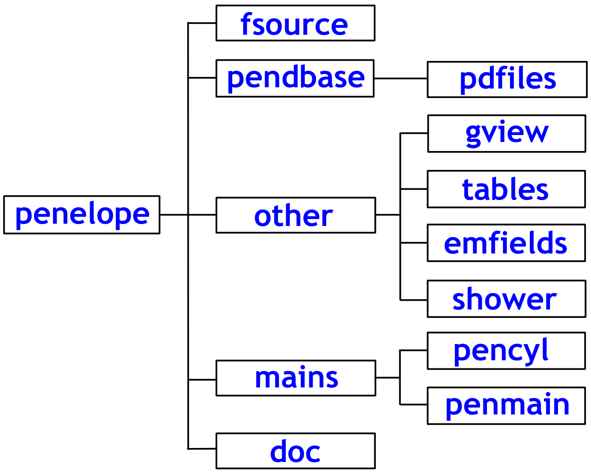
Tutorial for PENELOPE (version 2014)

The distribution package has the following directory structure and contents:



\fsource Fortran source files of the PENELOPE code system:

penelope.f … Transport/physics simulation routines.

rita.f ... Routines for random sampling from single-variate distributions.

pengeom.f … Quadric geometry package.

penvared.f … Variance-reduction routines.

material.f … The main program for creating cross-section data files.

timer.f … Portable timing routines.

\pendbase Files necessary for creating material cross-section data files. The PENELOPE data-base of interaction cross-section data is contained in subdirectory **\pdfiles** (do not remove or alter any of the files in directory **\pdfiles**). The executable binaries **material.exe**, **tables.exe**, and **shower.exe** must be placed in directory **\pendbase**.

\other Additional software for quadric geometry visualisation (**\gview**), for displaying particle tracks on the screen of the computer (**\shower**), for generating and displaying tables of interaction cross sections and other transport properties (**\tables**), and a routine package for simulating radiation transport in static electromagnetic fields (**\emfields**).

\mains This directory contains two subdirectories with the steering main programs **pencyl** and **penmain**. Several **examples** are provided for each main program, with the corresponding sets of input, geometry, and material data files in separate subdirectories. Gnuplot scripts for visualisation of simulation results from the example main programs are also included in the subdirectories **\gscripts**.

\doc Documentation of the code system. It includes the following files:

tutorial.pdf … This file.

penelope-2014-NEA.pdf … Official release by the OECD Nuclear Energy Agency Data Bank of PENELOPE documentation, distributed together with version 2014. This is the reference to be used in any publication. Cite it as:

> F. Salvat, "*PENELOPE–2014: A Code System for Monte Carlo Simulation of Electron and Photon Transport*" (OECD Nuclear Energy Agency, Issy-les-Moulineaux, France, 2015).

Have this document at hand in the initial stages when using PENELOPE.

COMPILER AND PLOTTER

- To generate the executable binary files of your simulation programs you need a Fortran 90 compiler. If you do not have one installed on your computer, you may use the GNU Fortran compiler (**gfortran**) from the Free Software Foundation. The GNU Fortran compilers for Win32 or Win64 can be downloaded from the site http://mingw-w64.sourceforge.net .
- To plot the results of the example main programs you will need a plotting program. We recommend using gnuplot; the Windows version can be downloaded from http://www.gnuplot.info (gnuplot is also available in most LINUX distributions). This program is not part of PENELOPE. On Windows, when gnuplot is properly installed, the command "**wgnuplot script.gnu**" executes the script file **script.gnu**. If files with the extension "**.gnu**" are associated with **wgnuplot**, the script can be run directly from the explorer window by double-clicking on its icon.

EXERCISES

(It is assumed you will be working on a "command" window)

1. Copy the (uncompressed) simulation package to your hard disk. Keep the directory structure of the **penelope.zip** file unaltered.

1a. Create a working directory **.\work\compile**. Copy the contents of the directory **.\penelope\fsource** into **.\work\compile**.

2. Run material

2a. Start compiling and linking[1] the codes **material.f**, **penelope.f**, and **rita.f** (directory **.\work\compile**),

```
> gfortran -Os -Wall material.f -o material.exe
```

The command switch "**-Os**" optimises code generation and "**-Wall**" issues compilation warning messages; with the option "**-o**" we can specify the name of the produced executable file (the extension "**.exe**" is automatically appended).

NOTE: To simplify the typing of compilation commands, all the subroutine packages used by a main program have been declared through **include** statements inside the main program (see, for example, the source file **material.f**) and do not have to be listed in the compilation command.

2b. Move **material.exe** to the directory **.\penelope\pendbase** (**material.exe** must be in the parent directory of **.\pdfiles**). Execute **material.exe** to create data for "copper", for "sodium iodide" and for "water" (ID nos. 29, 253 and 278 in the **material-list.txt** file). Call the output files **Cu.mat**, **NaI.mat**, and **H2O.mat**, respectively.

3. Run tables

3a. Copy the file **tables.f** (from directory **.\penelope\other\tables)** into directory **.\work\compile**. Compile and link **tables.f**, **penelope.f**, and **rita.f**,

```
> gfortran -Os -Wall tables.f -o tables.exe
```

3b. Move the file **tables.exe** to the directory **.\penelope\pendbase**. Copy the files ***.gnu** in directory **.\penelope\other\tables** into directory **.\penelope\pendbase**.

3c. Execute the code **tables.exe** for the materials generated in step 2b.

3d. Inspect the file **tables.dat**, which contains information on the material composition, atomic relaxation data, and extensive tables of radiation transport properties (cross sections, mean free paths, stopping powers, etc).

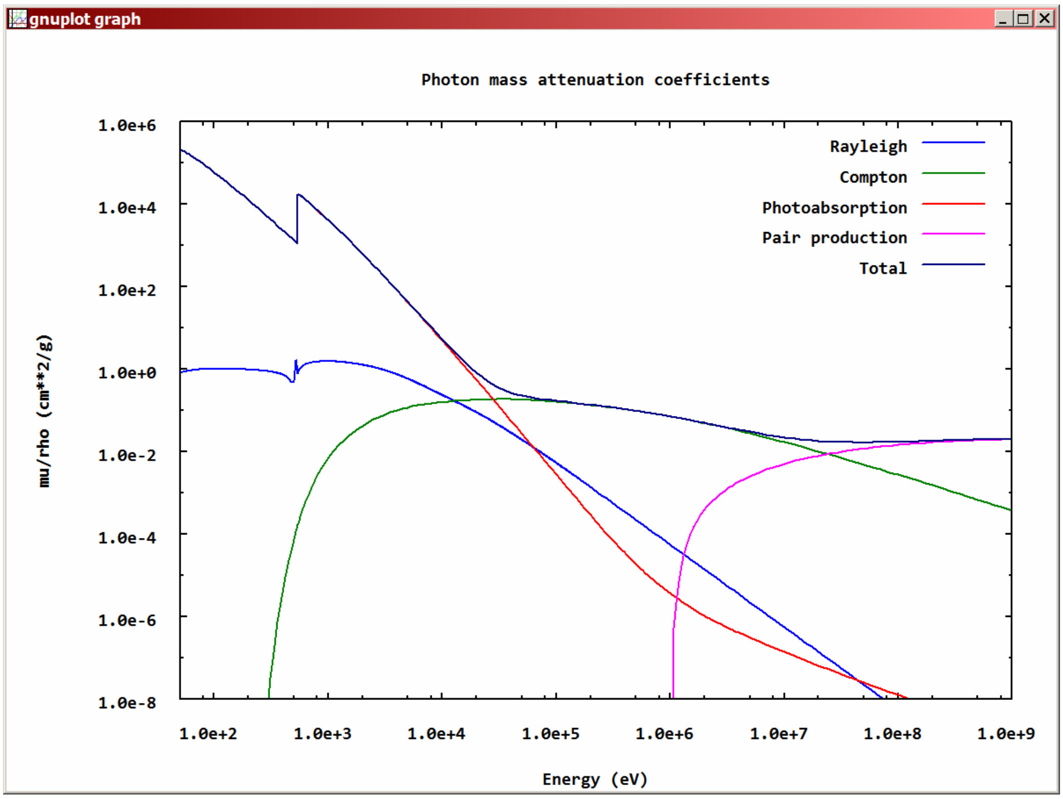
[1] The examples given in this tutorial are for the GNU Fortran compiler **gfortran**. If instead you use, for example, Compaq Visual Fortran 6.5, the equivalent instruction to the compiler is:

```
> DF material.f
```

3e. The code **tables.exe** also generates tables of energy-dependent quantities in separate ASCII files (with the extension "**.tab**"). To visualise the tables in the various files with gnuplot, type the command

> **wgnuplot fname.gnu** …where the file name indicates the associated output file or files.

The following is an example of gnuplot output window (photon mass attenuation coefficients for liquid water).

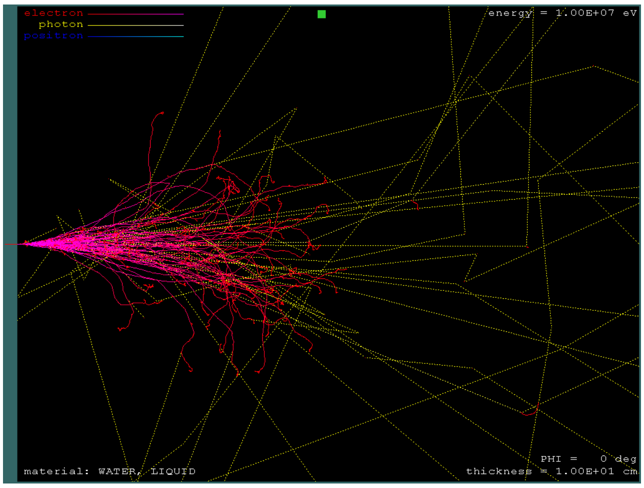


4. Run shower

Note that **shower.exe** *runs under Microsoft® Windows® only*

4a. Copy **shower.exe** from **.\penelope\other\shower** to the directory **.\penelope\pendbase**.

4b. Change directory to **.\penelope\pendbase**. Execute the program by typing **shower**, or by clicking on the **shower** icon in the explorer window.



The example shown here corresponds to:

`278` for material ID number (water, liquid)

`1` for primary particle (electrons)

`1E7` for initial energy (eV)

`1E4` for absorption energy of electrons

`1E4` for absorption energy of photons

`10` for slab thickness (cm)

`50` for particles in the bunch

NOTE: When the initial energy is entered with reversed sign, the code uses default values of the absorption energies; this minimises the amount of information that has to be typed.

5. Run pencyl

5a. Copy the file **pencyl.f** from directory **.\penelope\mains\pencyl** into directory **.\work\compile**.

5b. Compile and link the code **pencyl.f** (see the NOTE in 2a),

```
> gfortran -Os pencyl.f -o pencyl.exe
```

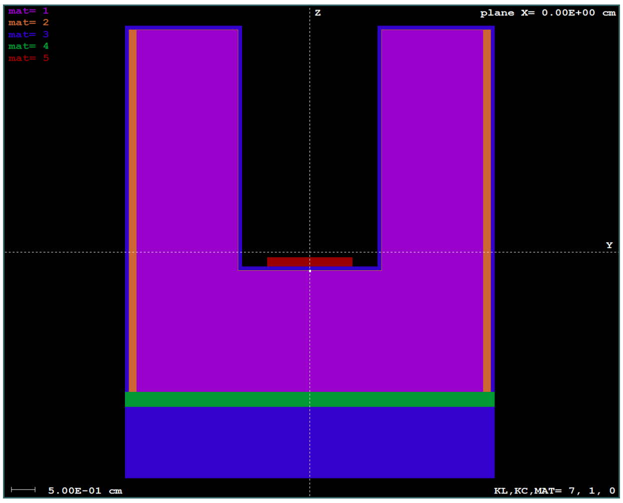
5c. Create a working directory **.\work\pencyl** and copy the directory **.\penelope\mains\pencyl \examples** to that working directory. Note that the new subdirectory **.\work\pencyl\examples** has four sub-directories, which contain the data files for four simulation exercises:
1-disc, an electron beam impinging on a Cu cylinder,
2-beta-source, a ^{32}P beta source in a water slab, and
3-detector, NaI scintillation detectors.
4-aba, ^{60}Co gamma rays on a water-Al-water phantom.

5d. To run the exercise **1-disc**, copy the file **pencyl .exe** from the directory **.\work\compile** to the directory **.\work\pencyl\examples\1-disc**.

5e. Inspect the file **Cu.mat** and verify that it is identical to the file of the same name that you created in **.\penelope\pendbase**.

5f. Inspect the input file **disc.in**, to see what is to be calculated. Details on the structure of this file, on the meaning of different keywords and on the geometry definition can be found in Chapter 7 of the PENELOPE manual (**penelope-2014-NEA.pdf**) or in the heading comments of the **pencyl.f** source file.

5g. The geometry viewer **gviewc.exe** (in directory **.\penelope\mains\pencyl**) displays a 2D view across the cylindrical geometry defined in the **pencyl** input file. Run **gviewc** and visualise the geometry in file **disc.in**. *Note:* **gviewc.exe** *works under Microsoft® Windows® only*
The example shown here corresponds to exercise **3-detector**:



welld.in (path+filename of the input file that contains the geometry definition)

0,0,0 (coordinates of the window centre)

1 (display mode)

Operation instructions for the **gviewc.exe** viewer can be displayed on the window by typing "**h**" or "**?**".

NOTE: gviewc.exe has two display modes. With mode 1, the program shows the materials present in the geometrical structure. In mode 2, the code displays bodies and body numbers, which are needed for scoring purposes.

5h. Execute **pencyl.exe**, using **disc.in** as input file, *i.e.*, type

```
> pencyl < disc.in
```

5i. Inspect the file **material.dat** (information on the materials read), **pencyl.dat** (transcript of data in the input files) and **pencyl-res.dat** (output file).

5j. There are other output files for plotting. To display the contents of each of these files on the screen, copy the gnuplot scripts ***.gnu** from directory **.\penelope\mains\pencyl\gscripts** to the working directory **.\examples\1-disc** and type

```
> wgnuplot file-name.gnu
```

5k. Run the other three exercises in subdirectory **.\work\pencyl\examples** (**2-beta-source**, **3-detector**, and **4-aba**) by following similar steps.

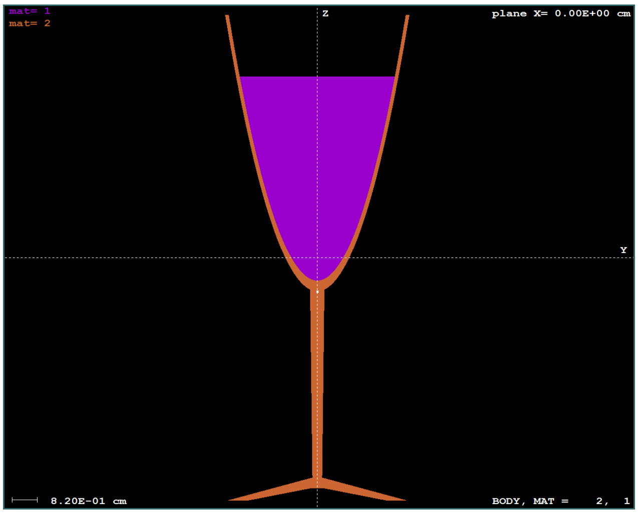
6. Run the quadric geometry viewers

Note that the viewers **gview2d.exe** *and* **gview3d.exe** *run under Microsoft® Windows® only*

Change directory to **.\penelope\other\gview**. From there, execute the program by typing its name, **gview2d.exe** or **gview3d.exe**, or by clicking on its icon in the explorer window.

6a. Inspect the example geometry files. Details on the quadric geometry package PENGEOM and on the structure and formats of the geometry definition file can be found in Chapter 6 of the PENELOPE manual (**penelope-2014-NEA.pdf**).

6b. The code **gview2d.exe** displays a 2D view across a geometry defined by quadric surfaces.



Operation instructions are displayed on the screen by typing "**h**" or "**?**" from the graphics window.

The example shown here corresponds to:

glass (path+name of the geometry definition file)

0,0,1.1 (coordinates of the window centre)

1 (display mode)

NOTE: gview2d.exe has two display modes. In mode 1, the program shows the materials present in the geometrical structure. In mode 2, the code displays individual bodies. The information shown in the lower right-hand corner of the window corresponds to the central body (the one at the intersection of the axes).

6c. **gview3d.exe** displays a 3D view of the geometry defined by quadric surfaces.

Note that 3D rendering is initially set to the lowest resolution (**9**, fast). The screenshots shown in the next page have been generated with the highest resolution (**1**, slow).

7. Run penmain

7a. Copy the files **penmain.f**, **pmcomms.f**, and **source.f** from directory **.\penelope\mains\penmain** into directory **.\work\ compile**.

7b. Compile and link the code **penmain.f** (see the NOTE in 2a),

```
> gfortran -Os penmain.f -o penmain.exe
```

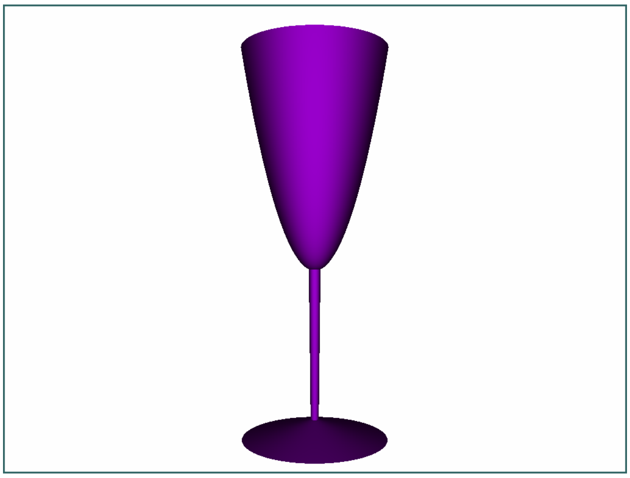
7c. Create a working directory **.\work\penmain** and copy the full directory **.\penelope\mains\penmain \examples** to that working directory. Note that the new directory **.\work\penmain\examples** has seven subdirectories, which contain the data files for seven simulation exercises:
1-disc, point electron source and a homogeneous Cu cylinder,
2-plane, an electron beam impinging on a semi-infinite water phantom with a plane surface,
3-detector, a cylindrical NaI scintillation detector with Fe backing,
4-x-ray-tube, a simple x-ray generator,
5-accelerator, a simple electron accelerator; note that this exercise involves two steps, and
6-polarisation, a scattering experiment with polarised photons.
7-aba, ^{60}Co gamma rays on a water-Al-water phantom.

7d. To run the exercise **1-disc**, copy the file **penmain.exe** from directory **.\work\compile** into the directory **.\work\penmain\examples\1-disc**.

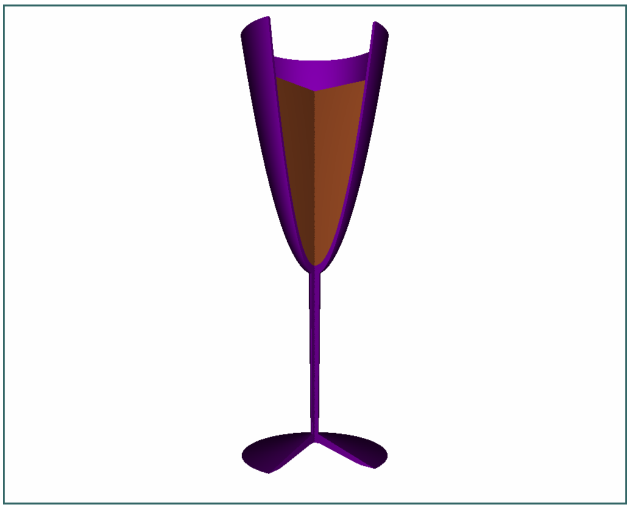
7e. Inspect the file **Cu.mat** and verify that it is identical to the file of the same name that you created in **.\penelope\pendbase**.

7f. Inspect the geometry definition file **disc.geo**, and visualise it using the viewers **gview2d.exe** and **gview3d.exe**.

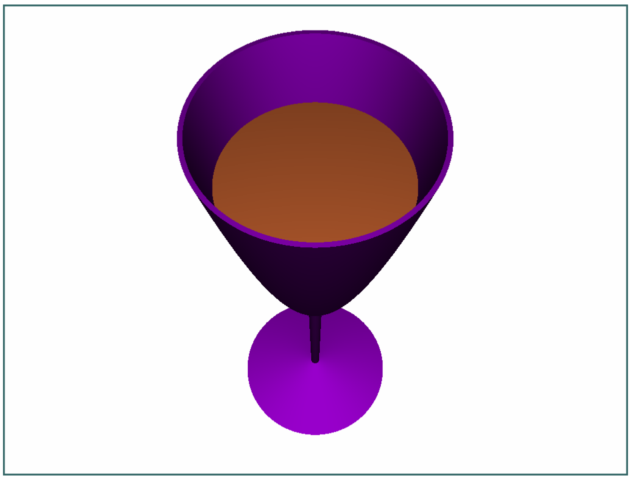
Screenshots from **gview3d** for two example geometries:



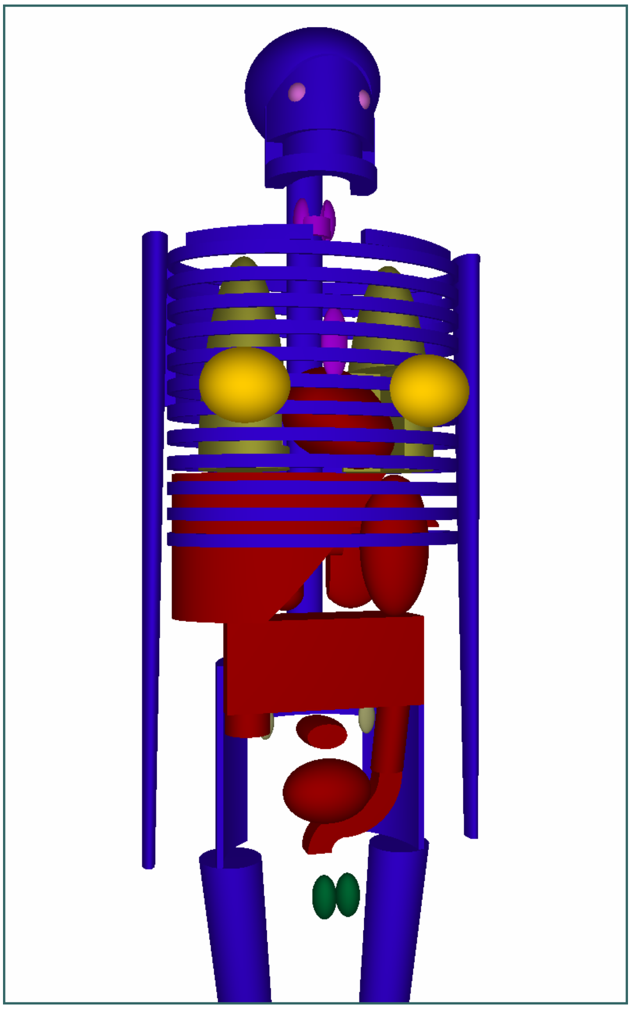
glass (path+name of geometry definition file)
10 (distance from the window to the object)
No (for excluding a sector; see below)



Note that a wedge can be excluded to show the interior of a geometry. For example, in this view the wedge between **-35,40** degrees has been excluded.



The geometry can be rotated. Type "**r**" and enter the Euler angles, for example: **0,60,0** degrees



Finally, complex geometries can be easily visualised from different perspectives. This is an example using the geometry definition file **phantom** which has been rotated to **105,0,0**

7g. Inspect the input file **disc.in**, to see what is to be calculated. Details on the structure of this file, on the meaning of different keywords and on the geometry definition can be found in Chapter 7 of the PENELOPE manual (**penelope-2014-NEA.pdf**) or in the heading comments of the **penmain.f** source file.

7h. Execute **penmain.exe**, using **disc.in** as input file, *i.e.*, type

```
> penmain < disc.in
```

7i. Inspect the file **material.dat** (check of the materials read), **penmain.dat** (transcript of data in the input files) and **penmain-res.dat** (output file).

7j. There are other output files for plotting. To display the contents of each of these files on the screen, copy the gnuplot scripts ***.gnu** from directory **.\penelope\mains\penmain\gscripts** into the working directory **.\examples\1-disc** and type

```
> wgnuplot file-name.gnu
```

7k. Run the other six exercises in subdirectory **.\work\penmain\examples** (**2-plane**, **3-detector**, **4-x-ray-tube**, **5-accelerator**, **6-polarisation**, and 7-**aba**) by following similar steps.

STRUCTURE OF A USER'S MAIN PROGRAM

To take full advantage of PENELOPE, the user should write a steering main program adapted to the peculiarities of the considered problem. The following Fortran listing illustrates the structure of a main program for simulation with quadric geometries (similar to the example program **penmain.f**). The comment lines beginning with 'cu' indicate parts of the program that are specific to each experiment and/or have to be coded by the user. These include the definition of the source characteristics (*i.e.*, the specification of the initial states of primary particles) and the scoring of relevant quantities and distributions.

```
C...+....1....+....2....+....3....+....4....+....5....+....6....+....7..
C     PROGRAM MAIN
C
      USE PENELOPE_mod  ! I/O of PENELOPE
C     Material data (MAXMAT=10), simulation parameters:
C     ABS(3,MAXMAT), C1(MAXMAT),C2(MAXMAT),WCC(MAXMAT),WCR(MAXMAT)
C     Energy loss due to soft collisions: E0STEP,SSOFT,DESOFT
      USE TRACK_mod  ! Particle track variables:
C     E,X,Y,Z,U,V,W,WGHT,KPAR,IBODY,MAT,ILB(5),PAGE
C     SP1,SP2,SP3,IPOL  ! Photon polarisation (Stokes parameters)
      USE PENGEOM_mod  ! I/O of the geometry routines
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER*4 (I-N)
      CHARACTER PMFILE(MAXMAT)*20  ! Material data files
C  ************  Geometry parameters.
      DIMENSION PARINP(20),DSMAX(5000)
      COMMON/RSEED/ISEED1,ISEED2 ! Random-number generator seeds
C
cu << Define counter arrays and initialise them to zero
cu    Set NTOT (total number of showers to be simulated) >>
C  ************  Initialisation of PENELOPE.
cu << Set the values of the simulation parameters in module PENELOPE_mod
cu    and the seeds of the random-number generator in common RSEED >>
cu << Define EPMAX (largest energy in the simulation) and NMATR (number
cu    of materials in the geometrical structure) >>
      PMFILE(1)='material_1.mat' ! Material data files (input)
      PMFILE(2)='material_2.mat' ! etc.
      INFO=3 ! Print detailed information on the transport models
      OPEN (UNIT=16) ! Output file
      CALL PEINIT(EPMAX,NMATR,16,INFO,PMFILE) ! Initialises PENELOPE
      CLOSE(UNIT=16)
C  ************  Geometry definition.
      NPINP=0 ! All geometry parameters are defined from the input file
      OPEN(17,FILE='my-geometry.geo')  ! Geometry definition file
      OPEN(18,FILE=’geometry.rep’) ! Geometry report
      CALL GEOMIN(PARINP,NPINP,NMATG,NBOD,17,18) ! Initialises PENGEOM
      CLOSE(UNIT=17)
      CLOSE(UNIT=18)
      IF(NMATG.GT.NMATR) STOP ! The geometry contains too many materials
cu << Define DSMAX(IBODY) for all bodies >>
C  ************  Simulation.
cu << Initialise global counters >>
      N=0
 10   N=N+1 ! New shower

C  ++++++++++++++++++++++++  Generate a new shower.
cu << Set the initial state variables of the primary particle, possibly
cu    by random sampling from the source distribution. Define _ALL_ the
cu    variables in TRACK_mod >>
C  ****  Check if the trajectory intersects the material system.
      CALL LOCATE ! Determines the body where the particle moves
      IF(MAT.EQ.0) THEN ! The particle is outside all material bodies
        CALL STEP(1.0D30,DSEF,NCROSS) ! Move the particle ahead
        IF(MAT.EQ.0) THEN ! The particle does not enter the system
          GOTO 40 ! Exit
        ENDIF
      ENDIF
      CALL CLEANS ! Cleans the secondary stack
C  ------------------------  Simulation of a new track.
 20   CALL START ! Starts simulation in current medium
 30   CALL JUMP(DSMAX(IBODY),DS) ! Determines segment length
      CALL STEP(DS,DSEF,NCROSS) ! Moves particle to end of step or boundary
cu << Score relevant quantities >>
      IF(NCROSS.GT.0) THEN
        IF(KPAR.NE.2) E=E0STEP-SSOFT*DSEF ! Soft energy loss correction
        IF(MAT.EQ.0) THEN ! The particle is outside the enclosure
          GOTO 40 ! Exit.
        ENDIF
        IF(E.LT.EABS(KPAR,MAT)) GOTO 40 ! The particle is absorbed
        GOTO 20
      ENDIF
      CALL KNOCK(DE,ICOL) ! Simulates the interaction event
cu << Score relevant quantities >>
      IF(E.LT.EABS(KPAR,MAT)) GOTO 40 ! The particle is absorbed
      GOTO 30
C  ------------------------  The simulation of the track ends here.
 40   CONTINUE
cu << Score relevant quantities >>
C  ****  Any secondary left?
      CALL SECPAR(LEFT)
      IF(LEFT.GT.0) THEN
cu << The secondary particle extracts energy from the site; modify
cu    deposited energy counters accordingly >>
        GOTO 20
      ENDIF
C  ++++++++++++++++++++++++  The simulation of the shower ends here.
      IF(N.LT.NTOT) GOTO 10
cu << Calculate final averages and write results in output files >>
      END
C...+....1....+....2....+....3....+....4....+....5....+....6....+....7..
```