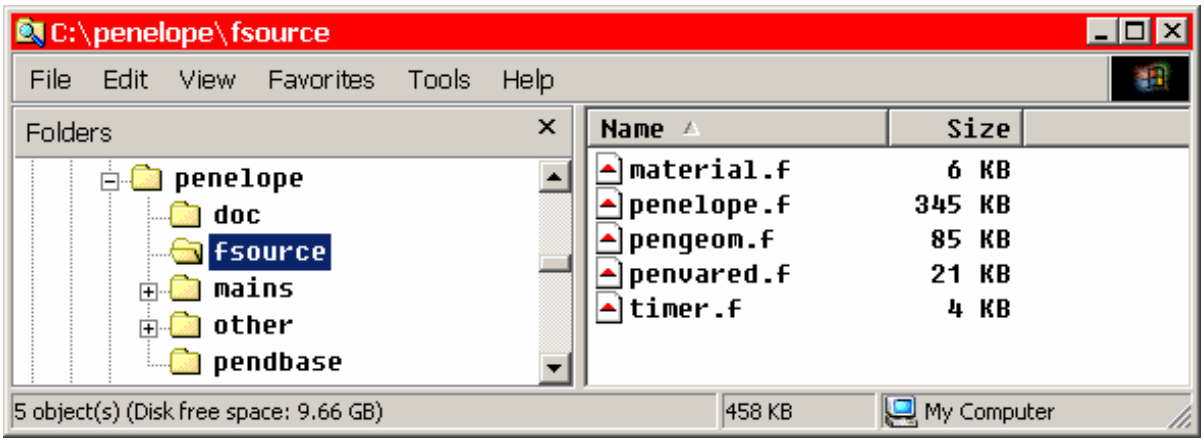
Tutorial for PENELOPE (version 2005)

The distribution package looks like (in Windows)



The subdirectories shown contain:

\doc Documentation of the code system. It contains the following files

tutorial.pdf … This file.

manual.pdf, **manual.txt** … This document provides a brief description of the PENELOPE system and its use. ***It is the document to have at hand in the initial stages of the use of PENELOPE.***

penelope_2003_NEA.pdf … Official release by the OECD Nuclear Energy Agency Data Bank of the PENELOPE documentation, distributed together with version 2003. This is the reference to be used in any publication. Cite it as:
F. Salvat, J.M. Fernández-Varea and J. Sempau, "PENELOPE - A Code System for Monte Carlo Simulation of Electron and Photon Transport" (OECD Nuclear Energy Agency, Issy-les-Moulineaux, France, 2003).

\fsource FORTRAN 77 source files of the PENELOPE code system:

penelope.f … Transport/physics simulation routines.

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 (to be used together with **material.exe** and **shower.exe**).

\mains This directory contains three examples of main programs (**penslab**, **pencyl** and **penmain**) together with examples of input, material and geometry data files. GNUPLOT scripts for visualization of simulation results from the example main programs are also included.

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

COMPILER AND PLOTTER

- To generate the executable binary files of your simulation programs you need a FORTRAN 77 compiler. If you do not have one installed on your computer, you may use the G77 compiler from the Free Software Foundation. The compact G77 for Win32 (Windows 9x/NT/2000/XP) package can be downloaded from the site "http://www.geocities.com/Athens/Olympus/5564".
- To plot the results of the example main programs you will need a plotting program. We shall use 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. 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 clicking on its icon.

EXERCISES

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

1. Run material

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

```
> g77 -O -Wall material.f -o material.exe
```

The command switch "**`-O`**" optimizes 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 modules used by a main program have been declared through **`include`** statements inside the main program (see, for example, the source file **penmain.f**) and do not have to be listed in the compilation command.

1b. Move **material.exe** to the directory **.\penelope\pendbase**. Execute **material** to create data for "aluminium", for "sodium iodide", for "aluminium oxide", for "glass, plate" and for "polystyrene" (nos. 13, 253, 106, 171 and 227 in the **pdcompos.p05** file). Call the output files **al.mat**, **nai.mat**, **al2o3.mat**, **glass.mat** and **polyst.mat** respectively.

2. Run tables

2a. Create a working directory **.\work\tables** and copy the files **penelope.f** (directory **.\penelope \fsource**), **tables.f** and **tables.gnu** (directory **.\penelope\other\tables)** into this directory. Compile and link **tables.f** and **penelope.f**,

```
> g77 -O tables.f -o tables.exe
```

2b. Copy the material data files generated in step 1b to directory **.\work\tables** and execute the code **tables** for these materials.

2c. 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).

2d. The program **tables** also generates tables of energy-dependent quantities in separate ASCII files (with the extension "**.tab**"). To visualize these functions with GNUPLOT, type

```
> wgnuplot tables.gnu
```

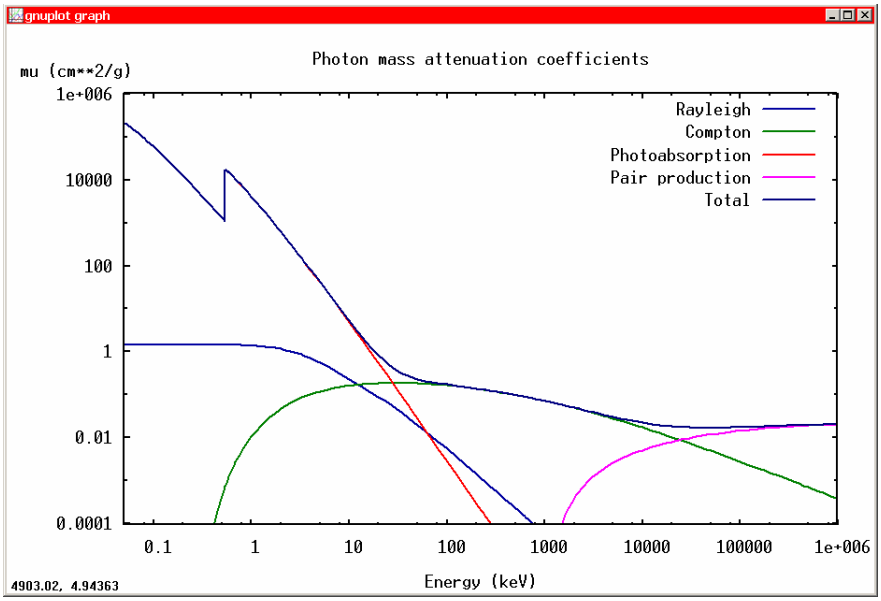
... and follow the instructions.

---

[1] The examples given in this tutorial are for the **g77** compiler. If instead you use, for example, Compaq Visual Fortran 6.5, the equivalent instruction to the compiler is:

```
> DF material.f
```

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

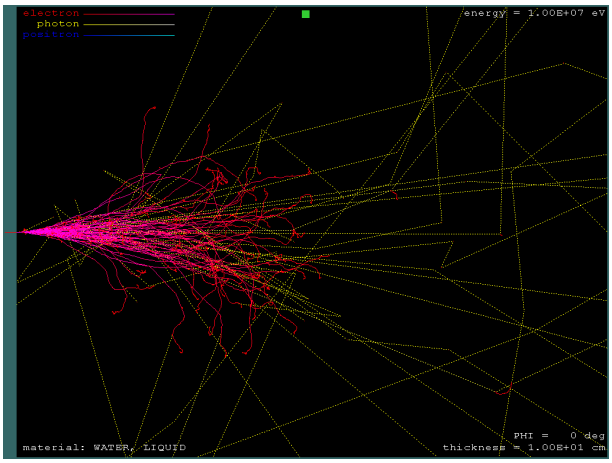


3. Run shower

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

3a. Copy **shower.exe** from **.\penelope\other\shower** to the directory **.\penelope\pendbase**.

3b. Change directory to **.\penelope\pendbase**. Execute the program from the command window by typing **`shower`** or by clicking on the **shower** icon.



The example shown below 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 minimizes the amount of information that has to be typed.

4. Run penslab

4a. Create a working directory **.\work\penslab**, and copy the contents of directory **.\penelope\mains\penslab** to this subdirectory, together with **penelope.f**, **penvared.f** and **timer.f**.

4b. Compile and link the code **penslab.f** with **penelope.f**, **penvared.f** and **timer.f** (see **NOTE** in 1a),

```
> g77 -O penslab.f -o penslab.exe
```

4c. Verify that the file **al.mat** is equivalent to the one that you have created in step 1b.

4d. Inspect **penslab.in**, to see what is to be calculated (an electron beam impinging on a slab). Details on the different keywords and formats can be found in the heading comments of the source file **penslab.f**.

4e. Execute **penslab**, using **penslab.in** as input file, i.e.

```
> penslab < penslab.in
```

(the "**<**" redirects the standard input unit to the file **penslab.in**).

4f. Inspect the file **material.dat** (check of the materials read) and **penslab.da**t (output file).

4g. There are 15 other output files with simulated distributions. To plot these distributions on the screen, type

```
> wgnuplot penslab.gnu
```

…and follow the instructions.

4h. Run **penslab** with the input file **penslab2**. This example illustrates the definition of a source with a continuous energy spectrum.

4i. To visualize the energy spectrum of the source and the actual initial energy distribution of the simulated particles, type

```
> wgnuplot psource.gnu
```

5. Run pencyl

5a. Create a working directory **.\work\pencyl**, and copy the contents of **.\penelope\mains\pencyl** to this subdirectory, together with the files **penelope.f**, **penvared.f** and **timer.f**.

5b. Compile and link the code **pencyl.f** with **penelope.f**, **penvared.f** and **timer.f** (see NOTE in 1a),

```
> g77 -O pencyl.f -o pencyl.exe
```

5c. Catenate files **nai.mat**, **al2o3.mat**, **al.mat, glass.mat** and **polyst.mat** in **.\penelope\pendbase** into a single file **welld.mat**

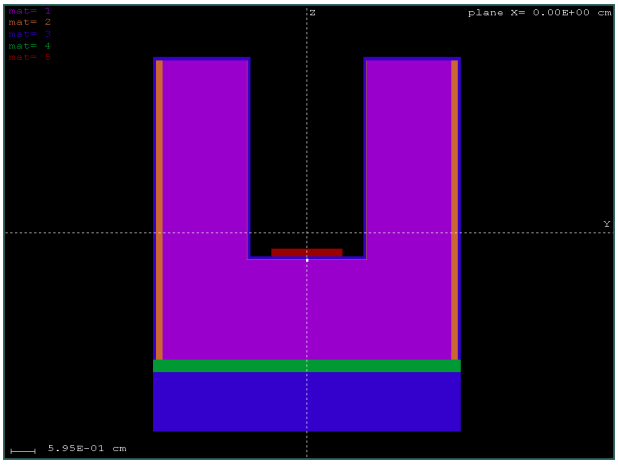
```
> copy nai.mat+al2o3.mat+al.mat+glass.mat+polyst.mat welld.mat
```

and verify that it is identical to the file of the same name in the working directory.

5d. Inspect the input file **welld.in**, to see what is to be calculated (!).

Details on the different keywords and the geometry definition can be found in the PENELOPE report (**penelope_2003_NEA.pdf**) and in the heading comments of the **pencyl.f** source file.

5e. Run the geometry viewer **gviewc**, which displays a 2D view across the cylindrical geometry defined in the input file.



The example shown here corresponds to:

welld.in (pathname of the input file that contains the geometry definition)

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

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

5f. Execute **pencyl**, using **welld.in** as input file, i.e. type

```
> pencyl < welld.in
```

5g. Inspect the file **material.dat** (check of the materials read) and **pencyl.dat** (output file).

5h. There are 18 other output files for plotting. To display them on the screen, type

```
> wgnuplot pencyl.gnu
```

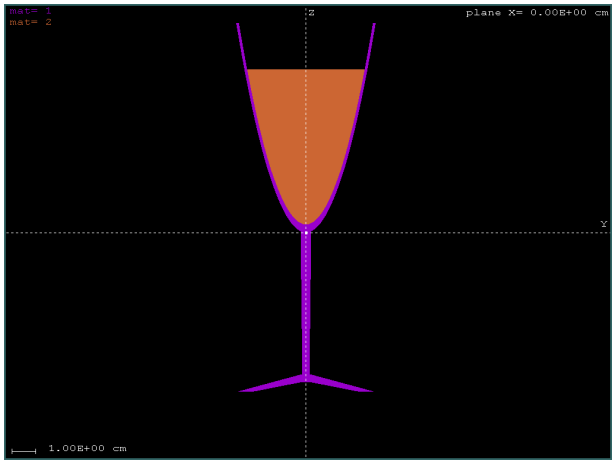
6. Run the quadric geometry viewers

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

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

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 5 of the PENELOPE report (**penelope_2003_NEA.pdf**).

6b. The code **gview2d** displays a 2D view across a geometry defined by quadric surfaces. Operation instructions can be 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)

`0,0,0` (coordinates of the screen centre)

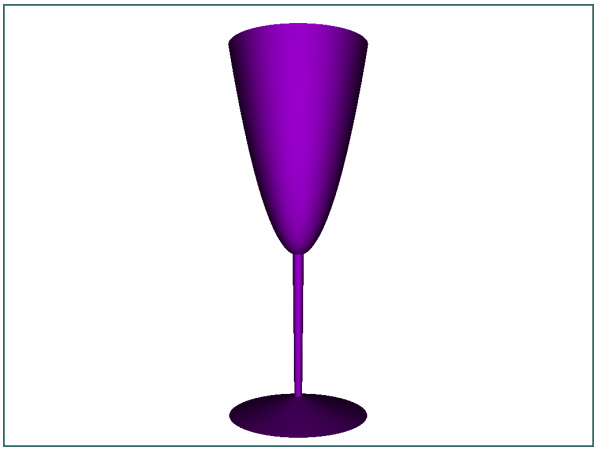
display mode: **`1`**

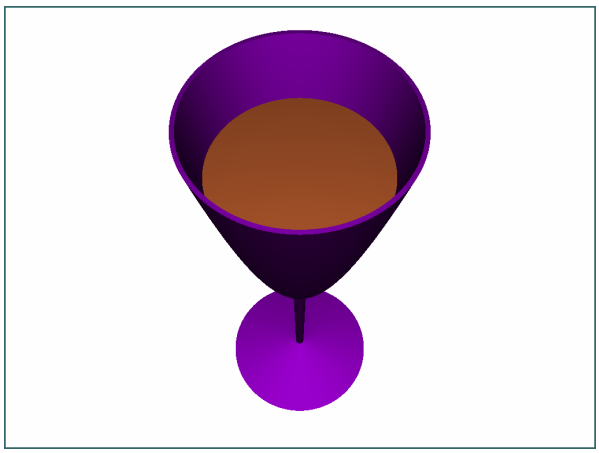
NOTE: gview2d has two display modes. With mode 1, the program shows the materials present in the geometrical structure. In mode 2, the code displays bodies and body numbers, which are needed for scoring purposes.

6c. **gview3d** 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 pictures shown below have been generated with the highest resolution (**`1`**, slow).

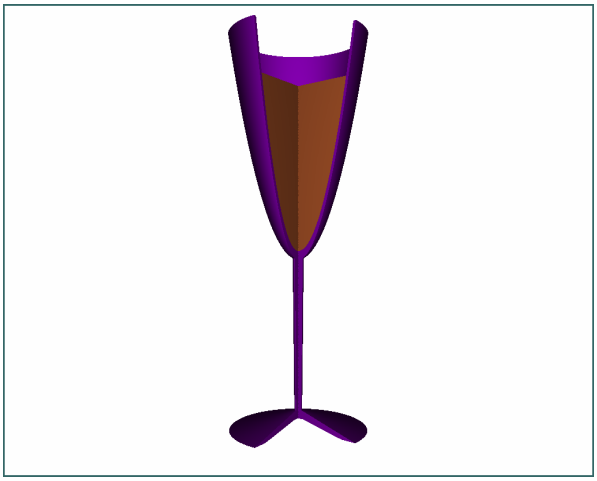
The example shown below is the same geometry used for **gview2d**, i.e., a glass of champagne:



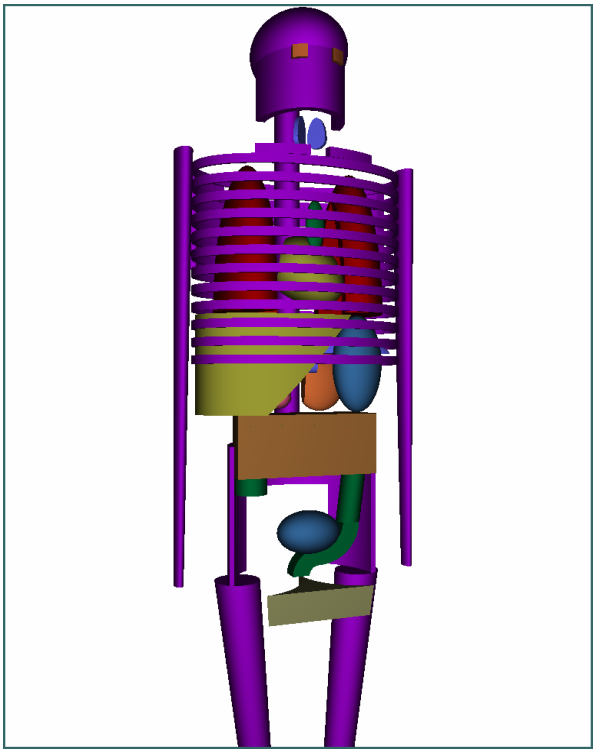


`glass` (path+name of geometry definition file)
`10` (distance from the window to the object)
`no` (for excluding a sector; see below)

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



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



Finally, the complex geometries possible in PENELOPE can be easily visualized from different perspectives. This is an example using the geometry definition file **male** which has been rotated to `(110,0,0)`.

7. Run penmain

7a. Create a new working directory named **.\work\penmain**, and copy there the contents (files and directories) of directory **.\penelope\mains\penmain**, together with **penelope.f**, **pengeom.f**, **penvared.f** and **timer.f**.

7b. Compile and link the file **penmain.f** with **penelope.f**, **pengeom.f** , **penvared.f** and **timer.f** (see the **NOTE** in 1a),

```
> g77 -O penmain.f -o penmain.exe
```

7c. Copy the file **penmain.exe** and the GNUPLOT scripts ***.gnu** in directory **.\work\penmain** to the directory **.\work\penmain\examples\accelerator**, which will be the working directory for what follows. This example simulates a simple electron accelerator head and calculates the dose distribution in a water phantom.

7d. Inspect the file **accel1.in**, to see what is to be calculated (!).
Details on the different keywords, and input formats can be found in the short manual of PENELOPE, **manual.txt**, and as heading comments in the source file **penmain.f**.

7e. Inspect the geometry definition file **accel.geo**, and visualize it using the viewers **gview2d** and **gview3d**.

7f. Inspect the file **accel.mat**, which contains the concatenated material-data files.

7g. Execute **penmain**, using **accel1.in** as input file, i.e.

```
> penmain < accel1.in
```

7h. Inspect the file **pm_material.dat** (check of the materials read) and **penmain.dat** (output file).

7i. The code **penmain** creates multiple output files with simulated distributions, which can be visualized by using the GNUPLOT scripts contained in the working directory. To display the distributions created by default, type

```
> wgnuplot penmain.gnu
```

Try to run the other GNUPLOT scripts to visualize distributions specific to this particular case.

7j. This first run generates the phase-space file at the exit of the accelerator head. We can now obtain the dose distribution in the water phantom by running **penmain** again with the input file **accel2.in**.

7k. Inspect the file **accel2.in**, to see what is to be calculated (!).

7l. Execute **penmain**, using **accel2.in** as input file, i.e.

```
> penmain < accel2.in
```

7m. Run the GNUPLOT scripts **pm_dose_map.gnu** and **pm_dose_surf.gnu** to display the simulated dose distribution.

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 77 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 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
C     PROGRAM MAIN
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER*4 (I-N)
C  ************  Main-PENELOPE commons.
      COMMON/TRACK/E,X,Y,Z,U,V,W,WGHT,KPAR,IBODY,MAT,ILB(5)
      PARAMETER (MAXMAT=10)
      COMMON/CSIMPA/EABS(3,MAXMAT),C1(MAXMAT),C2(MAXMAT),WCC(MAXMAT),
     1  WCR(MAXMAT)
      COMMON/RSEED/ISEED1,ISEED2
C  ****  Geometry.
      DIMENSION PARINP(20),DSMAX(125)

cu << Define counter arrays and initialize them to zero
cu    Set NTOT (total number of showers to be simulated) >>

C  ************  Initialization of PENELOPE.
cu << Set the values of the parameters in the common blocks CSIMPA
cu    (simulation parameters) and RSEED (seeds of the random number
cu    generator) >>
cu << Define EPMAX (largest energy in the simulation) and NMAT (number
cu    of materials in the geometrical structure) >>
      OPEN(15,FILE='my_materials.mat') !Materials data file (input)
         INFO=4 !Print detailed information on the transport models
         CALL PEINIT(EPMAX,NMAT,15,6,INFO) !Initializes PENELOPE
      CLOSE(UNIT=15)

C  ************  Geometry definition.
      NPINP=0 !All geometry parameters are defined from the input file
      OPEN(15,FILE='my_geometry.geo')
```

```
         CALL GEOMIN(PARINP,NPINP,NMATG,NBOD,15,6) !Initializes PENGEOM
      CLOSE(UNIT=15)
      IF(NMATG.GT.NMAT) STOP !The geometry contains too many materials
cu << Define DSMAX(IBODY) for all bodies >>

C  ************  Simulation.
cu << Initialize global counters >>
      N=0
 10   N=N+1

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    parameters in COMMON/TRACK/ >>
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 10 !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
      IF(MAT.EQ.0) THEN !The particle left the material system
         GOTO 40 !Exit
      ENDIF
      IF(NCROSS.GT.0) GO TO 20 !The particle crossed an interface
      CALL KNOCK(DE,ICOL) !Simulates the interaction event
cu << Score relevant quantities >>
      IF(E.LT.EABS(KPAR,MAT)) THEN !The particle has been absorbed
         GOTO 40 !Exit
      ENDIF
      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 on output files >>
      END
C...+....1....+....2....+....3....+....4....+....5....+....6....+....7..
```