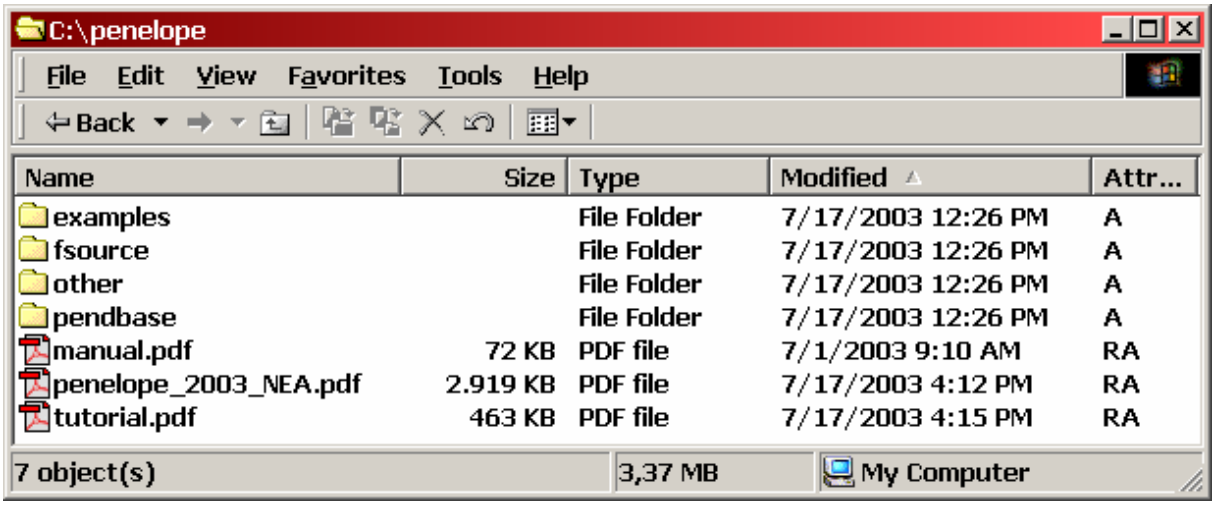
Tutorial for PENELOPE (version 2003)

The distribution package looks like (in Windows)



where:

tutorial.pdf This file.

manual.pdf 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).

The subdirectories shown contain:

\fsource FORTRAN 77 source files of the PENELOPE code system. Includes the transport/physics routines (**penelope.f**), the quadric geometry package (**pengeom.f**), variance-reduction routines (**penvared.f**), the main program to create cross-section data files (**material.f**) and a program to generate tables of material interaction properties (**tables.f**).

\pendbase Files necessary for creating cross-section data (to be used together with **material.exe** and **shower.exe**).

\other Additional software for geometry visualization (**gview**\), display of particle tracks (**shower**\), plotting (scripts for the plotting program GNUPLOT in subdirectory **plotter**\), and a routine package for including static electromagnetic fields (**emfields**\).

\examples Three examples of user codes are included (**penslab**, **pendoses** and **pencyl**) together with two material data files (**al.mat** and **naial.mat**[1]) and a clock subroutine (**timer.f, notimer.f**).

[1] If your computer has Microsoft Access installed, files with the extension "**.mat**" are automatically associated with Microsoft Access. Remove temporarily this association if necessary or change the extension of the material data files.

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 user codes 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[2] the code **material.f** and **penelope.f**,

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

The command switch **`-O`** optimizes code generation; with the option **`-o`** we can specify the name of the produced executable file (the extension "**`.exe`**" is automatically appended).

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

2. Run penslab

2a. Create a working directory **samples****penslab**\, and copy the files **penslab.f**, **penslab.in** and **timer.f** from **example**s\ to this subdirectory, together with **penelope.f** .

2b. Compile and link the code **penslab.f** with **penelope.f**, and **timer.f** ,

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

2c. Copy file **al.mat** from **pendbase**\ to this working directory.

2d. Inspect **penslab.in**, to see what is to be calculated (!).

2e. Execute **penslab**, using **penslab.in** as input file, i.e.

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

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

2f. Inspect the file **material.dat** (check of the materials read) and **penslab.da**t (output file).

2g. There are 15 other output files for plotting. After copying the file **penslab.gnu** from directory **penelope****other****plotter**\ to the working directory **samples****penslab**\, type

```
> wgnuplot penslab.gnu
```

and follow the instructions.

---

[2] 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 penelope.f
```

3. Run pencyl

3a. Create a working directory **\samples\pencyl**, and copy the files **pencyl.f**, **pencyl.in** and **timer.f** from **\examples** to this subdirectory, together with **penelope.f**, **penvared.f**.

3b. Compile and link the code **pencyl.f** with **penelope.f**, **penvared.f** and **timer.f**

```
> g77 -O pencyl.f penelope.f penvared.f timer.f -o pencyl.exe
```

3c. Concatenate files **nai.mat**, **al2o3.mat** and **al.mat** in **pendbase**\ into a single file **naial.mat**

```
> copy nai.mat+al2o3.mat+al.mat naial.mat
```

and copy it to the working directory.

3d. Inspect **pencyl.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.pdf**) and in the heading comments of the **pencyl.f** source file.
The geometry can be inspected using **gviewc**: copy the file **pencyl.in** to **penelope\other\gview** and run **gviewc**

3e. Execute **pencyl**, using **pencyl.in** as input file, i.e.

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

3f. Inspect the file **material.dat** (check of the materials read) and **pencyl.dat** (output file).

3g. There are 18 other output files for plotting. After copying **pencyl.gnu** to this working directory, type

```
> wgnuplot pencyl.gnu
```

4. Run pendoses

4a. Create a new working directory named **samples\pendoses**\, and copy the files **pendoses.f**, **pendoses.geo**, **pendoses.in** and **timer.f** from **\examples** to this working directory, together with **penelope.f** and **pengeom.f**.

4b. Compile and link the code **pendoses.f** with **penelope.f**, **pengeom.f** and **timer.f**

```
> g77 -O pendoses.f penelope.f pengeom.f timer.f -o pendoses.exe
```

4c. Copy the concatenated file **naial.mat** from **pendbase**\ into this working directory.

4d. Inspect **pendoses.in**, to see what is to be calculated (!).
Details on the different keywords and the geometry can be found in the thick PENELOPE report (**penelope_2003.pdf**) and as heading comments in the source file **pendoses.f**.
The geometry can be inspected using the viewers **gview2d** and **gview3d**: copy the file **pendoses.geo** to **\penelope\other\gview** and run **gview2d** and **gview3d**.

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

```
> pendoses < pendoses.in
```

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

4g. There is one other output file for plotting. After copying **pendoses.gnu** to this working directory, type

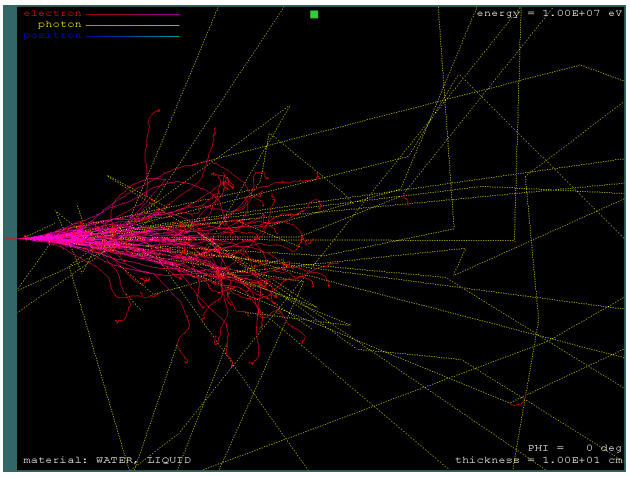
```
> wgnuplot pendoses.gnu
```

5. Run shower

Note that **shower** *runs under Windows (9x/NT/2000/XP) only*

5a. Copy **shower.exe** from **other\shower**\ to the directory **\pendbase**.

5b. Change directory to **\pendbase**. Execute the program from the command window by typing **shower** or by clicking on the **shower** icon. The example shown below corresponds to:



`277` 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

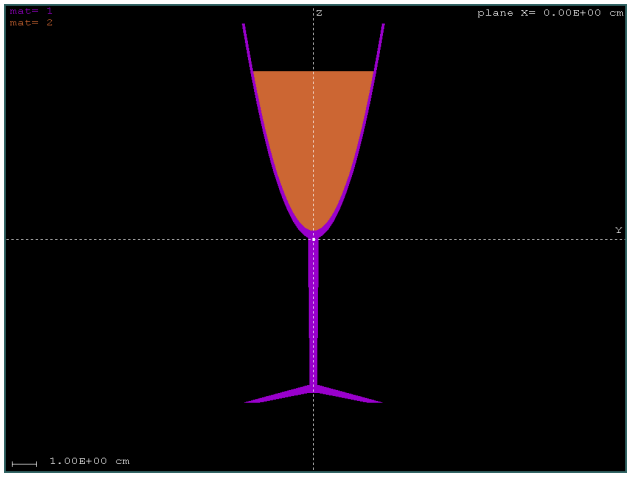
`10` cm for slab thickness

`50` for particles in the bunch

6. Run geometry viewers

Note that the viewers run under Windows (9x/NT/2000/XP) only

6a. **gview2d** displays a 2D view across a geometry defined by quadric surfaces (see chapter 5 of the PENELOPE report).

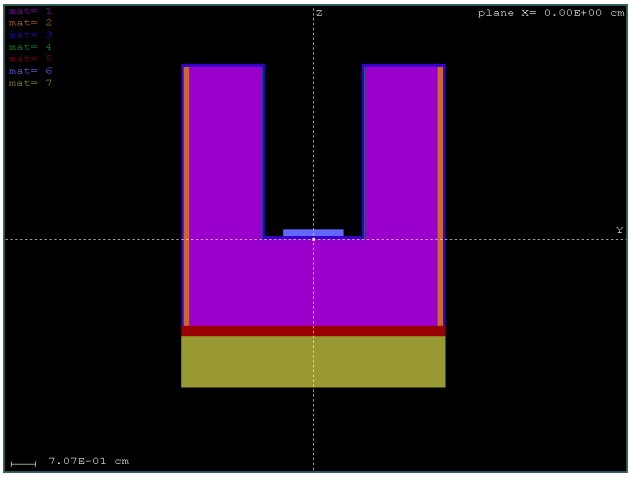


Change directory to **\other\gview**. Execute the program by typing **`gview2d`** or by clicking on the icon.

The example shown here corresponds to:
`glass` (path+name of the geometry definition file)
`0,0,0` (coordinates of the screen centre)

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

6b. **gviewc** displays a 2D view across the cylindrical geometry used in the code **pencyl**.



Change directory to **\other\gview\.** Execute the program by typing **`gviewc`** or by clicking on the icon.

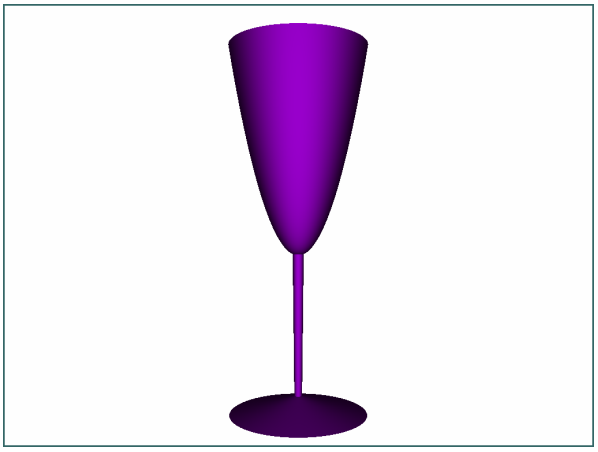
The example shown here corresponds to:
`well_c` (pathname of geometry definition file)
`0,0,0` (coordinates of the screen centre)

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

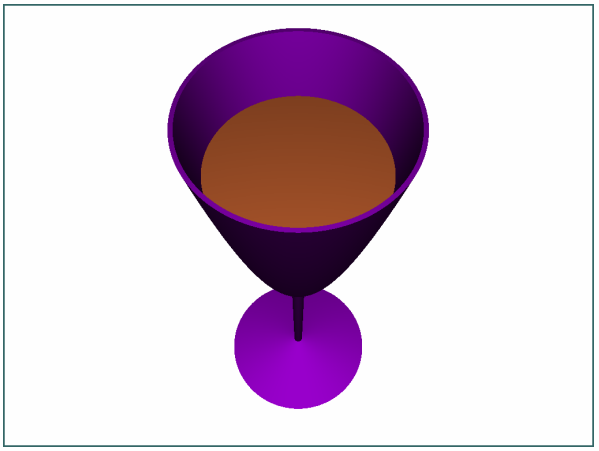
6c. **gview3d** displays a 3D view of the geometry defined by quadric surfaces.

Change directory to **\other\gview\.** Execute the program by typing **`gview3d`** (or click the icon). 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).

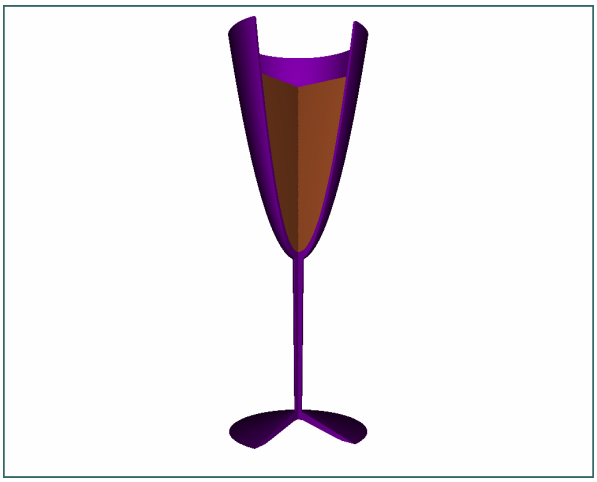
The example shown is the same geometry used for **gview2d**, i.e., a glass of wine:



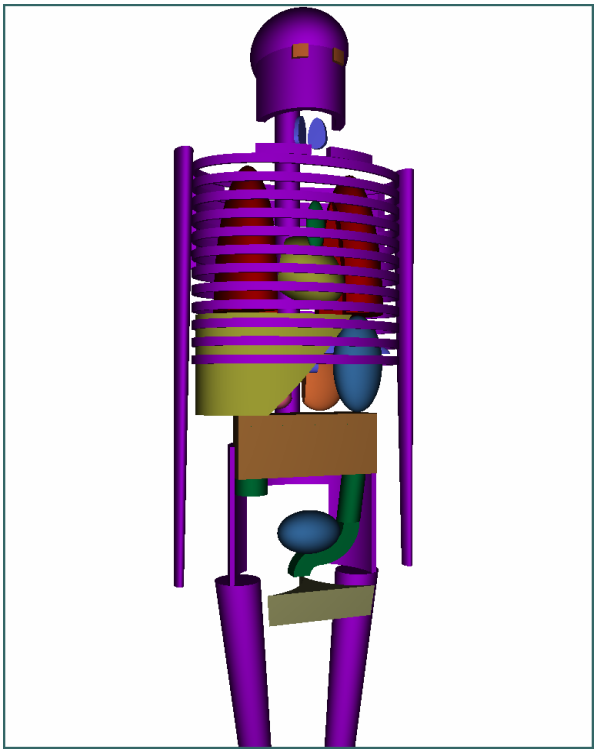
`glass` (path+name of geometry definition file)
`10` (distance from the window to the object)
`no` (for excluding a sector; see below)



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 a sector between **`-35,40`** degrees has been excluded.



Finally, the very complicated 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)`**.

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

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 **pendoses.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..
```