E-03  3.734E-02
10.0   2.072E-08  1.497E-04  1.714E-02  2.840E-05  4.575E-03  2.189E-02
11.0   3.392E-09  5.367E-05  1.017E-02  7.570E-06  2.714E-03  1.294E-02
12.0   5.578E-10  1.933E-05  6.063E-03  2.028E-06  1.619E-03  7.703E-03
13.0   9.210E-11  6.992E-06  3.633E-03  5.460E-07  9.699E-04  4.610E-03
14.0   1.526E-11  2.538E-06  2.186E-03  1.476E-07  5.836E-04  2.772E-03
15.0   2.537E-12  9.246E-07  1.320E-03  4.005E-08  3.525E-04  1.674E-03
16.0   4.230E-13  3.378E-07  8.001E-04  1.090E-08  2.136E-04  1.014E-03
17.0   7.073E-14  1.237E-07  4.864E-04  2.977E-09  1.299E-04  6.164E-04
18.0   1.185E-14  4.545E-08  2.965E-04  8.151E-10  7.917E-05  3.758E-04
19.0   1.991E-15  1.673E-08  1.812E-04  2.237E-10  4.839E-05  2.297E-04
20.0   3.352E-16  6.173E-09  1.110E-04  6.156E-11  2.965E-05  1.407E-04
```

The total dose rate is also plotted in Fig. 2, where one sees that the Radiation Worker and General Population tolerances are attained with about 5 and 11 feet of earth, respectively.

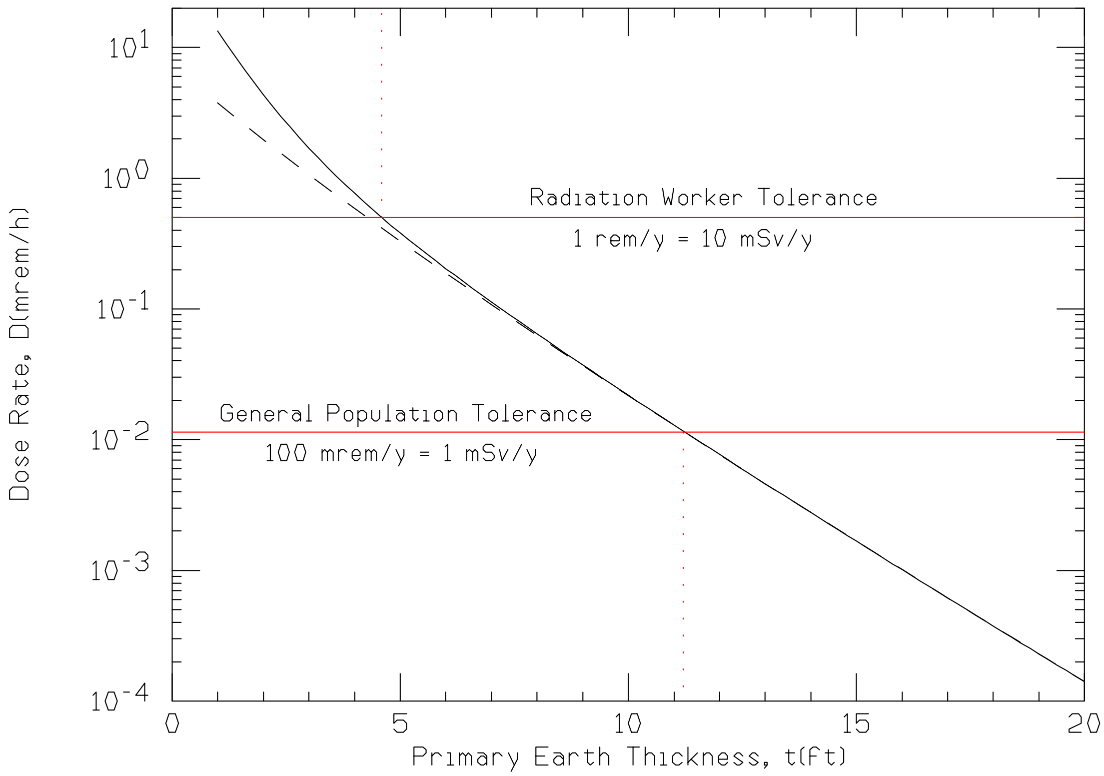

Fig. 2   Radiation level vs. earth shielding thickness.

The dashed line in this figure is the sum of the high-energy neutron and progeny components , demonstrating that it only takes four or five feet of earth, in addition to the 2-ft of concrete, to remove the other three components (ie , `GamD`, `GRN` and `MID`).

4 Concluding remarks

The MAIN purpose of iSHIELD11 is to provide a quick, “back-of-the envelope” estimate of the transverse shielding required for high-energy electron linacs. It takes only a few seconds to run iSHIELD11. However, since iSHIELD11 is based on the SHIELD11 code itself, one should fully appreciate the physics-scaling models that have been imployed in the parent code.

The full listing of iSHIELD11 is provided in the Appendix of this paper.

References

[1] W. R. Nelson and T. M. Jenkins, "The SHIELD11 Computer Code", SLAC Report Number SLAC-R-737 (February 2005).

[2] K. R. Kase and W. R. Nelson, *Concepts of Radiation Dosimetry* (Pergamon Press, New York, 1978).

[3] W. R. Nelson, S. H. Rokni and V. Vylet, "Radiation Calculations and Shielding Considerations for the Design of the Next Linear Collider", SLAC-PUB-7336 (November 1996).

[4] H. DeStaebler, Jr., "Transverse Radiation Shielding for the Stanford Two-Mile Accelerator", SLAC Report Number SLAC-R-9 (November 1962).

[5] H. DeStaebler, T. M. Jenkins and W. R. Nelson, "Shielding and Radiation", Chapter 26 in *The Stanford Two-Mile Accelerator*, R. B. Neal, Editor (Benjamin, 1968).

[6] T. M. Jenkins, "Neutron and photon measurements through concrete for a 15 GeV electron beam on a target—comparison with models and calculations", Nucl. Instr. Meth. 159 (1979) 265.

APPENDIX

A iSHIELD11 Code Listing

A.1 MAIN program

```
!***********************************************************************
!********************************   STANFORD LINEAR ACCELERATOR CENTER *
!***    i S H I E L D 1 1    ***                                       *
!********************************   Version:         11 NOV 2005/1800 *
!***********************************************************************
!***********************************************************************
!                        ----------------------
!                        COPYRIGHT (C) 1988-2005
!                        ----------------------
!    By the Board of Trustees of the Leland Stanford Junior University
!                          All Rights Reserved
!
!***********************************************************************
!                              ----------
!                              DISCLOSURE
!                              ----------
!   The SHIELD11 and iSHIELD11 codes have been developed at SLAC
!   under the sponsorship of the U.S. Government.  Neither the U.S.
!   nor the U.S.D.O.E., nor the Leland Stanford Junior University,
!   nor their employees, nor their respective contractors,
!   subcontractors, or their employees, makes any warranty, express
!   or implied, or assumes any liability or responsibility for
!   accuracy, completeness or usefulness of any information,
!   apparatus, product or process disclosed, or represents that its
!   use will not infringe privately-owned rights.  Mention of any
!   product, its manufacturer, or suppliers shall not, nor is it
!   intended to, imply approval, disapproval, or fitness for any par-
!   ticular use.  A royalty-free, non-exclusive right to use and dis-
!   seminate same for any purpose whatsoever is expressly reserved to
!   the U.S. and the University.
!
!***********************************************************************
!                           ---------------
!                           ACKNOWLEDGEMENT
!                           ---------------
!   The contributors acknowledge T. M. Jenkins for his part in develop-
!   ing the original is leading up to (and including) SHIELD11.
!
!***********************************************************************
!                        --------------------
!                        DISTRIBUTION CONTACT
!                        --------------------
!                 Walter R. Nelson
!                 Department Associate (retired)
!                 Radiation Physics Group, MS/48
!                 Stanford Linear Accelerator Center
!                 Stanford, CA 94309 U.S.A.
!                 Email: wrnrp@slac.stanford.edu
!
!                 and
!
!                 Sayed Rokni
!                 Head, Radiation Physics Group, MS/48
!                 Stanford Linear Accelerator Center
!                 Stanford, CA 94309 U.S.A.
!                 Email: rokni@slac.stanford.edu
!
!
!***********************************************************************
!                              -----------
```

```
!                              CODE HISTORY
!                              ------------
!  iSHIELD11 is a special application code related to SHIELD11,
!  whereby a numerical integration is performed using SHIELD11
!  source terms for a line source with parallel shield.
!
!  SHIELD11 is a code for performing shielding analysis around a high-
!  energy electron accelerator.  It makes use of simple analytic
!  expressions for the production of photons and neutrons by electron
!  beams striking thick targets (e.g., dumps, stoppers, etc.), and the
!  attenuation of these photons and neutrons.  The code, originally
!  written by T. M. Jenkins and W. R. Nelson, makes use of experimental
!  data from a number of sources and involves extrapolation (scaling)
!  based on rather simple physics concepts described in these sources
!  (e.g., see references listed below).
!
!  Because these scaling methods have only been tested over a rather
!  limited set of conditions (1-10 GeV electrons striking 10-20 r.l.
!  Fe and Pb targets), careful judgement should be exercised when using
!  SHIELD11.  To be more specific, users of SHIELD11 would be well
!  advised to fully understand the models used in the code before
!  trusting the results, particularly outside the range of design.
!
!***********************************************************************
!                          ---------------------
!                          PRIMARY DOCUMENTATION
!                          ---------------------
!
!       W. R. Nelson and T, M. Jenkins, "The SHIELD11 Computer Code",
!       SLAC-Report-737, February 2005.
!
!                            ------------------
!                            RELATED REFERENCES
!                            ------------------
!
!    1) H. DeStaebler, T. M. Jenkins and W. R. Nelson, "Shielding and
!       Radiation", Chapter 26 in THE STANFORD TWO-MILE ACCELERATOR,
!       R. B. Neal, Editor (Benjamin, 1968).
!    2) T. M. Jenkins, "Neutron and Photon Measurements through Con-
!       crete for a 15 GeV Electron Beam on a Target -- Comparison with
!       Models and Calculations", Nucl. Instr. Meth. 159 (1979) 265.
!    3) W. P. Swanson and R. H. Thomas, "Dosimetry for Radiological
!       Protection at High-Energy Particle Accelerators", Chapter 1
!       in THE DOSIMETRY OF IONIZING RADIATION, Volume III, K. R. Kase,
!       B. E. Bjarngard and F. H. Attix, Editors (Academic Press, 1990).
!    4) W. P. Swanson, RADIOLOGICAL SAFETY ASPECTS OF THE OPERATION OF
!       ELECTRON LINEAR ACCELERATORS, IAEA Technical Reports Series
!       No. 188 (Vienna, 1979).
!    5) A. Fasso, K. Goebel, M. Hoefert, J. Ranft, G. Stevenson,
!       "Shielding Against High-Energy Radiation", in Volume 11
!       Landolt-Bornstein - Numerical Data and Functional Relationships
!       in Science and Technology (Springer-Verlag, 1990).
!    6) H. Hirayama and S. Ban, "Review of Shielding Problems Concern-
!       ing Electron Accelerators", p.107-132 in Proceedings of the
!       Specialists' Shielding Meeting on SHIELDING ASPECTS OF ACCELERA-
!       TORS, TARGETS AND IRRADIATION FACILITIES, held in Arlington,
!       Texas (28-29 April 1994); Nuclear Energy Agency OECD Document.
!
!***********************************************************************
!
!                      ------------------------------
!                      Special Geometry for iSHIELD11
!                      ------------------------------
!
!                         Additional      Primary
!                         Shielding      Shielding
!                         (concrete)      (dirt)
!
!                       |<--AddCON-->|<----ThkShl---->|
!                          (fixed)        (varies)
```

```
!            .          |            |                |
!            .          |            |                |
!  Line      .          |            |                |
! Source --> .          |            |                |* <---Detector
!            .          |            |                |
!            .          |            |                |
!            .          |            |                |
!
!            |<-------DisShl-------->|                |
!                    (fixed)
!
!-----------------------------------------------------------------------
! Note1: In addition to concrete (AddCON), one can also add Fe (AddFE)
!        and/or Pb (AddPB) shielding between the source and the primary
!        earth (MISC=dirt) shielding.
!-----------------------------------------------------------------------
! Note2: An input file (ishield11.in5), similar to the one that is
!        used with SHIELD11 (shield11.in5), is required with iSHIELD11
!        and must reside in the same directory.
!-----------------------------------------------------------------------
!***********************************************************************
!23456789|123456789|123456789|123456789|123456789|123456789|123456789|12

!=======================================================================
!                            MAIN program
!=======================================================================
      implicit none

      include 'include/shield11_h.fh'     ! "Header" file for parameters

!     ----------------
!     SHIELD11 COMMONs
!     ----------------
      include 'include/beam.fh'                              ! Beam data
      include 'include/miscon.fh'             ! Miscellaneous constants
      include 'include/shield.fh'                          ! Shield data
      include 'include/target.fh'                          ! Target data

      common/passit/BigL1,BigL2,Qval,iThkShlMax,Isorc
      real*8 BigL1,BigL2,Qval
      integer iThkShlMax,Isorc

      real*8                                           ! Local variables
     + DoseR(5),
     + QvalP,epkWh,ThRAD1,ThRAD2,Total,Smalla
      integer
     + IThk,IOsw

      real*8
     + Fint(5),
     + DCADRE,FunD,AERR,RERR,
     + ERRDUM                         ! Bound on the absolute error of
                                      ! the output number, DCADRE.
      integer
     + IER                                             ! Error parameter

      external FunD

      data AERR/1.D-60/   ! Desired absolute error in the answer (INPUT)
      data RERR/1.D-6/    ! Desired relative error in the answer (INPUT)

!     ----------
!     Open files
!     ----------
      open(UNIT= 5,FILE='ishield11.in5',STATUS='OLD')
      open(UNIT= 6,FILE='ishield11.out6',STATUS='UNKNOWN')
      open(UNIT=66,FILE='ishield11.out66',STATUS='UNKNOWN')

!     ==================
      call iDataIO(IOsw)                         ! Read/print input data
```

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

!     --------------------------
!     Determine other parameters
!     --------------------------
      QvalP=Qval*1.D-5                        ! Linear beam loss (kW/cm)
      epkWh=2.25D+16/Ebeam                           ! Electrons per kWh

!     =================================
!     Start of MAIN calculation (loops)
!     =================================
      write(6,'(/,
     +'' ------------------------------------------------------------'',
     +''-----------'',/,
     +30X,''Dose Rate (mrem/h)'',/,
     +'' ------------------------------------------------------------'',
     +''-----------'',/,
     +'' T(ft)'',5X,''GRN'',8X,''MID'',8X,''HEN'',7X,''GamD'',7X,
     +''GamI'',7X,''TOTAL'',/,
     +'' ------------------------------------------------------------'',
     +''-----------'')')

!     -------------------------------------------------
!     Step over total number of feet for PRIMARY shield
!     -------------------------------------------------
      do IThk=1,iThkShlMax
        ThkShl=IThk*30.48D0              ! PRIMARY shield thickness (cm)
        Smalla=DisShl + ThkShl

        ThRAD1=atan(BigL1/Smalla)    ! Lower integration limit (radians)
        ThRAD2=atan(BigL2/Smalla)    ! Upper integration limit (radians)

        Total=0.D0
!       ---------------------------------------------------
!       Get integrated dose rate (mrem/h)for each component
!       ---------------------------------------------------
        do Isorc=1,5
!         -----------
!         rem-sq.cm/e
!         -----------
          Fint(Isorc)=DCADRE(FunD,ThRAD1,ThRAD2,AERR,RERR,ERRDUM,IER)

!         -----------------------------------------------------------
!         Check for possible error messages from DCADRE that are fatal
!         (i.e., IER=131, 132 or 133).  Warning errors (IER=65 and 66)
!         are also possible, but only IER=66 type have been observed
!         so far with iSHIELD11, which is acceptable.
!         -----------------------------------------------------------
          if (IER.gt.130) then
            CALL ErrDCADRE(IER,ThRAD1,ThRAD2,Fint,ERRDUM)
          end if

!         ------------------
!         mrem/h at detector
!         ------------------
          DoseR(Isorc)=Fint(Isorc)*epkWh*QvalP*1.D3/Smalla

          Total=Total + DoseR(Isorc)
        end do                                       ! End of Isorc loop

        write(6,'(F5.1,1X,1P,6E11.3)') ThkShl/30.48D0,
     +    (DoseR(Isorc),Isorc=1,5),Total

        write(66,'(F5.1,1X,1PE11.3)') ThkShl/30.48D0,DoseR(3)+DoseR(5)

      end do                                          ! End of IThk loop

!     ----------
!     Close files
!     ----------
```

```
      close(UNIT=5)
      close(UNIT=6)

      stop
      end
!----------------------last line of MAIN program-----------------------
```

A.2 SUBROUTINE iDataIO

```
!***********************************************************************
!                                  STANFORD LINEAR ACCELERATOR CENTER *
!     -----------------------
      SUBROUTINE iDataIO(IOsw)
!     -----------------------
!                                  Version:          11 NOV 2005/1800 *
!***********************************************************************
!     Subprogram to read in and print out various input data.
!***********************************************************************
      implicit none

      include 'include/shield11_h.fh'     ! "Header" file for parameters

!     ----------------
!     SHIELD11 COMMONs
!     ----------------
      include 'include/beam.fh'                               ! Beam data
      include 'include/matdta.fh'                         ! Material data
      include 'include/shield.fh'                           ! Shield data
      include 'include/target.fh'                           ! Target data

      common/passit/BigL1,BigL2,Qval,iThkShlMax,Isorc
      real*8 BigL1,BigL2,Qval
      integer iThkShlMax,Isorc

      real*8                                            ! Local variables
     + ThkShlMax
      integer IOsw,i,j
      character*4 NamTar,NamShl

!     ==================
      read(5,'(I1)') IOsw                                    ! I/O switch
!     ==================
      if (IOsw.lt.0 .or. IOsw.gt.1) then
        write(6,*) 'Program stopped: BAD IOsw = ',IOsw
        stop
      end if

!     -------------------------------------------
!     Print (echo) the material data in BLOCK DATA
!     -------------------------------------------
      if (IOsw.eq.1) then
        write(6,'('' ***************************************************
     +*******************'')')
        write(6,(''                               MATERIAL DATA'')')
        write(6,'('' ***************************************************
     +*******************'')')
        write(6,'('' MatID           '',3X,4(5X,A4),/)') MatID
        write(6,'('' Zmat            '',3X,4(2X,F7.2))') Zmat
        write(6,'('' Amat            '',3X,4(2X,F7.2))') Amat
        write(6,'('' RHOmat (g/cu.cm)'',3X,4(2X,F7.2),/)') RHOmat
        write(6,'('' RLmat  (g/sq.cm)'',3X,4(2X,F7.2))') RLmat
        write(6,'('' XMUmat (g/sq.cm)'',3X,4(2X,F7.2))') XMUmat
        write(6,'(60X,''Isorc'')')
        do j=1,5
          if (j.eq.1) then
            write(6,'('' Xmfp   (g/sq.cm)'',3X,4(2X,F7.2),3X,I1,
     +        '' ('',A4,'')'')')
     +       (Xmfp(i,j),i=1,4),j,SorNam(j)
```

```
          else
            write(6,'(20X,4(2X,F7.2),3X,I1,'' ('',A4,'')'')')
     +         (Xmfp(i,j),i=1,4),j,SorNam(j)
          end if
        end do
        write(6,'('' ***************************************************
     +*******************'')')
      end if

!     -------------------------------------------------------
!     Read in the necessary beam, target and shield parameters
!     -------------------------------------------------------
!     ============================
      read(5,'(2F10.0)') Ebeam,Qval ! Beam energy (GeV), beam loss (W/m)
!     ============================

!     =====================
      read(5,'(A4)') NamTar                            ! Target material
!     =====================
      IDtar = 0
      do i=1,4
        if (NamTar.eq.MatID(i)) then
          IDtar = i
          go to 1
        end if
      end do
      write(6,*) 'Program stopped: Could not find NamTar = ',NamTar
      stop
 1    continue

!     ================================
      read(5,'(2F10.0)') TarLen,TarRad  ! Target length, radius (inches)
!     ================================

!     =====================
      read(5,'(I1)') IattSW          ! Neutron target-attenuation switch
!     =====================
      if (IattSW.ne.0 .and. IattSW.ne.1) then
        write(6,*) 'Program stopped: BAD IattSW = ',IattSW
        stop
      end if

!     =====================
      read(5,'(A4)') NamShl                    ! PRIMARY shield material
!     =====================
      IDshl = 0
      do i=1,4
        if (NamShl.eq.MatID(I)) then
          IDshl = i
          go to 2
        end if
      end do
      write(6,*) 'Program stopped: Could not find NamShl = ',NamShl
      stop
 2    continue

!     ============================
      read(5,'(10X,F10.0)') DisShl     ! Distance to PRIMARY shield (ft)
!     ============================

!     ===========================
      read(5,'(F10.0)') ThkShlMax ! Maximum PRIMARY shield thickness(ft)
!     ===========================
      iThkShlMax=ThkShlMax

!     ==================================
      read(5,'(10X,2F10.0)') BigL1,BigL2            ! Integration limits
!     ==================================
!     ========================
      read(5,'(F10.0)') AddCON              ! Additional concrete (inch)
```

```
      read(5,'(F10.0)') AddFE                   ! Additional iron (inch)
      read(5,'(F10.0)') AddPB                   ! Additional lead (inch)
      read(5,'(F10.0)') AddMIS         ! Additional MISC material (inch)
!     ========================


!     ------------------------------------
!     Print out the above input parameters
!     ------------------------------------
      write(6,'(/,'' Ebeam  ='',F7.2,'' GeV'',T30,
     + ''-----------------'',
     + T55,''NamTar ='',2X,A4)') Ebeam,NamTar

      write(6,'('' Qval   ='',F7.2,'' W/m'',T30,
     + ''iSHIELD11 results'',
     + T55,''TarLen ='',F7.2,'' in'')') Qval,TarLen

      write(6,'('' NamShl ='',3X,A4,T30,
     + ''-----------------'',
     + T55,''TarRad ='',F7.2,'' in'')') NamShl,TarRad

      write(6,'('' DisShl ='',F7.2,'' ft'',
     +    T30,''AddCON ='',F6.2,'' in'',
     +    T55,''IattSW ='',I3)') DisShl,AddCON,IattSW

      write(6,'('' L1     ='',F11.2,'' ft''
     +    T30,''AddFE  ='',F6.2,'' in'')') BigL1,AddFE

      write(6,'('' L2     ='',F11.2,'' ft''
     +    T30,''AddPB  ='',F6.2,'' in'')') BigL2,AddPB

      write(6,'(
     +    T30,''AddMIs ='',F6.2,'' in'')') AddMIS

!     ------------------------------
!     Convert inches and/or ft to cm
!     ------------------------------
      TarLen = TarLen*2.54D0
      TarRad = TarRad*2.54D0
      DisShl = DisShl*30.48D0
      BigL1     = BigL1*30.48
      BigL2     = BigL2*30.48

      AddCON = AddCON*2.54D0
      AddFE  =  AddFE*2.54D0
      AddPB  =  AddPB*2.54D0
      AddMIS = AddMIS*2.54D0

      return
      end
!--------------------last line of SUBROUTINE iDataIO--------------------
```

A.3 FUNCTION FunD

```
!***********************************************************************
!                                   STANFORD LINEAR ACCELERATOR CENTER *
!     -------------------------------------
      DOUBLE PRECISION FUNCTION FunD(ThRAD)
!     -------------------------------------
!                                   Version:          11 NOV 2005/1800 *
!***********************************************************************
!     Subprogram to provide integrand for SUBROUTINE DCADRE.
!***********************************************************************
      implicit none

      include 'include/shield11_h.fh'     ! "Header" file for parameters

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

```
!     SHIELD11 COMMONs
!     ----------------
      include 'include/beam.fh'                               ! Beam data
      include 'include/matdta.fh'                         ! Material data
      include 'include/miscon.fh'              ! Miscellaneous constants
      include 'include/shield.fh'                           ! Shield data
      include 'include/target.fh'                           ! Target data

      common/passit/BigL1,BigL2,Qval,iThkShlMax,Isorc
      real*8 BigL1,BigL2,Qval
      integer iThkShlMax,Isorc

      real*8                                            ! Local variables
     + Sorc,Dsorc,
     + ThRAD,ThDEG,ThSorcDEG,NumMFP,Atten

      external Sorc

      ThDEG=ThRAD/DegRad                   ! Integration angle (degrees)
      ThSorcDEG=ThDEG + 90.D0 !Integration angle wrt to source (degrees)

!     -------------------------------
!     Get Sorc term (in rem/e at 1 cm)
!     -------------------------------
      Dsorc=Sorc(Isorc,IDtar,TarLen,TarRad,IattSW,Ebeam,ThSorcDEG)

!     ----------------------------------------------
!     Apply PRIMARY and ADDITIONAL shield attenuation
!     ----------------------------------------------
      NumMFP=ThkShl*RHOmat(IDshl)/Xmfp(IDshl,Isorc) +           ! Primary
     +       AddCON*RHOmat(1)/Xmfp(1,Isorc)         +          ! Concrete
     +       AddFE*RHOmat(2)/Xmfp(2,Isorc)          +                ! Fe
     +       AddPB*RHOmat(3)/Xmfp(3,Isorc)                           ! Pb

      NumMFP=NumMFP/cos(ThRAD)

      if (NumMFP.gt.75.D0) then
        Atten=0.D0
      else
        Atten=exp(-NumMFP)
      end if

      FunD=Dsorc*Atten                            ! rem/e at 1 cm (still)

      return
      end
!----------------------last line of FUNCTION FunD----------------------
```

A.4 BLOCK DATA

```
!*********************************************************************
!********************************   STANFORD LINEAR ACCELERATOR CENTER *
      BLOCK DATA
!********************************   Version:          11 NOV 2005/1800 *
!*********************************************************************
!---------------------------------------------------------------------
!                       Material Constants (notes)
!---------------------------------------------------------------------
!   1) The Xmfp values for GRN (fast neutron) attenuation by high-Z
!      materials are based on the so-called `lid-tank' experiment.
!      These removal cross sections are given in Table 8.3 (p.231) of
!      Chilton, Shultis and Faw, PRINCIPLES OF RADIATION SHIELDING
!      (Prentice-Hall, 1984).  For example:
!
!         Iron     1.98 b/atom --> 47 g/sq.cm
!         Lead     3.53 b/atom --> 97 g/sq.cm
!
!      which are used in SHIELD11 (see below).  However, use of these
```

```
!      Xmfp values implies that an appropriate amount of hydrogeneous
!      material FOLLOWS the high-Z shield.  Typically, this amounts to
!      about 2-ft of concrete or 20-30 cm of polyethylene (e.g., see
!      Figure 8.6 ibid).  If there is NO moderating material between
!      the detector and the source (or source plus high-Z shielding),
!      as in the case when a value of zero is assigned to ThkShl,
!      a series of MORSE calculations and experiments by Jenkins
!      suggest that Xmfp=135 should be used for FE (and PB).
!      Accordingly, this is the default value we have assigned below
!      to Xmfp(4,1) for 'MISC' (i.e., we assume 'MISC' is 'PB' without
!      moderator).
!   2) For earth, one can use CONC directly by changing RHOmat(1) from
!      2.35 to 1.70 g/cu.cm, or one can copy all the 'CONC' data to
!      'MISC' and make RHOmat(4)=1.70, thus allowing the use of both
!      'CONC' and 'MISC', simultaneously.
!***********************************************************************
      implicit none

      include 'include/shield11_h.fh'     ! "Header" file for parameters

!     ----------------
!     SHIELD11 COMMONs
!     ----------------
      include 'include/matdta.fh'                         ! Material data
      include 'include/miscon.fh'              ! Miscellaneous constants
      include 'include/sorcon.fh'              ! FUNCTION Sorc constants

!     ------------------
!     For common/MATDTA/
!     ------------------
      DATA MatID /'CONC',  'FE  ', 'PB  ','MISC'/          ! Material ID
      DATA Zmat  /  13.0,  26.0,   82.0,   13.0 /        ! Atomic Number
      DATA Amat  / 26.98, 55.85, 207.19,  26.98 / ! Atomic Mass (g/mole)
      DATA RHOmat/  2.35,  7.87,  11.35,   1.70 /    ! Density (g/cu.cm)
      DATA RLmat /  26.7, 13.84,   6.37,   26.7 /  ! Rad. Len. (g/sq.cm)
      DATA XMUmat/  11.1,  10.7,   14.2,   11.1 /! Moliere Len.(g/sq.cm)
      DATA Xmfp  /  30.0,  47.0,   97.0,   30.0,  ! MFP (g/sq.cm) - GRNs
     +              55.0, 145.0,  200.0,   55.0,  !               - MIDs
     +             120.0, 145.0,  200.0,  120.0,  !               - HENs
     +              42.0,  33.6,   24.0,   42.0,  !               - GamD
     +             120.0, 145.0,  200.0,  120.0 / !               - GamI
      DATA SorNam/'GRNs','MIDs','HENs','GamD','GamI'/
!     -----------------------------------------------------------------

!     ------------------
!     For common/SORCON/
!     ------------------
!     ------------------------------------------
!     Thresholds and minimums (for FUNCTION Sorc)
!     ------------------------------------------
      DATA ThrHEN/0.150D0/     ! Threshold energy (GeV) for HEN production
      DATA TarLenG/0.01D0/     ! Minimum target length for photons (r.l.)
      DATA TarLenN/17.332196D0/! Minimum target length for neutrons (r.l.)
      DATA TarRadG/1.189869D0/ ! Minimum target radius for photons
      DATA TarRadN/3.736411D0/ ! Minimum target radius for neutrons (Mol.units)
!     ---------------------------------------------
!     Data fit to HEN cross section by T. M. Jenkins
!     ---------------------------------------------
      DATA E_HEN/
     +         0.15,   0.16,   0.17,   0.18,   0.19,   0.20,   0.21,
     +         0.22,   0.23,   0.24,   0.25,   0.27,   0.30,   0.35,
     +         0.40,   0.45,   0.50,   0.55,   0.60,   0.70,   0.80,
     +         0.90,   1.00/
      DATA CS_HEN/
     +      0.00444,0.00711, 0.0111, 0.0156, 0.0222, 0.0298, 0.0382,
     +       0.0489, 0.0547, 0.0622, 0.0711, 0.0889,  0.116,  0.162,
     +        0.211,  0.276,  0.338,  0.404,  0.502,  0.601,  0.711,
     +        0.813,   1.00/
!     -----------------------------------------------------------------
```

```
!     -----------------
!     For common/MISCON/
!     -----------------
      DATA DegRad/1.745329D-2/                ! Pi/180 (radians/degree)

      end
!--------------------last line of BLOCK DATA---------------------------
```

A.5 FUNCTION Sorc

```
!***********************************************************************
!*******************************   STANFORD LINEAR ACCELERATOR CENTER *
!
      FUNCTION Sorc(Isorc,IDtar,TarLen,TarRad,IattSW,Energy,Angle)
!
!*******************************   Version:           1 FEB 2005/0800 *
!***********************************************************************
! This subprogram returns the dose equivalent (rem/e) at 1 cm.
!***********************************************************************
!    Isorc    Source Type:  1 = Giant Resonance Neutrons (GRNs)
!                           2 = Mid-Energy Neutrons (MIDs)
!                           3 = High-Energy Neutrons (HENs)
!                           4 = Gamma Direct (EM cascade) (GamD)
!                           5 = Gamma Indirect (hadron cascade) (GamI)
!
!    IDtar    Target ID:    1 = CONC   (concrete)
!                           2 = FE     (iron)
!                           3 = PB     (lead)
!                           4 = MISC   (default is lead w/o moderator)
!    TarLen   Target length (cm).  Restrictions: Should be larger than
!             0.01 r.l. for photons & larger than 17.3 r.l. for neutron
!             attenuation by the target itself (i.e., when IattSW=1).
!    TarRad   Target radius (cm).  Restrictions: Should be larger than
!             2-in for Fe (or an equivalent number of GamD relaxation
!             lengths for other materials) for neutrons.  Also, should
!             be at least 3.74 Moliere Units in radius for neutron
!             attenuation by the target itself (i.e., when IattSW=1).
!    IattSW   Switch for neutron attenuation by the target material
!             itself (0=no attenuation, 1=attenuation).
!             [Note: Target length must be .GE. 17.3 r.l. and target
!                    radius must be .GE. 3.74 Moliere Units].
!    Energy   Energy of electron beam (GeV)
!             [Note: Below 1 GeV, approximation schemes (based on data
!                    fitting, etc.) are applied to the MID and HEN
!                    neutron terms.]
!    Angle    Production angle (degrees)
!***********************************************************************
      implicit none

      include 'include/shield11_h.fh'     ! "Header" file for parameters

!     ---------------
!     SHIELD11 COMMONs
!     ---------------
      include 'include/matdta.fh'                          ! Material data
      include 'include/miscon.fh'               ! Miscellaneous constants
      include 'include/sorcon.fh'               ! FUNCTION Sorc constants

      real*8                                             ! Local variables
     + Sorc,TarLen,TarRad,Energy,Angle,
     + AbsAng,CosThe,SinThe,SltSor,CritTar,SltTar,CritCor,SltCor,
     + TarLenGCM,TarLenRL,TarRadGCM,TarRadMU,TarRadRelax,
     + DelCS,Sorc1,Sorc2
      integer Isorc,IDtar,IattSW,k

      AbsAng = ABS(Angle)
      CosThe = COS(DegRad*AbsAng)
      SinThe = SIN(DegRad*AbsAng)
```

```
      TarLenGCM = TarLen*RHOmat(IDtar)         ! Target length (g/sq.cm)
      TarLenRL = TarLenGCM/RLmat(IDtar)           ! Target length (r.l.)
      TarRadGCM = TarRad*RHOmat(IDtar)         ! Target radius (g/sq.cm)
      TarRadMU = TarRadGCM/XMUmat(IDtar) ! Target radius (Moliere units)
      TarRadRelax = TarRadGCM/Xmfp(IDtar,4)           ! Relaxation units

      SltSor = 0.0

!     ---------------------------------------
!     Check for neutron attenuation by target
!         (GRN, MID, HEN, and GamI only)
!     ---------------------------------------
      if (IattSW.eq.1 .and. Isorc.ne.4) then
        if (TarLenRL.lt.TarLenN) then
          write(6,'(/,''Stopped in FUNCTION Sorc with TarLenRL='',G15.7,
     +      '' r.l. < TarLenN='',G15.7,'' r.l.)'')')
     +      TarLenRL,TarLenN
          stop
        end if

        if (TarRadMU.lt.TarRadN) then
          write(6,'(/,''Stopped in FUNCTION Sorc with TarRadMU '',G15.7,
     +      '' m.u.) < TarRadN ('',G15.7,'' m.u.)'')')
     +      TarRadMU,TarRadN
          stop
        end if

        if (AbsAng.gt.90.D0) then                ! NO target attenuation
          SltSor = 0.0                           ! in backward direction
          go to 1000
        end if

!       ----------------------------------------------------
!       Find critical angle and SltTar for cylindrical target
!       ----------------------------------------------------
        CritTar = atan(TarRad/TarLen)/DegRad                 ! (degrees)
        if (AbsAng.ge.CritTar) then
          SltTar = TarRadGCM/SinThe                          ! (g/sq.cm)
        else
          SltTar = TarLenGCM/CosThe                          ! (g/sq.cm)
        end if

!       --------------------------------------------------
!       Find critical angle and SltCor for cylindrical core
!       --------------------------------------------------
        CritCor = atan(TarRadN*XMUmat(IDtar)/TarLenN/RLmat(IDtar))/
     +            DegRad                                     ! (degrees)
        if (AbsAng.ge.CritCor) then
          SltCor = TarRadN*XMUmat(IDtar)/SinThe              ! (g/sq.cm)
        else
          SltCor = TarLenN*RLmat(IDtar)/CosThe               ! (g/sq.cm)
        end if

        SltSor = SltTar - SltCor                             ! (g/sq.cm)

      end if                  ! End of IattSW.eq.1 .and. Isorc.ne.4 loop

!     =====================================
!     Branch to the appropriate source term
!     =====================================
!             GRN  MID  HEN  GamD  GamI
!            -------------------------
 1000 go to (  1,   2,   3,   4,    3   ) Isorc
!            -------------------------

      write(6,'(/,'' Stopped in FUNCTION Sorc: Isorc='',
     +  I5,'' is bad.'')')
     +  Isorc
      stop
```

```
!     ==============================
!     GIANT-RESONANCE NEUTRONS (GRN)
!     ==============================
 1    Sorc = 4.93D0*Zmat(IDtar)**0.662D0
      Sorc=Sorc*Energy*1.D-11                            ! rem/e at 1 cm
      if (IattSW.eq.1) then                    ! Apply target attenuation
        Sorc = Sorc*exp(-SltSor/Xmfp(IDtar,Isorc))
      end if
      return

!      =========================
!      MID-ENERGY NEUTRONS (MID)
!      =========================
 2    Sorc = 43.9D0/Amat(IDtar)**0.37D0/(1.D0 - 0.75D0*CosThe)
      if (Energy.le.0.5D0) then
        Sorc = Sorc*1.6D0*Energy**1.5D0
      else if (Energy.gt.0.5D0 .and. Energy.lt.1.D0) then
        Sorc = Sorc*(0.566D0 + 0.434D0*(Energy - 0.5D0)/0.5D0)
      end if
      Sorc = Sorc*Energy*1.D-11                          ! rem/e at 1 cm
      if (IattSW.eq.1) then                    ! Apply target attenuation
        Sorc = Sorc*exp(-SltSor/Xmfp(IDtar,Isorc))
      end if
      return

!     ==========================
!     HIGH-ENERGY NEUTRONS (HEN)
!     ==========================
 3    if (Energy.le.ThrHEN) then                       ! Below threshold
        Sorc = 0.D0
      else
        Sorc = 13.7D0/Amat(IDtar)**0.65D0/(1.D0 - 0.72D0*CosThe)**2D0
      end if

      if (Energy.lt.1.D0) then   ! Loop over the energy bins below 1 GeV
        do k=2,23
          if (Energy.lt.E_HEN(k)) then
            DelCS = (CS_HEN(k) - CS_HEN(k-1))*(Energy - E_HEN(k-1))/
     +              (E_HEN(k) - E_HEN(k-1))
            Sorc = (CS_HEN(k-1) + DelCS)*Sorc
            go to 1100
          end if
        end do
      end if
                          ! ==========================================
 1100 if (Isorc.eq.5) then  ! INDIRECT PHOTONS FROM HADRON SHOWER (GamI)
                          ! ==========================================
        Sorc = 0.267D0*Sorc          ! Sorc on the right is for the HENs
      end if
      Sorc = Sorc*Energy*1.D-11                          ! rem/e at 1 cm
      if (IattSW.eq.1) then                    ! Apply target attenuation
        Sorc = Sorc*exp(-SltSor/Xmfp(IDtar,Isorc))
      end if
      return

!     =================================================
!     DIRECT PHOTONS FROM ELECTROMAGNETIC SHOWER (GamD)
!     =================================================
 4    if (TarLenRL.le.TarLenG) then        ! TarLenG=0.01=minimum target
                                           !          length for photons

        write(6,'(/,''Stopped in FUNCTION Sorc with TarLenRL'',
     +    G15.7,'' r.l. <= TarLenG ='',G15.7,'' r.l.'')')
     +    TarLenRL,TarLenG
        stop
      end if

      if (TarRadRelax.lt.TarRadG) then      ! TarRadG=1.189869D0=minimum
                                            !  target radius for photons
        write(6,'(/,''Stopped in FUNCTION Sorc with TarRadRelax='',
```

```
     +    G15.7,'' < TarRadG='',G15.7)')
     +    TarRadRelax,TarRadG
        stop
      end if

!     -----------------------------------------------------------------
!     The GamD model is based on fitting experimental data (vs. theta)
!     for the standard target (12" long cylinder of Fe (R=2")) using two
!     functional forms:    y=a*exp(b*sqrt(theta)) for 0 thru   5 degrees
!                      and y=a*exp(b*theta)       for 5 thru 180 degrees
!     After normalization, the two source terms are summed.  However,
!     as shown below, two different attenuation factors are applied to
!     the 5-180 degree source term.  Namely,
!
!       exp(-TarRadRelax) for     0 < theta <  90 degrees
!       exp(-TarRadG)     for    90 < theta < 180 degrees
!     -----------------------------------------------------------------
      Sorc1 = 1.06D6*Energy*exp(-TarLenGCM/Xmfp(IDtar,4))*
     +          exp(-0.959D0*AbsAng**0.5D0)

      if (AbsAng.le.90.D0) then
        Sorc2 = 683.D0*exp(-TarRadRelax)*exp(-AbsAng/72.2D0)
      else
        Sorc2 = 683.D0*exp(-TarRadG)*exp(-AbsAng/72.2D0)
      end if

      Sorc = Sorc1 + Sorc2
      Sorc = Sorc*Energy*1.D-11                          ! rem/e at 1 cm
      return

      end
!--------------------last line of FUNCTION Sorc-------------------------
```

A.6 SUBROUTINE ErrDCADRE

```
!***********************************************************************
!                                   STANFORD LINEAR ACCELERATOR CENTER *
!     -----------------------------------------
      SUBROUTINE ErrDCADRE(IER,X1,X2,Fint,ERRDUM)
!     -----------------------------------------
!                                                     11 NOV 2005/1800 *
!***********************************************************************
!     Subprogram to output error messages from SUBROUTINE DCADRE
!***********************************************************************
      implicit none

      common/passit/BigL1,BigL2,Qval,iThkShlMax,Isorc
      real*8 BigL1,BigL2,Qval
      integer iThkShlMax,Isorc

      real*8
     + X1,X2,Fint(5),ERRDUM
      integer
     + IER,N,Ipass1,Ipass2

      data Ipass1/1/
      data Ipass2/1/

      if (IER.eq.65) then
        N=IER - 64
        write(6,*) ''
        write(6,*) 'DCADRE warning: IER=',IER,' --> N=',N
        write(6,*) ''
        if (Ipass1.eq.1) then
           write(6,*) 'N = 1 IMPLIES THAT ONE OR MORE SINGULARITIES'
          write(6,*) '      WERE SUCCESSFULLY HANDLED.'
          write(6,*) ''
          Ipass1=0
```

```
  end if
  write(6,*) 'X1    =',X1
  write(6,*) 'X2    =',X2
  write(6,*) 'Fint  =',Fint(Isorc),' (Isorc=',Isorc,')'
  write(6,*) 'ERRDUM=',ERRDUM
else if (IER.eq.66) then
  N=IER - 64
  write(6,*) ''
  write(6,*) 'DCADRE warning: IER=',IER,' --> N=',N
  write(6,*) ''
  if (Ipass2.eq.1) then
    write(6,*) 'N = 2 IMPLIES THAT, IN SOME SUBINTERVAL(S),'
    write(6,*) '      THE ESTIMATE OF THE INTEGRAL WAS'
    write(6,*) '      ACCEPTED MERELY BECAUSE THE ESTIMATED'
    write(6,*) '      ERROR WAS SMALL, EVEN THOUGH NO REGULAR'
    write(6,*) '      BEHAVIOR WAS RECOGNIZED.'
    write(6,*) ''
    Ipass2=0
  end if
  write(6,*) 'X1    =',X1
  write(6,*) 'X2    =',X2
  write(6,*) 'Fint  =',Fint(Isorc),' (Isorc=',Isorc,')'
  write(6,*) 'ERRDUM=',ERRDUM
else if (IER.eq.131) then
  N=IER - 128
  write(6,*) ''
  write(6,*) 'DCADRE fatal error: IER=',IER,' --> N=',N
  write(6,*) ''
  write(6,*) 'N = 3 FAILURE DUE TO INSUFFICIENT INTERNAL'
  write(6,*) '      WORKING STORAGE.'
  write(6,*) ''
  write(6,*) 'X1    =',X1
  write(6,*) 'X2    =',X2
  write(6,*) 'Fint  =',Fint(Isorc),' (Isorc=',Isorc,')'
  write(6,*) 'ERRDUM=',ERRDUM
  write(6,*) ''
  write(6,*) 'Program stopped'
  stop
else if (IER.eq.132) then
  N=IER - 128
  write(6,*) ''
  write(6,*) 'DCADRE fatal error: IER=',IER,' --> N=',N
  write(6,*) ''
  write(6,*) 'N = 4 FAILURE.  THIS MAY BE DUE TO TOO MUCH'
  write(6,*) '      NOISE IN THE FUNCTION (RELATIVE TO'
  write(6,*) '      THE GIVEN ERROR REQUIREMENTS) OR DUE'
  write(6,*) '      TO AN ILL-BEHAVED INTEGRAND.'
  write(6,*) ''
  write(6,*) 'X1    =',X1
  write(6,*) 'X2    =',X2
  write(6,*) 'Fint  =',Fint(Isorc),' (Isorc=',Isorc,')'
  write(6,*) 'ERRDUM=',ERRDUM
  write(6,*) ''
  write(6,*) 'Program stopped'
  stop
else if (IER.eq.133) then
  N=IER - 128
  write(6,*) ''
  write(6,*) 'DCADRE fatal error: IER=',IER,' --> N=',N
  write(6,*) ''
  write(6,*) 'N = 5 INDICATES THAT RERR IS GREATER THAN'
  write(6,*) '    0.1, OR RERR IS LESS THAN 0.0, OR'
  write(6,*) '    RERR IS TOO SMALL FOR THE PRECISION'
  write(6,*) '    OF THE MACHINE.'
  write(6,*) ''
  write(6,*) 'X1    =',X1
  write(6,*) 'X2    =',X2
  write(6,*) 'Fint  =',Fint(Isorc),' (Isorc=',Isorc,')'
  write(6,*) 'ERRDUM=',ERRDUM
  write(6,*) ''
```

```
        write(6,*) 'Program stopped'
        stop
      else
        write(6,*) ''
        write(6,*) 'Stopped in ErrDCADRE with bad IER=',IER
        stop
      end if

      return
      end

!-------------------last line of SUBROUTINE ErrDCADRE-------------------
```

A.7 SUBROUTINE DCADRE

```
!***********************************************************************
!                                   STANFORD LINEAR ACCELERATOR CENTER *
!     -----------------------------------------------------------
      DOUBLE PRECISION FUNCTION DCADRE(F,A,B,AERR,RERR,ERROR,IER)
!     -----------------------------------------------------------
!                                   Version:         11 NOV 2005/1800 *
!***********************************************************************
!     DCADRE is a double precision version of CADRE, a quadrature
!     routine that has been available for many years at SLAC as part
!     of the library package.  CADRE was converted to double precision
!     by W. R. Nelson on October 8, 1974.
!***********************************************************************
! -------------------------------------------------------------------
! -DCADRE--------D-------LIBRARY 1----------------------------------
! -------------------------------------------------------------------
!
! FUNCTION:           - INTEGRATE F(X) FROM A TO B, USING CAUTIOUS
!                       ADAPTIVE ROMBERG EXTRAPOLATION.
!
! USAGE:              - FUNCTION DCADRE(F,A,B,AERR,RERR,ERROR,IER)
!
! PARAMETERS: DCADRE - ESTIMATE OF THE INTEGRAL OF F(X) FROM A TO B.
!
!             F       - A SINGLE-ARGUMENT REAL FUNCTION SUBPROGRAM
!                       SUPPLIED BY THE USER.  F MUST BE DECLARED
!                       EXTERNAL IN THE CALLING PROGRAM.
!
!             A,B     - THE TWO ENDPOINTS OF THE INTERVAL OF
!                       INTEGRATION (INPUT).
!
!             AERR    - DESIRED ABSOLUTE ERROR IN THE ANSWER (INPUT).
!
!             RERR    - DESIRED RELATIVE ERROR IN THE ANSWER (INPUT).
!
!             ERROR   - ESTIMATED BOUND ON THE ABSOLUTE ERROR OF
!                       THE OUTPUT NUMBER, DCADRE.
!
!             IER     - ERROR PARAMETER
!
!                       WARNING ERROR(WITH FIX) = 64 + N
!
!                         N = 1 IMPLIES THAT ONE OR MORE SINGULAR-
!                               ITIES WERE SUCCESSFULLY HANDLED.
!
!                         N = 2 IMPLIES THAT, IN SOME SUBINTERVAL(S),
!                               THE ESTIMATE OF THE INTEGRAL WAS
!                               ACCEPTED MERELY BECAUSE THE ESTIMATED
!                               ERROR WAS SMALL, EVEN THOUGH NO REG-
!                               ULAR BEHAVIOR WAS RECOGNIZED.
!
!                       TERMINAL ERROR = 128 + N
!
!                         N = 3 FAILURE DUE TO INSUFFICIENT INTERNAL
```

```
!                                WORKING STORAGE.
!
!                          N = 4 FAILURE.  THIS MAY BE DUE TO TOO MUCH
!                                NOISE IN THE FUNCTION (RELATIVE TO
!                                THE GIVEN ERROR REQUIREMENTS) OR DUE
!                                TO AN ILL-BEHAVED INTEGRAND.
!
!                          N = 5 INDICATES THAT RERR IS GREATER THAN
!                                0.1, OR RERR IS LESS THAN 0.0, OR
!                                RERR IS TOO SMALL FOR THE PRECISION
!                                OF THE MACHINE.
!
!***********************************************************************
      implicit none

      real*8
     + T(10,10),R(10),AIT(10),DIF(10),RN(4),TS(2049),
     + BEGIN(30),FINIS(30),EST(30),
     + AITLOW,H2TOL,AITTOL,LENGTH,JUMPTL,ZERO,P1,HALF,ONE,
     + TWO,FOUR,FOURP5,TEN,HUN,CADRE,F,ERROR,A,B,
     + AERR,RERR,STEPMN,STEPNM,STAGE,CUREST,FNSIZE,
     + PREVER,BEG,FBEG,END,FEND,STEP,ASTEP,TABS,HOVN,
     + FN,SUM,SUMABS,ABSI,VINT,TABTLM,ERGL,ERGOAL,
     + ERRA,ERRR,FEXTRP,ERRER,DIFF,SING,FEXTM1,ALG4O2,
     + H2NXT,SINGNX,SLOPE,FBEG2,ALPHA,
     + ERRET,H2TFEX,FI

      integer
     + IBEGS(30),
     + IER,IBEG,ISTAGE,IEND,L,N,LM1,N2,ISTEP,I,II,III,ISTEP2,IT,NNLEFT,
     + MAXTS,MAXTBL,MXSTGE

      logical
     + REGLSV(30),
     + H2CONV,AITKEN,RIGHT,REGLAR

      data
     + AITLOW/1.1D0/,
     + H2TOL/0.15D0/,
     + AITTOL/0.1D0/,
     + JUMPTL/0.01D0/,
     + MAXTS/2049/,
     + MAXTBL/10/,
     + MXSTGE/30/

      data
     + RN(1)/0.7142005D0/,
     + RN(2)/0.3466282D0/,
     + RN(3)/0.843751D0/,
     + RN(4)/0.1263305D0/

      data
     + ZERO/0.0D0/,
     + P1/0.1D0/,
     + HALF/0.5D0/,
     + ONE/1.0D0/,
     + TWO/2.0D0/,
     + FOUR/4.0D0/,
     + FOURP5/4.5D0/,
     + TEN/10.0D0/,
     + HUN/100.0D0/

      ALG4O2=DLOG10(TWO)
      CADRE=ZERO
      ERROR=ZERO
      CUREST=ZERO
      VINT=ZERO
      IER=0
      LENGTH=DABS(B-A)
```

```
      if (LENGTH.eq.ZERO) GO TO 215
      if (RERR.gt.P1 .or. RERR.lt.ZERO) GO TO 210
      if (AERR.eq.ZERO .and. (RERR+HUN).le.HUN) GO TO 210

      ERRR=RERR
      ERRA=DABS(AERR)
      STEPMN=(LENGTH/FLOAT(2**MXSTGE))
      STEPNM=DMAX1(LENGTH,DABS(A),DABS(B))*TEN
      STAGE=HALF
      ISTAGE=1
      FNSIZE=ZERO
      PREVER=ZERO
      REGLAR=.false.

!     -----------------------------------------------------------------
!     The given interval of integration is the first interval considered
!     -----------------------------------------------------------------
      BEG=A
      FBEG=F(BEG)*HALF
      TS(1)=FBEG
      IBEG=1
      END=B
      FEND=F(END)*HALF
      TS(2)=FEND
      IEND=2
5     RIGHT=.false.

!     -------------------------------------------------------------
!     Investigation of a particular subinterval begins at this point
!     -------------------------------------------------------------
10    STEP=END - BEG
      ASTEP=DABS(STEP)

      if (ASTEP.lt.STEPMN) GO TO 205
      if (STEPNM+ASTEP.eq.STEPNM) GO TO 205

      T(1,1)=FBEG + FEND
      TABS=DABS(FBEG) + DABS(FEND)
      L=1
      N=1
      H2CONV=.false.
      AITKEN=.false.
15    LM1=L
      L=L + 1

!     ----------------------------------------------------------
!     Calculate the next trapezoid sum, T(L,1), which is based on
!     N2 + 1 equispaced points. Here, N2=N*2=2**(L-1).
!     ----------------------------------------------------------
      N2=N + N
      FN=N2
      ISTEP=(IEND - IBEG)/N

      if (ISTEP.gt.1) GO TO 25

      II=IEND
      IEND=IEND + N

      if (IEND.gt.MAXTS) GO TO 200

      HOVN=STEP/FN
      III=IEND
      FI=ONE

      do I=1,N2,2
        TS(III)=TS(II)
        TS(III-1)=F(END - FI * HOVN)
        FI=FI+TWO
        III=III-2
        II=II-1
```

```
      end do

      ISTEP=2
25    ISTEP2=IBEG + ISTEP/2
      SUM=ZERO
      SUMABS=ZERO

      do I=ISTEP2,IEND,ISTEP
        SUM=SUM + TS(I)
        SUMABS=SUMABS + DABS(TS(I))
      end do

      T(L,1)=T(L-1,1)*HALF+SUM/FN
      TABS=TABS*HALF+SUMABS/FN
      ABSI=ASTEP*TABS
      N=N2

!     -----------------------------------------------------------------
!     Get preliminary value for VINT from last trapezoid sum and update
!     the error requirement ERGOAL for this subinterval.
!     -----------------------------------------------------------------
      IT=1
      VINT=STEP*T(L,1)
      TABTLM=TABS*TEN
      FNSIZE=DMAX1(FNSIZE,DABS(T(L,1)))
      ERGL=ASTEP*FNSIZE*TEN
      ERGOAL=STAGE*DMAX1(ERRA,ERRR*DABS(CUREST+VINT))

!     --------------------------------------
!     Complete row L and column L of T Array.
!     --------------------------------------
      FEXTRP=ONE

      do I=1,LM1
        FEXTRP=FEXTRP*FOUR
        T(I,L)=T(L,I) - T(L-1,I)
        T(L,I+1)=T(L,I) + T(I,L)/(FEXTRP-ONE)
      end do

!     ----------------------------------------------------------
!     Preliminary decision procedure.  If L=2 and T(2,1)=T(1,1),
!     GO TO 135 to follow up the impression that intergrand is
!     straight line.
!     ----------------------------------------------------------
      ERRER=ASTEP*DABS(T(1,L))

      if (L.gt.2) GO TO 40
      if (TABS+P1*DABS(T(1,2)).eq.TABS) GO TO 135
      GO TO 15

!     ----------------------------------------------------------------
!     Calculate next ratios for columns 1,...,L-2 of T-table.
!     ratio is set to zero if difference in last two entries of column
!     is about zero.
!     ----------------------------------------------------------------
40    do I=2,LM1
        DIFF=ZERO
        if (TABTLM+DABS(T(I-1,L)).ne.TABTLM) DIFF=T(I-1,LM1)/T(I-1,L)
        T(I-1,LM1)=DIFF
      end do

      if (DABS(FOUR-T(1,LM1)).le.H2TOL) GO TO 60
      if (T(1,LM1).eq.ZERO) GO TO 55
      if (DABS(TWO-DABS(T(1,LM1))).lt.JUMPTL) GO TO 130
      if (L.eq.3) GO TO 15

      H2CONV=.false.

      if (DABS((T(1,LM1)-T(1,L-2))/T(1,LM1)).le.AITTOL) GO TO 75
50    if (REGLAR) GO TO 55
```

```
      if (L.eq.4) GO TO 15
55    if (ERRER.gt.ERGOAL .and. (ERGL+ERRER).ne.ERGL) GO TO 175
      GO TO 145

!     ------------------------------
!     Cautious ROMBERG extrapolation
!     ------------------------------
60    if (H2CONV) GO TO 65

      AITKEN=.false.
      H2CONV=.true.
65    FEXTRP=FOUR
70    IT=IT + 1
      VINT=STEP*T(L,IT)
      ERRER=DABS(STEP/(FEXTRP-ONE)*T(IT-1,L))

      if (ERRER.le.ERGOAL) GO TO 160
      if (ERGL+ERRER.eq.ERGL) GO TO 160
      if (IT.eq.LM1) GO TO 125
      if (T(IT,LM1).eq.ZERO) GO TO 70
      if (T(IT,LM1).le.FEXTRP) GO TO 125
      if (DABS(T(IT,LM1)/FOUR-FEXTRP)/FEXTRP.lt.AITTOL) then
        FEXTRP=FEXTRP*FOUR
      end if
      GO TO 70

!     ----------------------------------------------------------------
!     Integrand may have X**ALPHA type singularity resulting in a ratio
!     of SING=2**(ALPHA + 1)
!     ----------------------------------------------------------------
75    if (T(1,LM1).lt.AITLOW) GO TO 175
      if (AITKEN) GO TO 80

      H2CONV=.false.
      AITKEN=.true.
80    FEXTRP=T(L-2,LM1)

      if (FEXTRP.gt.FOURP5) GO TO 65
      if (FEXTRP.lt.AITLOW) GO TO 175
      if (DABS(FEXTRP-T(L-3,LM1))/T(1,LM1).gt.H2TOL) GO TO 175

      SING=FEXTRP
      FEXTM1=ONE/(FEXTRP - ONE)
      AIT(1)=ZERO

      do I=2,L
        AIT(I)=T(I,1) + (T(I,1)-T(I-1,1))*FEXTM1
        R(I)=T(1,I-1)
        DIF(I)=AIT(I) - AIT(I-1)
      end do

      IT=2
90    VINT=STEP*AIT(L)
      ERRER=ERRER*FEXTM1

      if (ERRER.gt.ERGOAL .and. (ERGL+ERRER).ne.ERGL) GO TO 95

      ALPHA=DLOG10(SING)/ALG4O2 - ONE
      IER=MAX0(IER,65)
      GO TO 160

95    IT=IT + 1

      if (IT.eq.LM1) GO TO 125
      if (IT.gt.3) GO TO 100

      H2NXT=FOUR
      SINGNX=SING+SING

100   if (H2NXT.lt.SINGNX) GO TO 105
```

```
      FEXTRP=SINGNX
      SINGNX=SINGNX+SINGNX
      GO TO 110

105   FEXTRP=H2NXT
      H2NXT=FOUR*H2NXT

110   do I=IT,LM1
        R(I+1)=ZERO
        if (TABTLM+DABS(DIF(I+1)).ne.TABTLM) R(I+1)=DIF(I)/DIF(I+1)
      end do

      H2TFEX=-H2TOL*FEXTRP

      if (R(L)-FEXTRP.lt.H2TFEX) GO TO 125
      if (R(L-1)-FEXTRP.lt.H2TFEX) GO TO 125

      ERRER=ASTEP*DABS(DIF(L))
      FEXTM1=ONE/(FEXTRP - ONE)

      do I=IT,L
        AIT(I)=AIT(I) + DIF(I)*FEXTM1
        DIF(I)=AIT(I) - AIT(I-1)
      end do
      GO TO 90

!     -------------------------------------------------------------------
!     Current trapezoid sum and resulting extrapolated values did not
!     give a small enough ERRER.  Note -- Having PREVER.LT.ERRER
!     is an almost certain sign of beginning trouble with in the
!     function values.  Hence, a watch for, and control of, noise should
!     begin here.
!     -------------------------------------------------------------------
125   FEXTRP=DMAX1(PREVER/ERRER,AITLOW)
      PREVER=ERRER

      if (L.lt.5) GO TO 15
      if (L-IT.gt.2 .and. ISTAGE.lt.MXSTGE) GO TO 170

      ERRET=ERRER/(FEXTRP**(MAXTBL-L))

      if (ERRET.gt.ERGOAL .and. (ERGL+ERRET).ne.ERGL) GO TO 170
      GO TO 15

!     ------------------------------
!     Integrand has jump (see NOTES)
!     ------------------------------
130   if (ERRER.gt.ERGOAL .and. (ERGL+ERRER).ne.ERGL) GO TO 170

!     -------------------
!     Note that 2*FN=2**L
!     -------------------
      DIFF=DABS(T(1,L))*(FN+FN)
      GO TO 160

!     -------------------------------------------------------------------
!     Integrand is straight line.  Test this assumption by comparing the
!     value of the integrand at four "randomly chosen" points with the
!     value of the straight line interpolating the integrand at the two
!     end points of the subinterval.  If test is passed, accept VINT.
!     -------------------------------------------------------------------
135   SLOPE=(FEND-FBEG)*TWO
      FBEG2=FBEG+FBEG

      do I=1,4
        DIFF=DABS(F(BEG+RN(I)*STEP) - FBEG2-RN(I)*SLOPE)
        if (TABTLM+DIFF.ne.TABTLM) GO TO 155
      end do
      GO TO 160
```

```
!     ---------------------------------------------------------------
!     Noise may be dominant feature.  Estimate noise level by comparing
!     the value of the integrand at four "randomly chosen" points with
!     the value of the straight line interpolating the integrand at the
!     two endpoints.  If small enough, accept VINT.
!     ---------------------------------------------------------------
145   SLOPE=(FEND-FBEG)*TWO
      FBEG2=FBEG+FBEG
      I=1
150   DIFF=DABS(F(BEG+RN(I)*STEP) - FBEG2-RN(I)*SLOPE)
155   ERRER=DMAX1(ERRER,ASTEP*DIFF)

      if (ERRER.gt.ERGOAL .and. (ERGL+ERRER).ne.ERGL) GO TO 175

      I=I+1
      if (I.le.4) GO TO 150

      IER=66

!     --------------------------------------------------------------
!     Integration ovnder current subinterval successful.  Add VINT to
!     DCADRE and ERRER to ERROR, then set up next subinterval, if any.
!     --------------------------------------------------------------
160   CADRE=CADRE + VINT
      ERROR=ERROR + ERRER

      if (RIGHT) GO TO 165

      ISTAGE=ISTAGE - 1

      if (ISTAGE.eq.0) GO TO 220

      REGLAR=REGLSV(ISTAGE)
      BEG=BEGIN(ISTAGE)
      END=FINIS(ISTAGE)
      CUREST=CUREST - EST(ISTAGE+1) + VINT
      IEND=IBEG - 1
      FEND=TS(IEND)
      IBEG=IBEGS(ISTAGE)
      GO TO 180

165   CUREST=CUREST + VINT
      STAGE=STAGE+STAGE
      IEND=IBEG
      IBEG=IBEGS(ISTAGE)
      END=BEG
      BEG=BEGIN(ISTAGE)
      FEND=FBEG
      FBEG=TS(IBEG)
      GO TO 5

!     -----------------------------------------------------------
!     Integration over current subinterval is unsuccessful. Mark
!     subinterval for further subdivision. Set up next subinterval.
!     -----------------------------------------------------------
170   REGLAR=.true.

175   if (ISTAGE.eq.MXSTGE) GO TO 205
      if (RIGHT) GO TO 185

      REGLSV(ISTAGE+1)=REGLAR
      BEGIN(ISTAGE)=BEG
      IBEGS(ISTAGE)=IBEG
      STAGE=STAGE*HALF
180   RIGHT=.true.
      BEG=(BEG+END)*HALF
      IBEG=(IBEG+IEND)/2
      TS(IBEG)=TS(IBEG)*HALF
      FBEG=TS(IBEG)
```

```
      GO TO 10

185   NNLEFT=IBEG - IBEGS(ISTAGE)

      if (IEND+NNLEFT.ge.MAXTS) GO TO 200

      III=IBEGS(ISTAGE)
      II=IEND

      do I=III,IBEG
        II=II + 1
        TS(II)=TS(I)
      end do

      do I=IBEG,II
        TS(III)=TS(I)
        III=III + 1
      end do

      IEND=IEND + 1
      IBEG=IEND - NNLEFT
      FEND=FBEG
      FBEG=TS(IBEG)
      FINIS(ISTAGE)=END
      END=BEG
      BEG=BEGIN(ISTAGE)
      BEGIN(ISTAGE)=END
      REGLSV(ISTAGE)=REGLAR
      ISTAGE=ISTAGE + 1
      REGLAR=REGLSV(ISTAGE)
      EST(ISTAGE)=VINT
      CUREST=CUREST + EST(ISTAGE)
      GO TO 5

!     -------------------------------------------
!     Failure to handle given integration problem.
!     -------------------------------------------
200   IER=131
      GO TO 215

205   IER=132
      GO TO 215

210   IER=133
215   CADRE=CUREST + VINT

220   DCADRE=CADRE

      return
      end

!---------------------last line of FUNCTION DCADRE----------------------
```

A.8 Required PARAMETER (header) file (in subdirectory include/)

```
!---------------------------shield11_h.fh-----------------------------
! Version: 050117-0200                    COMMON used in shield11.f code
!----------------------------------------------------------------------
!23456789|123456789|123456789|123456789|123456789|123456789|123456789|12

! ------------------------------------------
! SHIELD11 "header" file containing PARAMETERS
! ------------------------------------------

! Maximum number of materials
      integer MXMAT
      parameter (MXMAT = 4)
```

```
! Maximum number of source types (i.e., like GamD)
      integer MXSOR
      parameter (MXSOR = 5)

!-----------------------last line of shield11_h.fh---------------------
```

A.9 Required COMMON files (in subdirectory include/)

```
!--------------------------------beam.fh-------------------------------
! Version: 050117-0200                    COMMON used in shield11.f code
!----------------------------------------------------------------------
!23456789|123456789|123456789|123456789|123456789|123456789|123456789|12

      common/BEAM/                                             ! Beam data
     * Ebeam                                      ! Energy of beam (GeV)

      real*8 Ebeam

!-------------------------last line of beam.fh-------------------------

!-------------------------------detect.fh------------------------------
! Version: 050117-0200                    COMMON used in shield11.f code
!----------------------------------------------------------------------
!23456789|123456789|123456789|123456789|123456789|123456789|123456789|12

      common/DETECT/                                       ! Detector data
     * Theta0,                                  ! Starting angle (degrees)
     * dTheta,                              ! Angular increments (degrees)
     * Ntheta                                           ! Number of angles

      real*8 Theta0,dTheta
      integer Ntheta

!-------------------------last line of detect.fh-----------------------

!-------------------------------dosval.fh------------------------------
! Version: 050117-0200                    COMMON used in shield11.f code
!----------------------------------------------------------------------
!23456789|123456789|123456789|123456789|123456789|123456789|123456789|12

      common/DOSVAL/                                    ! Dose information
     * Dose(MXSOR)                        ! Dose for each source component

      real*8 Dose

!-------------------------last line of dosval.fh-----------------------

!-------------------------------matdta.fh------------------------------
! Version: 050117-0200                    COMMON used in shield11.f code
!----------------------------------------------------------------------
!23456789|123456789|123456789|123456789|123456789|123456789|123456789|12

      common/MATDTA/                                       ! Material data
     * Xmfp(MXMAT,MXSOR),                       ! Mean free path (g/sq.cm)
     * Zmat(MXMAT),                                        ! Atomic number
     * Amat(MXMAT),                                ! Atomic mass (g/mole)
     * RHOmat(MXMAT),                                 ! Density (g/cu.cm)
     * RLmat(MXMAT),                          ! Radiation length (g/sq.cm)
     * XMUmat(MXMAT),                           ! Moliere length (g/sq.cm)
     * MatID(MXMAT),                                 ! Material identifier
     * SorNam(MXSOR)                                         ! Source name

      real*8 Xmfp,Zmat,Amat,RHOmat,RLmat,XMUmat
      character*4 MatID,SorNam

!-------------------------last line of matdta.fh-----------------------
```

```
!--------------------------------miscon.fh-----------------------------
! Version: 050117-0200                    COMMON used in shield11.f code
!----------------------------------------------------------------------
!23456789|123456789|123456789|123456789|123456789|123456789|123456789|12

      common/MISCON/                            ! Miscellaneous constants
     * DegRad                                   ! Pi/180 (radians/degree)

      real*8 DegRad

!-------------------------last line of miscon.fh------------------------

!-------------------------------shield.fh------------------------------
! Version: 050117-0200                    COMMON used in shield11.f code
!----------------------------------------------------------------------
!23456789|123456789|123456789|123456789|123456789|123456789|123456789|12

      common/SHIELD/                                        ! Shield data
     * AngShl,                     ! Angle of primary shield (degrees)
     * DisShl,              ! Distance of primary shield (inches-->cm)
     * ThkShl,             ! Thickness of primary shield (inches-->cm)
     * AddCON,             ! Added thickness of concrete (inches-->cm)
     * AddFE,                  ! Added thickness of iron (inches-->cm)
     * AddPB,                  ! Added thickness of lead (inches-->cm)
     * AddMIS,                 ! Added thickness of MISC (inches-->cm)
     * IDshl

      real*8 AngShl,DisShl,ThkShl,AddCON,AddFE,AddPB,AddMIS
      integer IDshl

!-------------------------last line of shield.fh------------------------

!--------------------------------sorcon.fh-----------------------------
! Version: 050117-0200                    COMMON used in shield11.f code
!----------------------------------------------------------------------
!23456789|123456789|123456789|123456789|123456789|123456789|123456789|12

      common/SORCON/                            ! FUNCTION Sorc constants
     * E_HEN,                                             ! Energy (GeV)
     * CS_HEN,                          ! Cross section multiplying factor
     * ThrHEN,              ! Threshold energy (GeV) for HEN production
     * TarLenG,              ! Minimum target length for photons (r.l.)
     * TarLenN,             ! Minimum target length for neutrons (r.l.)
     * TarRadG,                     ! Minimum target radius for photons
     * TarRadN         ! Minimum target radius for neutrons (Mol.units)

      real*8 E_HEN(23),CS_HEN(23),ThrHEN,TarLenG,TarLenN,TarRadG,TarRadN

!-------------------------last line of sorcon.fh------------------------

!-------------------------------target.fh------------------------------
! Version: 050117-0200                    COMMON used in shield11.f code
!----------------------------------------------------------------------
!23456789|123456789|123456789|123456789|123456789|123456789|123456789|12

      common/TARGET/                                        ! Target data
     * TarLen,                        ! Length of target (inches-->cm)
     * TarRAD,                        ! Radius of target (inches-->cm)
     * IDtar,                                               ! Target ID
     * IattSW                ! Switch for neutron attenuation by target

      real*8 TarLen,TarRad
      integer IDtar,IattSW

!-------------------------last line of target.fh------------------------
```