Structure and operation of PENELOPE / penmain

UNIVERSITAT DE BARCELONA

PENELOPE

- **PENELOPE** is an acronym for "**PEN**enetration and **E**nergy **LO**ss of **P**ositrons and **E**lectrons"
- **A general-purpose Monte Carlo simulation code system** with
  - Realistic, well defined interaction models
  - Fast and accurate random sampling algorithms
  - Efficient tools for tracking particles through complex geometries (constructive quadric geometry)
  - Complementary tools: variance reduction, transport in electro-magnetic fields, tabulation of macroscopic interaction parameters, ...
- **Distributed by the OECD/Nuclear Energy Agency Data Bank (Paris) and the RSICC (Oak Ridge).**
  More than 1,500 copies distributed
  List server: **http://www.oecd-nea.org/lists/penelope.html**
- **Main applications:**
  - Radiotherapy and Nuclear Medicine
  - Dosimetry and radiation metrology
  - Electron microscopy (SEM, electron-probe microanalysis)
  - Detector response, x-ray generators, ...

PENELOPE

- **Main reference:** very detailed manual, free hard copies available

F. Salvat
PENELOPE-2014: A Code System for Monte Carlo Simulation of Electron and Photon Transport
OECD NEA Data Bank/NSC DOC(2011)/5
(OECD Nuclear Energy Agency, Issy-les-Moulineaux, 2011)
http://www.oecd-nea.org/lists/penelope.html

- **Other references:**

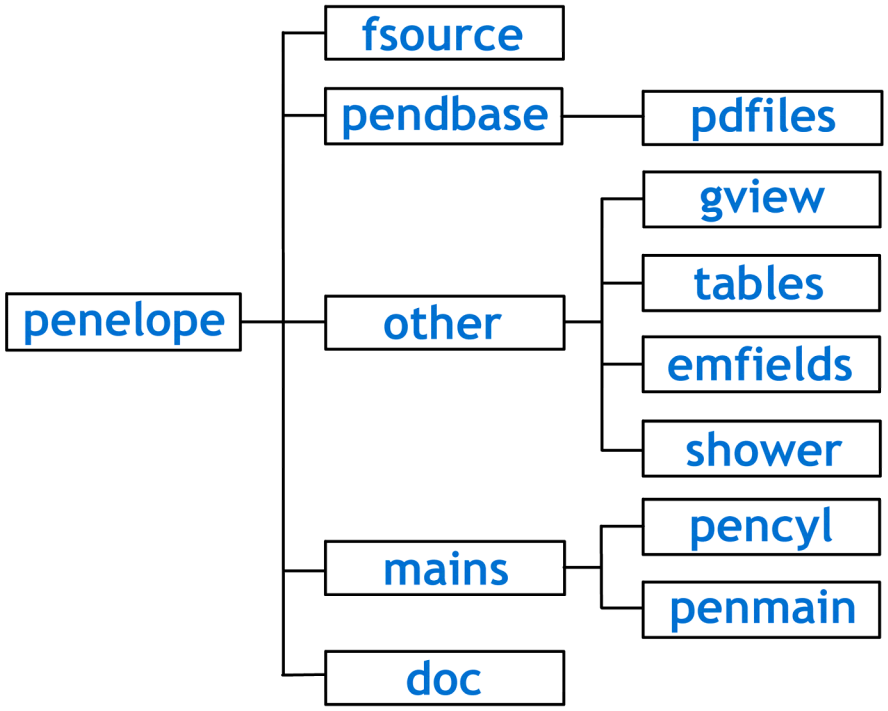
J. Baró, J. Sempau, J.M. Fernández-Varea and F. Salvat
"PENELOPE: an algorithm for Monte Carlo simulation of the penetration and energy loss of electrons and positrons in matter"
Nucl. Instrum. Meth. B **100** (1995) 31–46

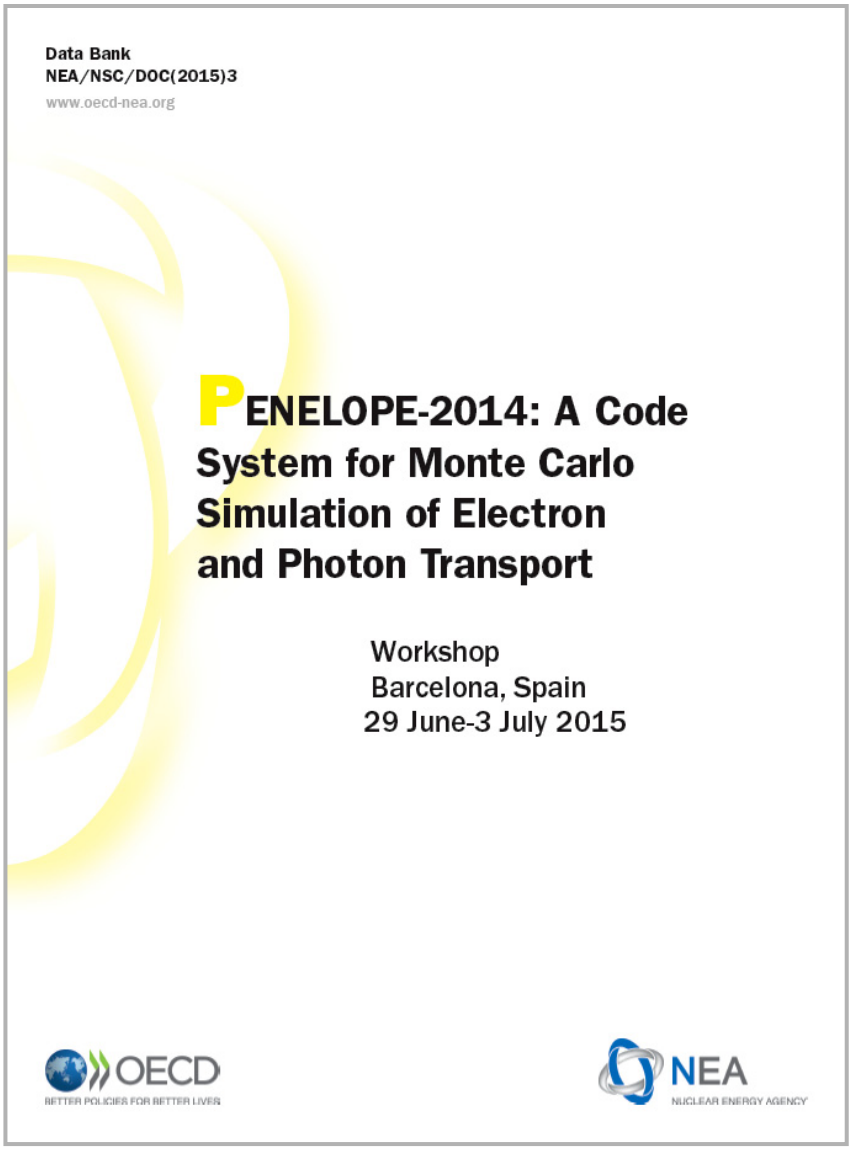
J. Sempau, J.M. Fernández-Varea, E. Acosta and F. Salvat
"Experimental benchmarks of the Monte Carlo code PENELOPE"
Nucl. Instrum. Meth. B **207** (2003) 107-123

F. Salvat and J. M. Fernández-Varea
"Overview of physical interaction models for photon and electron transport used in Monte Carlo codes"
Metrologia **46** (2009) S112–S138

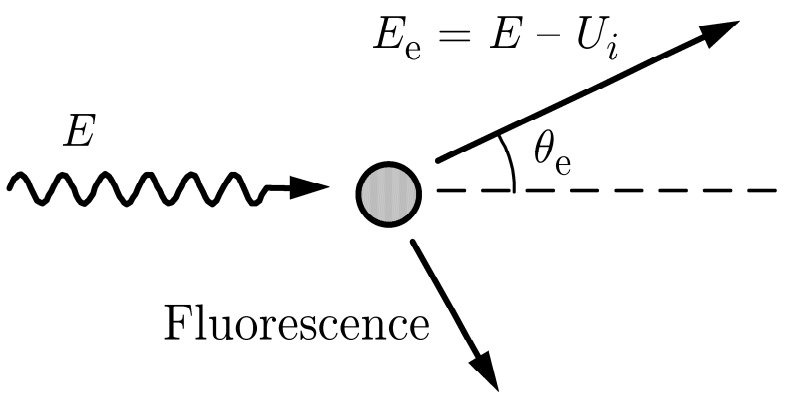
Distribution package

- A single zip compressed file, **penelope.zip** (~79 Mb)

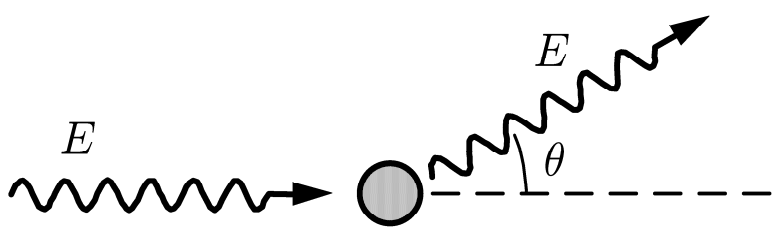




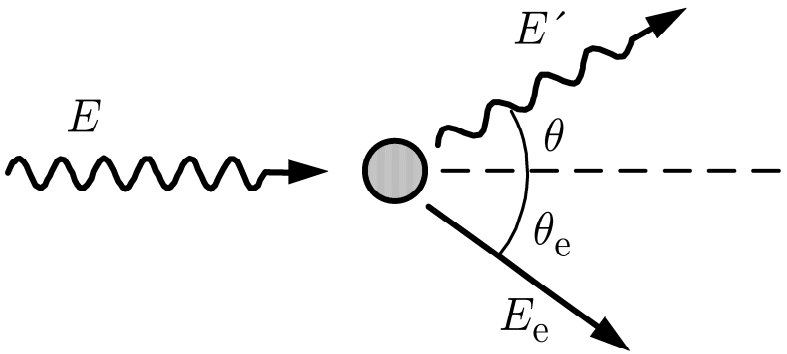
Interactions of photons



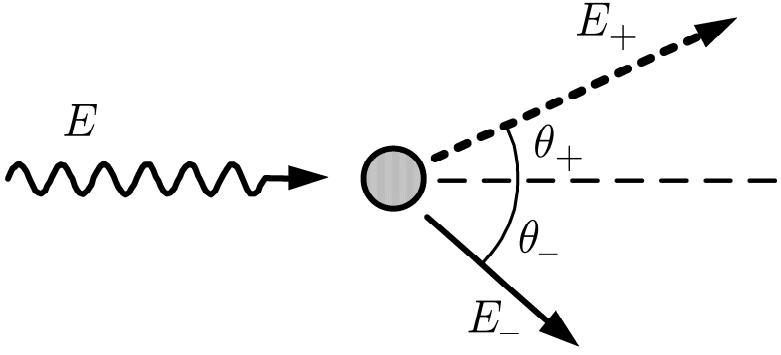
Photoelectric absorption



Rayleigh scattering



Compton scattering

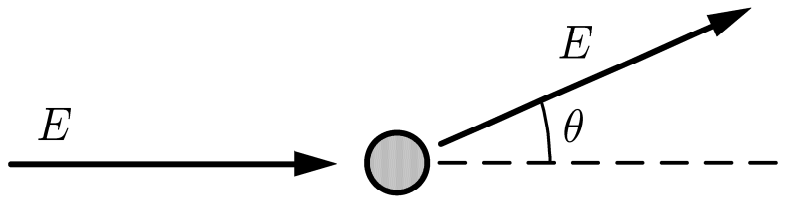


Pair production

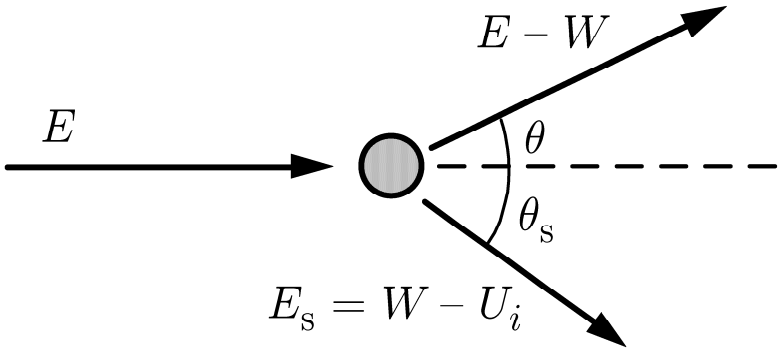
$m_e c^2 \simeq 511$ keV, electron rest energy

- **Photoelectric effect:**
  - Total cross sections calculated from the DHFS atomic potential (equivalent to Scofield's LLNL database; Cullen *et al.*, 1997)
  - Angular distribution of photoelectrons from Sauter's (1931) formula (plane-wave Born approximation for K-shell hydrogenic ions)
  - Atomic relaxation from the EADL (Perkins *et al.*, 1991)
- **Coherent (Rayleigh) scattering:**
  - Total cross sections from the EPDL (Cullen *et al.*, 1997), includes anomalous atomic scattering factors
  - Angular distribution from atomic form factors
- **Incoherent (Compton) scattering:**
  - Double-differential cross sections (DDCS) calculated from the relativistic impulse approximation (Ribberfors, 1983) using analytical Compton profiles (Brusa *et al.*, 1996)
  - Total cross sections obtained as integrals of the DDCS
  - Subsequent atomic relaxation from the EADL (Perkins *etal.*, 1991)
- **Electron-positron pair production:**
  - Total cross sections from the EPDL (Cullen *et al.*, 1997), includes triplet production
  - Energies and directions of the pair particles from the Bethe-Heitler theory

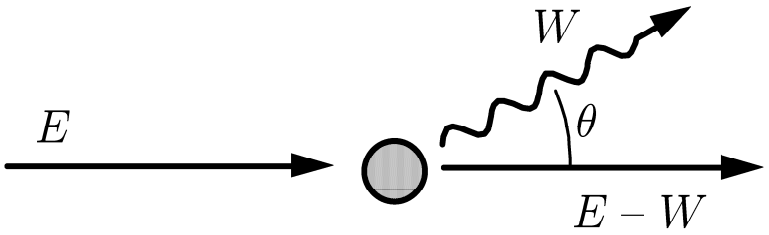
Interactions of electrons and positrons



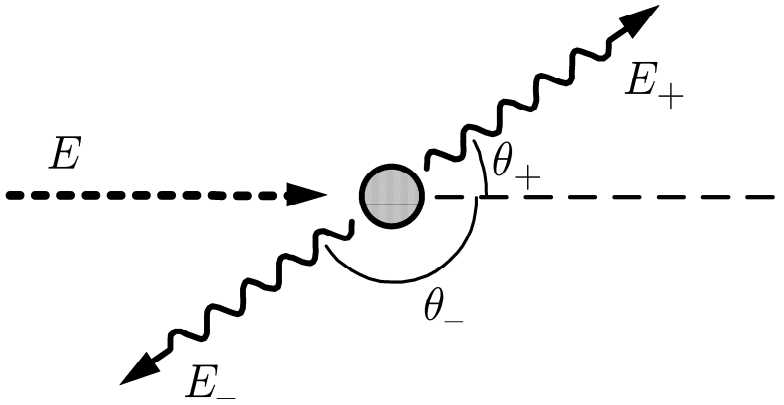
Elastic scattering



Inelastic scattering



Bremsstrahlung emission



Positron annihilation

$m_e c^2 \simeq 511$ keV, electron rest energy

- **Elastic collisions:**
  - Atomic differential cross sections (DCS) calculated using the Dirac partial-wave expansion method (ICRU 77, 2007)
  - High-energy modified Wentzel model with correct first and second moments (ICRU 77, 2007) for energies above 100 MeV

- **Inelastic collisions:**
  - DDCS from the Born approximation, using the Sternheimer-Liljequist GOS model, with resonance energies fitted to reproduce the mean excitation energies from ICRU 37 (1984)
  - Secondary electrons emitted in the direction of momentum transfer

- **Bremsstrahlung emission:**
  - Photon-energy scaled DCSs of Seltzer and Berger (1985, 1986)
  - Photon angular distribution fitted to partial-wave data of Kissel *et al.* (1983)

- **Impact ionization of inner shells**
  - Total cross sections for K, L and M shells calculated from the distorted-wave Born approximation (Bote *et al.*, 2009)
  - Subsequent atomic relaxation from the EADL (Perkins *et al.*, 1991)

Electron/positron transport mechanics

- **Mixed simulation algorithm:**

PENELOPE uses a pure class II (mixed) algorithm for electrons/positrons. Allows verifying the stability under variations of simulation parameters

Hard interactions (with angular deflection larger than a cutoff angle θ_c or energy loss larger than selected cutoffs) **are simulated individually**

- **Hard elastic interactions:**

The cutoff angle is determined by two user parameters, C_1 and C_2 , according to the formula (Eq. 4.85 of the manual)

$$\lambda_{\rm el}^{\rm (h)}(E) = \max\left\{\lambda_{\rm el}(E), \min\left[C_1\lambda_{{\rm el},1}(E), C_2\frac{E}{S(E)}\right]\right\}$$

- **Hard energy loss events:**

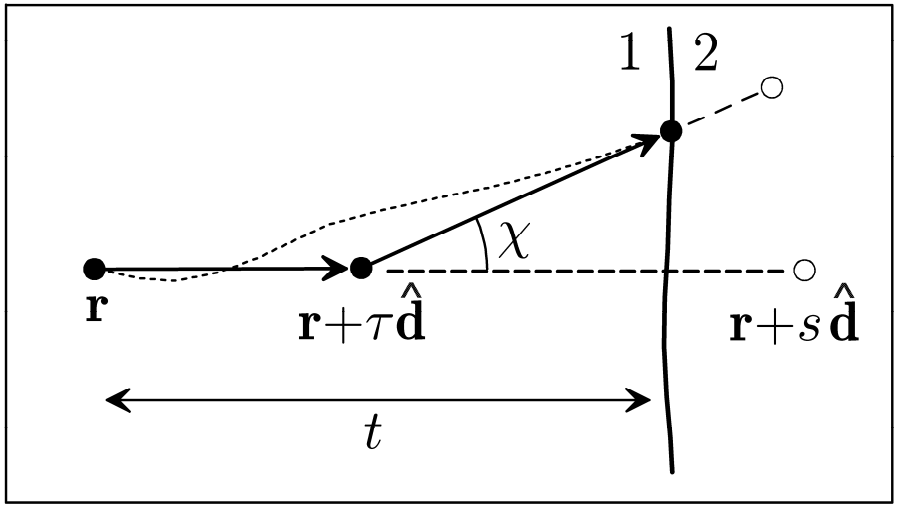
The user defines the cutoff energies $W_{\rm cc}$ (col) and $W_{\rm cr}$ (brems), in accordance with the required energy resolution

- **Maximum allowed step length between hard interactions:**

An additional parameter, $s_{\rm max}$, sets a limit to the step length (needed to account for the variation of energy along the step, and for consistency of the simulation of soft events)

Electron/positron transport mechanics

- **Simulation of soft interactions:** Random hinge method
  - The global effect all the soft interactions in a step s between a pair of hard interactions is simulated as a single event, a **hinge**
  - The angular deflection and the energy loss at the hinge are sampled from approximate distributions having the correct first and second moments
  - The position of the hinge, τ, is sampled uniformly in $(0,s)$
    $\Rightarrow$ simple and accurate scheme for interface crossing



- Includes elaborate corrections to account for the variation of energy along the step

Role/effect of the simulation parameters

- **Step-length control (for each material):**

C_1 limits the average angular deflection per step, $1 - \langle \cos\theta \rangle \lesssim C_1$
Influences the simulation speed only at intermediate energies

C_2 limits the average fractional energy loss per step, $\langle E_0 - E \rangle \lesssim C_2 E_0$
Affects simulation speed only at high energies

- **Energy-straggling control (for each material):**

$W_{\rm cc}$ energy-loss threshold (in eV) for hard inelastic collisions

$W_{\rm cr}$ energy-loss threshold (in eV) for hard bremsstrahlung events

These cutoffs govern energy resolution. Mild effect on speed

- **Geometrical constraints (local):**

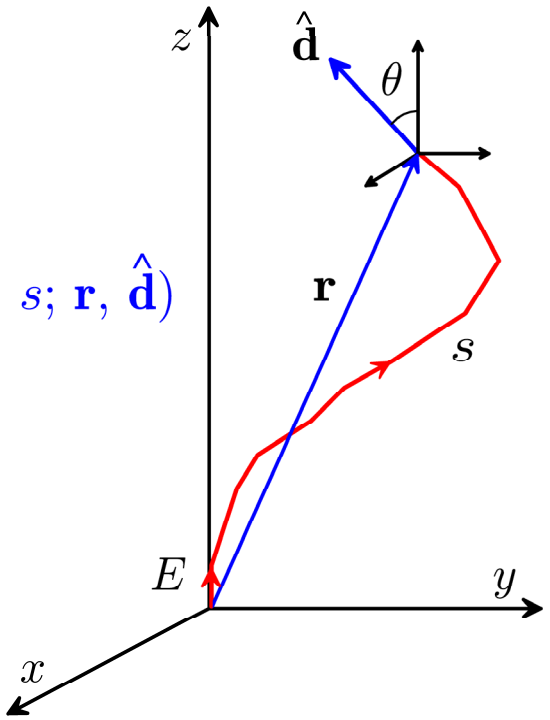
$s_{\rm max}$ maximum step length for "critical" geometries (needed for thin bodies, backscattering, ...)

- **Reasonable "blind" choices:**

C_1 and C_2 : 0.05 to 0.1; $W_{\rm cc}$ and $W_{\rm cr}$: ~ 1,000 eV
$s_{\rm max}$: one tenth of the minimal thickness

Stability under variation of simulation parameters

- **Example:** 500 keV electrons in Al.
  s = 200 μm.



1) Detailed simulation

C_1 = C_2 = 0; W_{cc} = 0 eV

W_{cr} = -100 eV (soft bremss disregarded)

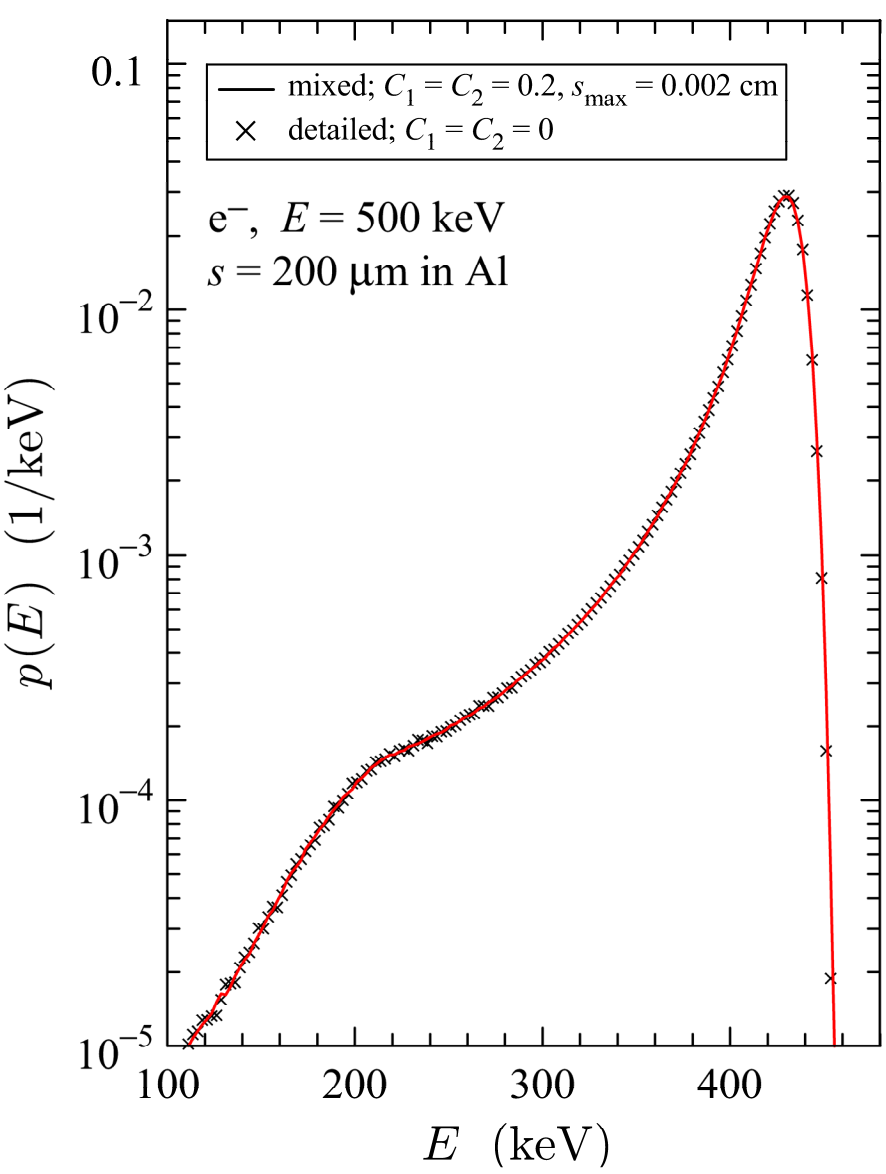
2) Mixed simulation

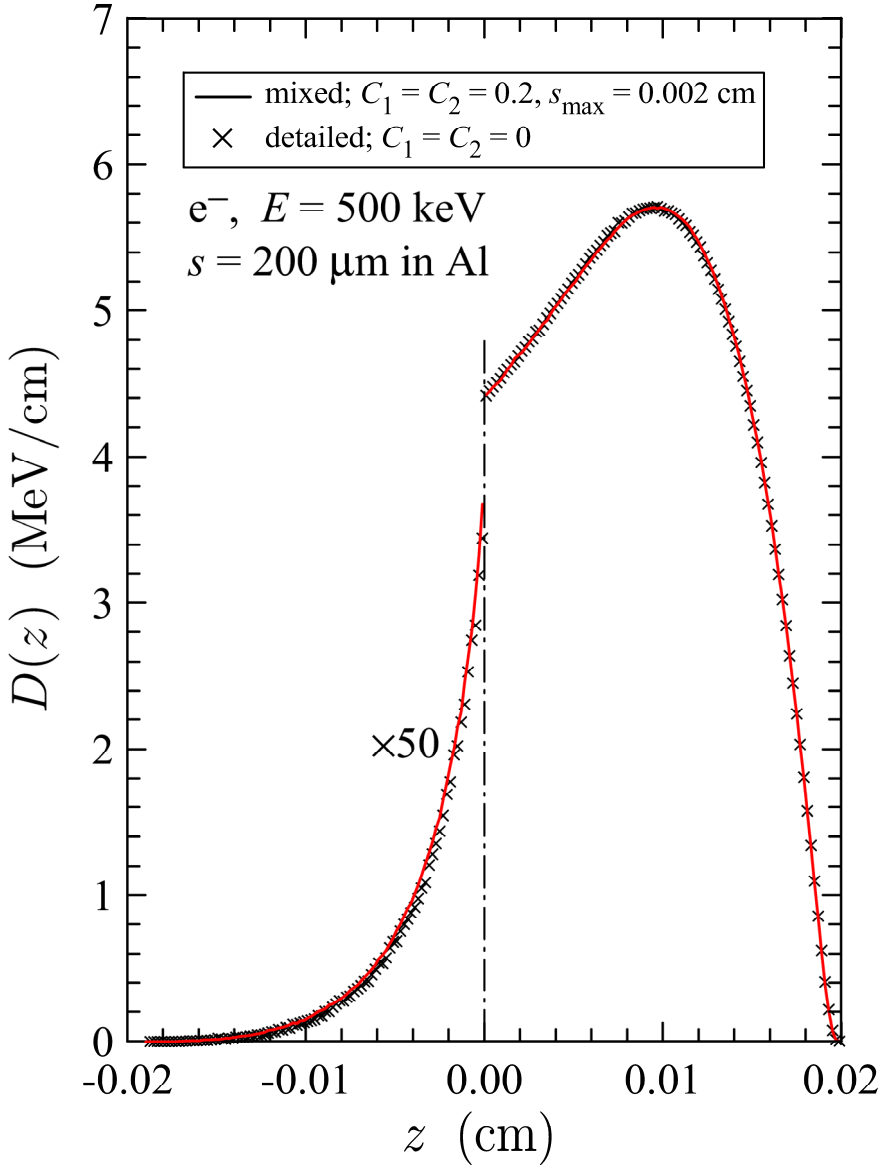
C_1 = C_2 = 0.2 (extreme case!)

W_{cc} = 1 keV; W_{cr} = −100 eV (soft bremss disregarded)

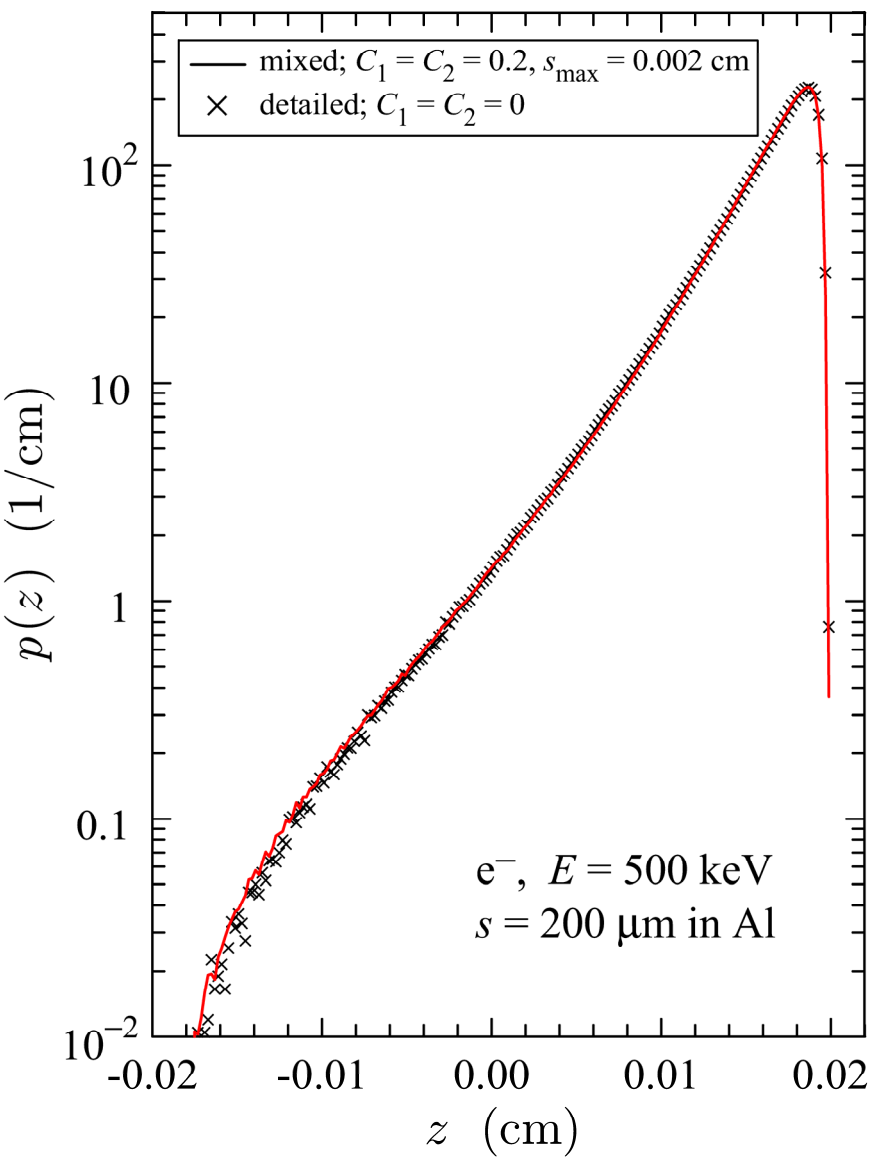
s_{max} = 20 μm

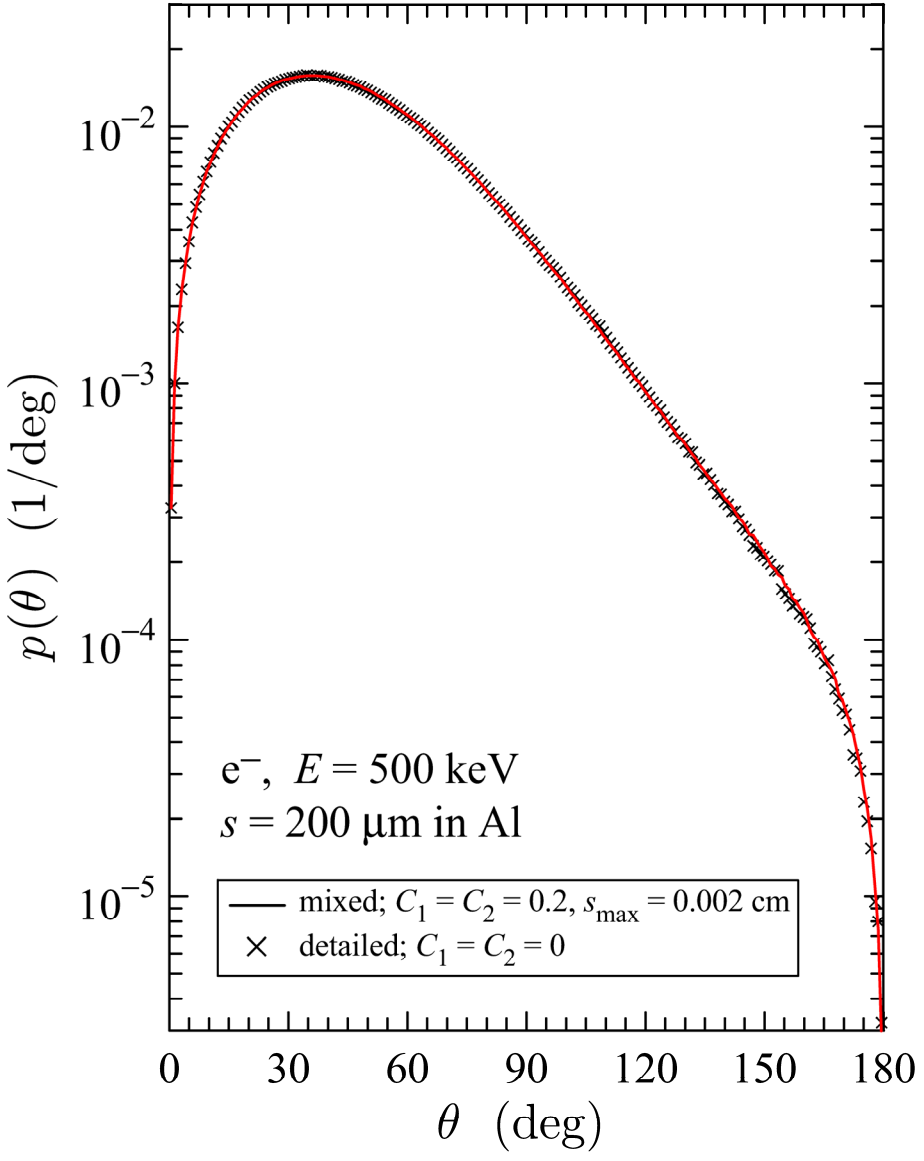
about 75 times faster





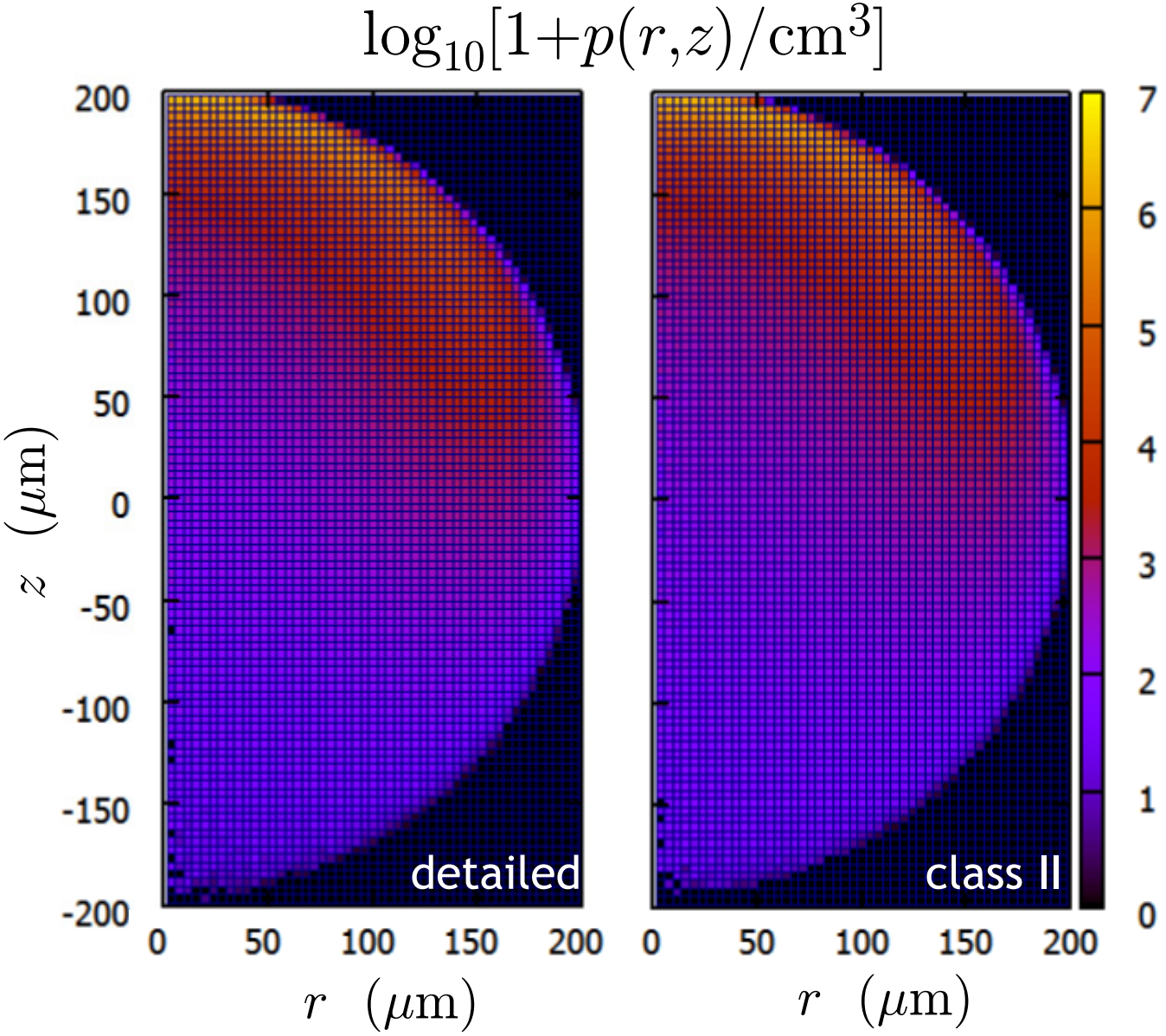
crosses: detailed simulation
solid lines: mixed simulation





crosses: detailed simulation
solid lines: mixed simulation

Stability under variation of the simulation parameters



Structure of the code system

- The **PENELOPE** code system consists of

  - The subroutine package **penelope.f**, which defines the interaction properties of materials and performs the simulation of interactions

  - The geometry package **pengeom.f**, and the 2D and 3D viewers (**gview2d.exe** and **gview3d.exe**)

  - The variance-reduction routines **penvared.f** (include particle splitting, Russian roulette, interaction forcing, bremsstrahlung splitting, x-ray splitting, and delta scattering of photons)

  - The **database**: 995 ascii files with interaction cross sections and other properties of the elements Z=1–99 (hydrogen to einstenium)

  - Steering main programs for cylindrical and quadric geometries, **pencyl.f** and **penmain.f**. They can simulate a variety of radiation sources, allowing scoring of different transport properties

  - Routines **penfield.f** for tracking charged particles in static electromagnetic fields

- ...

  - Program **tables.f** for displaying plots of energy-dependent interaction properties. Macroscopic quantities are made available numerically and graphically

  - A program for displaying electron-photon showers in material slabs, **shower.exe**

  - Documentation: **Manual and tutorial**

- All source programs are written in Fortran, *i.e.*, they can be run on any operating system with a Fortran compiler

- The geometry viewers **gview2d** and **gview3d**, and the program **shower** work only on MS Windows (for the time being)

- The output of the simulation programs and of **tables.f** is formatted for visualization with the free plotting program **gnuplot,** available for Windows and Linux (**http://www.gnuplot.info**)

Complements and software

- Geometry definition files can be edited and debugged by using the Java GUI **PenGeomJar**, which is distributed as a separate package. It contains 2D and 3D viewers which operate in the same way as the MS Windows codes **gview2d** and **gview3d**. Versions are provided for Windows, Linux and generic UNIX platforms

  J. Almansa *et al.*, *Comp. Phys. Commun.* **199** (2016) 102--113

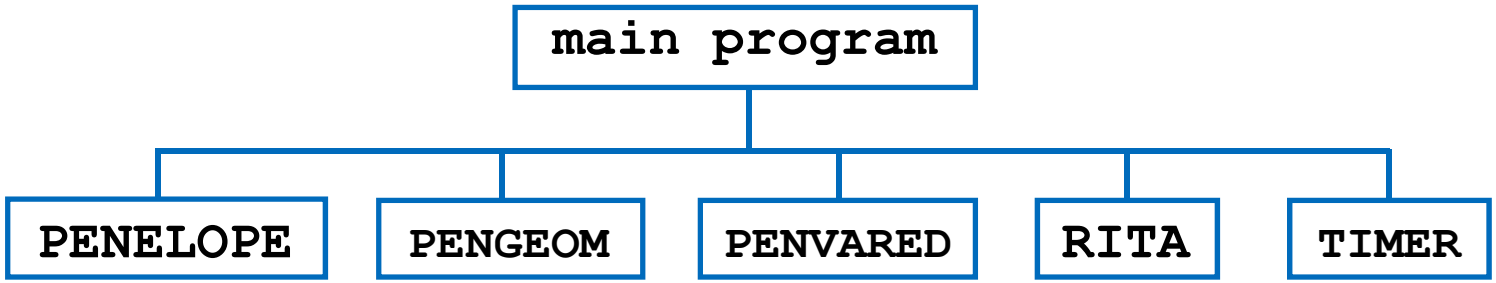
- The software needed to run **PENELOPE** consists of
  - A Fortran compiler (Fortran 90 or later)
  - A scientific plotting program
  - A file manager (**PENELOPE** usually produces multiple output files)

  The package **sofware.zip** contains freeware tools that we are using on MS Windows:
  - The Fortran compiler **gfortran**
  - The plotting program **gnuplot**
  - The file manager **Total_Commander**

  Each of these tools comes with manuals and installation instructions

PENELOPE/PENGEOM

- PENELOPE and PENGEOM are Fortran subroutine packages
- The user must provide a **steering main program** that generates the initial states of primary particles, controls the evolution of the particle histories and keeps score of relevant quantities
- The generic main programs **`penmain`** and **`pencyl`** can handle a wide variety of problems, but yours may require some special treatment **...**



- The communication between the main program and the simulation sub-routines is through a few Fortran modules

In the I/O of the PENELOPE and PENGEOM routines, all energies are in **eV** and all lengths are in **cm**

All real variables are double precision and integers are all integer*4

- To start the simulation of a particle, its initial state variables must be set by the main program. PENELOPE modifies the energy and direction cosines only when the particle undergoes an interaction. The tracking is performed by PENGEOM (updates particle coordinates and controls interface crossings)

- Random sampling from numerical distributions is performed by the subroutine package RITA, which uses the RANECU random number generator implemented in the

```
FUNCTION RAND(DUMMY)
```

The "seeds" **ISEED1** and **ISEED2** must be initialized in the main program and transferred through the named common block

```
COMMON/RSEED/ISEED1,ISEED2
```

This RNG allows repeatability of the simulation. Useful for debugging the the program

Fortran module TRACK_mod

```
      MODULE TRACK_mod
C
C  ****  Particle TRACK variables (to be initialised before calling
C         subroutine START).
      SAVE  ! Saves all items in the module.
C  ----  Energy, position, direction, and weight.
      DOUBLE PRECISION :: E,X,Y,Z,U,V,W,WGHT
C  ----  Particle type, current body, and material.
      INTEGER*4 :: KPAR,IBODY,MAT
C  ----  Particle history flags.
      INTEGER*4, DIMENSION (5) :: ILB
C
C  ****  Photon polarisation
C  ----  Polarised photons if IPOL=1, otherwise unpolarised photons.
      INTEGER*4 :: IPOL=0
C  ----  Stokes parameters.
      DOUBLE PRECISION :: SP1,SP2,SP3
C
C  ****  The particle age (time elapsed since start of shower)
C         is recorded when LAGE=.TRUE.
      LOGICAL :: LAGE =.FALSE.
      DOUBLE PRECISION :: PAGE=0.0D0
C
      END MODULE TRACK_mod
```

Code number (ICOL) for the various interaction events

ICOL	electrons (KPAR=1)	photons (KPAR=2)	positrons (KPAR=3)
1	artificial soft event (random hinge)	coherent (Rayleigh) scattering	artificial soft event (random hinge)
2	hard elastic collision	incoherent (Compton) scattering	hard elastic collision
3	hard inelastic collision	photoelectric absorption	hard inelastic collision
4	hard bremsstrahlung emission	electron-positron pair production	hard bremsstrahlung emission
5	inner-shell impact ionisation		inner-shell impact ionisation
6			annihilation
7	delta interaction	delta interaction	delta interaction
8	auxiliary interaction	auxiliary interaction	auxiliary interaction

Electron subshell labels

label	shell		label	shell		label	shell	
1	K	$(1s_{1/2})$	11	N2	$(4p_{1/2})$	21	O5	$(5d_{5/2})$
2	L1	$(2s_{1/2})$	12	N3	$(4p_{3/2})$	22	O6	$(5f_{5/2})$
3	L2	$(2p_{1/2})$	13	N4	$(4d_{3/2})$	23	O7	$(5f_{7/2})$
4	L3	$(2p_{3/2})$	14	N5	$(4d_{5/2})$	24	P1	$(6s_{1/2})$
5	M1	$(3s_{1/2})$	15	N6	$(4f_{5/2})$	25	P2	$(6p_{1/2})$
6	M2	$(3p_{1/2})$	16	N7	$(4f_{7/2})$	26	P3	$(6p_{3/2})$
7	M3	$(3p_{3/2})$	17	O1	$(5s_{1/2})$	27	P4	$(6d_{3/2})$
8	M4	$(3d_{3/2})$	18	O2	$(5p_{1/2})$	28	P5	$(6d_{5/2})$
9	M5	$(3d_{5/2})$	19	O3	$(5p_{3/2})$	29	Q1	$(7s_{1/2})$
10	N1	$(4s_{1/2})$	20	O4	$(5d_{3/2})$	30	outer shells	

Values and meanings of the particle labels ILB(.)

ILB(1)	generation of the particle; 1 for primary particles, 2 for their direct descendants, etc. Primary (source) particles are assumed to be labelled with ILB(1)=1 in the main program.
ILB(2)	kind KPAR of parent particle, only if ILB(1)>1 (secondary particles and "scattered" photons).
ILB(3)	interaction mechanism ICOL that originated the particle, only when ILB(1)>1.
ILB(4)	a non-zero value identifies particles emitted from atomic relaxation events and describes the atomic transition where the particle was released. The numerical value is $= Z \cdot 10^6 +$ IS1 $\cdot 10^4 +$ IS2 $\cdot$ 100 + IS3, where Z is the atomic number of the emitting atom and IS1, IS2 and IS3 are the labels of the active atomic electron shells. For instance, ILB(4) = 29010300 designates a K-L2 x ray from copper ($Z = 29$), and ILB(4) = 29010304 indicates a K-L2-L3 Auger electron from the same element.
ILB(5)	this label can be defined by the user; it is transferred to all descendants of the particle.

Fortran module PENELOPE_mod

```
      MODULE PENELOPE_mod
      SAVE  ! Saves all items in the module.
C  ****  Maximum number of materials in the geometry.
      INTEGER*4, PARAMETER :: MAXMAT=10
C  ****  Simulation parameters (input of subroutine PEINIT).
C  ----  Absorption energies, EABS(KPAR,MAT).
      DOUBLE PRECISION, DIMENSION (3,MAXMAT) :: EABS=50.0D0
C  ----  Electron/positron transport parameters.
      DOUBLE PRECISION, DIMENSION (MAXMAT) :: C1=0.01D0,C2=0.01D0
      DOUBLE PRECISION, DIMENSION (MAXMAT) :: WCC=1.0D2,WCR=1.0D2
C  ****  Size of the secondary stack.
      INTEGER*4, PARAMETER :: NMS=1000
C  ****  Energy interpolation, number of grid points.
      INTEGER*4, PARAMETER :: NEGP=200
C  ****  Global information on the material system (defined by
C        subroutine PEINIT).
C  ----  Number of materials present.
      INTEGER*4 :: NMAT
C  ----  Material densities and its reciprocals.
      DOUBLE PRECISION, DIMENSION (MAXMAT), SAVE :: DEN=1.0D0,
     1  RDEN=1.0D0
C  ****  Random-hinge slowing-down parameters (output from subroutines
C        JUMP and KNOCK).
      DOUBLE PRECISION :: E0STEP,DESOFT,SSOFT
      END MODULE PENELOPE_mod
```

```
SUBROUTINE PEINIT(EMAX,NMAT,IWR,INFO,PMFILE)
```

Reads material data file and prepares look-up tables. Simulation parameters must have been defined previously and loaded in module **PENELOPE_mod**

- **Input parameters:**

EMAX	...	maximum energy of particles Warning: for positrons, add ~ $2m_e c^2$ = 1.022 MeV
NMAT	...	number of different materials
IWR	...	output unit (open in the main program)
INFO	...	amount of information written in output unit (1, minimal; ...; 3, very detailed)
PMFILE	...	array of 20-character strings. The first **NMAT** elements are the filenames of the material data files. The file **PMFILE(M)** contains radiation interaction data for material **M** (the order is important!)

PENELOPE subroutines

SUBROUTINE CLEANS

Initializes the secondary stack (sets **NSEC=0**)
Must be called before starting the simulation of a new primary particle

SUBROUTINE START

Forces next interaction to be a soft one; it also verifies that **E>EABS**.
Must be called before starting a new particle (primary or secondary) and after each interface crossing.

SUBROUTINE JUMP(DSMAX,DS)

Determines the distance **DS** to the next interaction

DSMAX ... (input) maximum allowed path length for electrons and positrons. Recommended: < [body "thickness"/10]
DS ... (output) actual distance to travel

PENELOPE subroutines

SUBROUTINE KNOCK(DE,ICOL)

Simulates the next interaction event and modifies the state variables in module **TRACK_mod** (energy, direction, ...)
Modifies the energy and direction of movement of the particle

DE ... (output) deposited energy in the material
ICOL ... (output) type of interaction

SUBROUTINE SECPAR(LEFT)

Delivers the initial state of a secondary particle. This particle is removed from the secondary stack and loaded in module **TRACK_mod**

LEFT ... (output) number of particles in the stack at calling time. When **LEFT**=0, the shower simulation is completed.

SUBROUTINE STORES(E,X,Y,Z,U,V,W,WGHT,KPAR,ILB,IPOL)

Stores a particle in the secondary stack. The input arguments are the particle's state variables and the array of labels **ILB**. The values of **IBODY** and **MAT** are read from module **TRACK_mod**

Fortran module PENGEOM_mod

```
      MODULE PENGEOM_mod
C  ****  Geometry definition parameters and I/O quantities.
      SAVE  ! Saves all items in the module.
C  ----  Geometry array sizes.
C        Maximum numbers of surfaces, bodies, and limiting elements.
      INTEGER*4, PARAMETER :: NS=10000, NB=5000, NXG=250
C        Number of bodies in the material system (given by PENGEOM).
      INTEGER*4 :: NBODY
C  ----  Body aliases (user labels).
      CHARACTER*4 :: BALIAS(NB)='    '
C  ----  Body materials. MATER(KB) is the material in body KB.
      INTEGER*4 :: MATER(NB)=0
C  ----  Detector definition.
C        KDET(KB)=ID if body KB is part of detector ID.
      INTEGER*4 :: KDET(NB)=0
C  ****  Warning messages for accidental undershots or round-off errors
C        are issued when LVERB=.TRUE.
      LOGICAL :: LVERB=.FALSE.
C  ****  Last step features (output from subroutine STEP).
C  ----  Travelled path length, including segments in void volumes.
      DOUBLE PRECISION :: DSTOT
C  ----  Label of the last surface crossed by the particle before
C        entering a material body (defined only when NCROSS /= 0).
      INTEGER*4 :: KSLAST
      END MODULE PENGEOM_mod
```

PENGEOM subroutines

```
SUBROUTINE GEOMIN(PARINP,NPINP,NMAT,NBOD,IRD,IWR)
```

Reads the geometry definition file, initializes the geometry package and prints the file **geometry.rep**

```
SUBROUTINE LOCATE
```

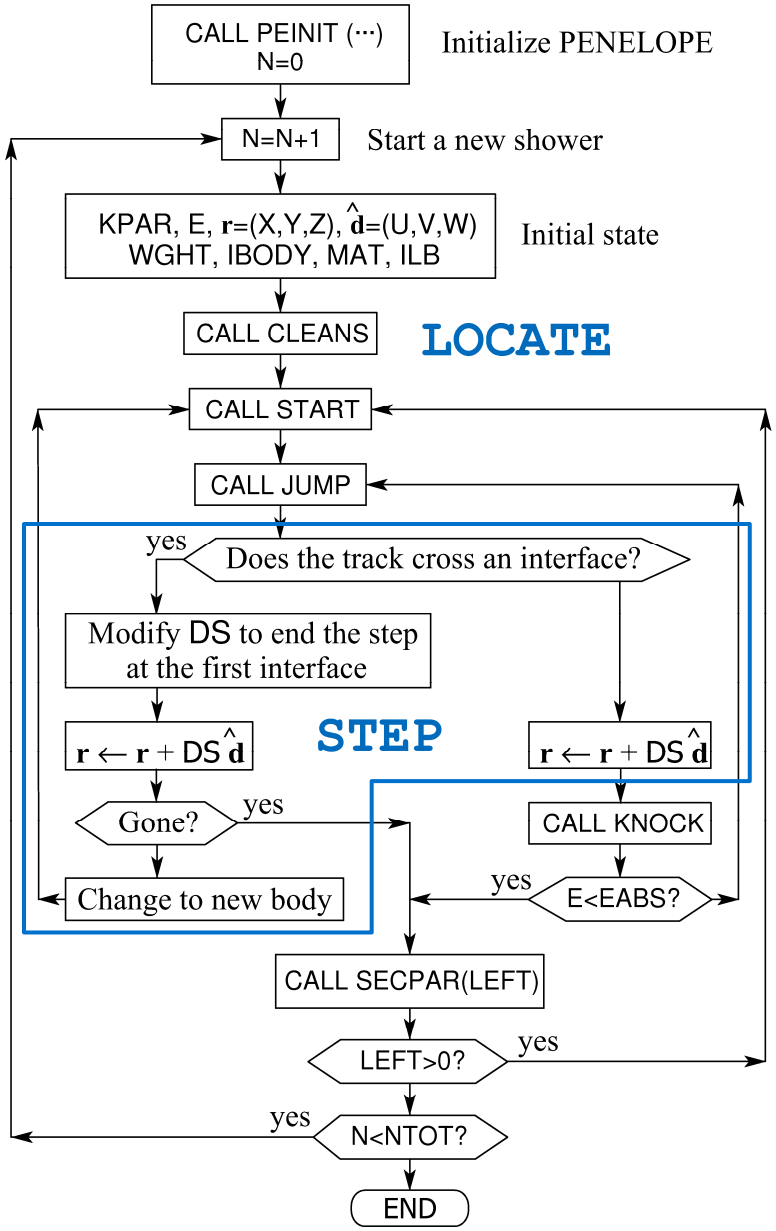
Determines the body **IBODY** that contains a point **(X,Y,Z)** and its material **MAT** (values delivered though module **TRACK_mod**)

```
SUBROUTINE STEP(DS,DSEF,NCROSS)
```

Performs the geometrical part of the tracking. Moves the particle and changes body and material numbers as appropriate. New values of the variables **X,Y,Z,IBODY,MAT** are delivered though **TRACK_mod**

DS	...	(input) path length to travel
DSEF	...	(output) travelled path length before leaving the initial material or completing the jump
NCROSS	...	(output) number of interfaces crossed

Structure of the main program




```
      PROGRAM MAIN
      CALL PEINIT(EMAX,NMATER,IWR,INFO,PMFILE)
      CALL
GEOMIN(PARINP,NPINP,NMATG,NBOD,IRD,IWR)
      N=0
C **** Start a new shower
 10   N=N+1
Cu Set the initial state variables.
      CALL LOCATE
      CALL CLEANS
C **** Start tracking in the material
 20   CALL START
 30   CALL JUMP(DSMAX(IBODY),DS)
      CALL STEP STEP(DS,DSEF,NCROSS)
      IF(MAT.EQ.0) THEN  ! The left the system.
        GO TO 40  ! Cu: Scoring?
      ENDIF
      IF(NCROSS.GT.0) GO TO 20
      CALL KNOCK(DE,ICOL)
Cu Score relevant quantities
      IF(E.GT.EABS(KPAR,MAT)) GO TO 30
 40   CONTINUE
Cu Score relevant quantities
C **** Any secondary left?
      CALL SECPAR(LEFT)
      IF(LEFT.GT.0) THEN
        GO TO 20  ! Cu: Scoring?
      ENDIF
      IF(N.LT.NTOT) GO TO 10
Cu Compute averages and write results
      END
```

penmain: a generic main program

The main program **penmain.f** is designed to solve a wide variety of problems. It operates as a "black box" and is completely controlled through several text (ascii) input files. **No programming is required.**

The geometry is described by using the package PENGEOM (constructive quadric geometry)

To define a new problem, we must usually prepare the following:

- A **geometry definition file** (PENGEOM format).
  Tools for editing and debugging the geometry definition file are available (simple viewers for Windows, the Java GUI PenGeomJar).
- The corresponding set of **material-data files**,
  which are generated by running the program **material**, or **tables**.
  The number label of a material is determined by its position in the **penmain** input file.
- The **penmain input file**, which defines the radiation source, the simulation parameters, the variance reduction methods employed, and the desired output information.

Layout of the `penmain` input file

The basic control of **`penmain`** is through a partially formatted input file

Each line in this file has the following structure:

- A 6-character keyword (columns 1-6), case-sensitive
- 1 blank column
- Alphanumeric information, free format (after column 7)
- The text after the input information is not read (reminder to the user)
- The keyword "`      `" (6 blanks) indicates comment lines (empty lines are not allowed!)

Notice that:

- The order of lines must be kept unaltered
- Lines marked with "**`*`**" can be repeated as many times as needed
- Groups marked with "**`&`**" can be repeated, but the order of lines within each group must be preserved

```
....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
TITLE  Title of the job, up to 65 characters.
       . (the dot prevents editors from removing trailing blanks)
       >>>>>>>> Source definition.
SKPAR  KPARP    [Primary particles: 1=electron, 2=photon, 3=positron]
       KPARP=0 activates a user-defined SOURCE model.
SENERG SE0                [Initial energy (monoenergetic sources only)]
SPECTR Ei,Pi                   [E bin: lower end and total probability] *
SGPOL  SP1,SP2,SP3           [Stokes parameters for polarised photons]
SPOSIT SX0,SY0,SZ0                          [Coordinates of the source]
SBOX   SSX,SSY,SSZ                             [Source box dimensions]
SBODY  KB                  [Active source body; one line for each body] *
SCONE  THETA,PHI,ALPHA                    [Conical beam; angles in deg]
SRECTA THETAL,THETAU,PHIL,PHIU       [Rectangular beam; angles in deg]
       .
       >>>>>>>> Input phase-space file (psf).
IPSFN  psf-filename.ext          [Input psf name, up to 20 characters] *
IPSPLI NSPLIT                                        [Splitting number]
WGTWIN WGMIN,WGMAX           [Weight window, RR & spl of psf particles]
EPMAX  EPMAX                   [Maximum energy of particles in the psf]
       .
....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
```

```
....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
       >>>>>>>> Material data and simulation parameters.
                Up to MAXMAT materials; 2 lines for each material.
MFNAME mat-filename.ext                  [Material file, up to 20 chars] &*
MSIMPA EABS(1:3),C1,C2,WCC,WCR               [EABS(1:3),C1,C2,WCC,WCR] &*
       .
       >>>>>>>> Geometry and local simulation parameters.
GEOMFN geo-filename.ext                  [Geometry file, up to 20 chars]
PARINP IP,PARINP(IP)                             [Replacement parameter] *
DSMAX  KB,DSMAX(KB)                [KB, maximum step length in body KB] *
EABSB  KB,EABSB(1:3,KB)    [KB, local absorption energies, EABSB(1:3)] *
       .
       >>>>>>>> Interaction forcing.
IFORCE KB,KPAR,ICOL,FORCER,WLOW,WHIG  [KB,KPAR,ICOL,FORCER,WLOW,WHIG] *
       .
       >>>>>>>> Bremsstrahlung splitting.
IBRSPL KB,IBRSPL                                 [KB,splitting factor] *
       .
       >>>>>>>> X-ray splitting.
IXRSPL KB,IXRSPL                                 [KB,splitting factor] *
       .
       >>>>>>>> Emerging particles. Energy and angular distributions.
NBE    EL,EU,NBE                         [Energy window and no. of bins]
NBANGL NBTH,NBPH           [Nos. of bins for the angles THETA and PHI]
       .
....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
```

```
....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
       >>>>>>>> Impact detectors (up to 25 different detectors).
       IPSF=0; no psf is created.
       IPSF=1; a psf is created (for only one detector).
       IDCUT=0; tracking is discontinued at the detector entrance.
       IDCUT=1; the detector does not affect the tracking.
       IDCUT=2; the detector does not affect tracking, the energy
                distribution of particle fluence (averaged over the
                volume of the detector) is calculated.
IMPDET EL,EU,NBE,IPSF,IDCUT      [E-window, no. of bins, IPSF, IDCUT] &
IDSPC  spc-impdet-##.dat               [Spectrum file name, 20 chars] &
IDPSF  psf-impdet-##.dat            [Phase-space file name, 20 chars] &
IDFLNC fln-impdet-##.dat       [Fluence spectrum file name, 20 chars] &
IDAGEL AGEL,AGEU,NAGE                  [Age interval and no. of bins] &
IDAGEF age-impdet-##.dat       [Age-distribution file name, 20 chars] &
IDBODY KB                       [Active body; one line for each body] &*
IDKPAR KPAR                [Kind of detected particles, one line each] &*
       .
       >>>>>>>> Outer angular detectors (up to 25).
ANGDET THETA1,THETA2,PHI1,PHI2        [Angular window, angles in deg] &
ADENER EL,EU,NBE                         [Energy window, no. of bins] &
ADSPC  spc-angdet-##.dat               [Spectrum file name, 20 chars] &
       .
....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
```

```
....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
 >>>>>>>> Energy-deposition detectors (up to 25).
ENDETC EL,EU,NBE                    [Energy window and number of bins] &
EDSPC  spc-enddet-##.dat               [Spectrum file name, 20 chars] &
EDBODY KB                        [Active body; one line for each body] &*
       .
       >>>>>>>> Absorbed dose distribution.
GRIDX  XL,XU,NDBX          [X coords of the box vertices, no. of bins]
GRIDY  YL,YU,NDBY          [Y coords of the box vertices, no. of bins]
GRIDZ  ZL,ZU,NDBZ          [Z coords of the box vertices, no. of bins]
GRIDR  RU,NDBR                [Radius of the dose volume, no. of bins]
       .
       >>>>>>>> Job properties.
RESUME dump1.dmp               [Resume from this dump file, 20 chars]
DUMPTO dump2.dmp                  [Generate this dump file, 20 chars]
DUMPP  DUMPP                                  [Dumping period, in sec]
       .
RSEED  ISEED1,ISEED2           [Seeds of the random-number generator]
NSIMSH DSHN                     [Desired number of simulated showers]
TIME   TIMEA                       [Allotted simulation time, in sec]
END                                  [Ends the reading of input data]
....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
```

Simple example. Point isotropic source in a sphere

```
....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
TITLE  Radial dose distribution within a water sphere.
       >>>>>>>> Source definition.
SKPAR  1          [Primary particles: 1=electron, 2=photon, 3=positron]
SENERG 1.0e4              [Initial energy (monoenergetic sources only)]
SPOSIT 0.0 0.0 0.0                          [Coordinates of the source]
SCONE  0.0 0.0 180.0                [Beam axis direction angles, in deg]
       >>>>>>>> Material data and simulation parameters.
MFNAME H2O.mat                      [Material definition file, 20 chars]
MSIMPA 1.0e2 1.0e2 1.0e2 0.0  0.0  0.0e2 -100   [EABSs,C1,C2,WCC,WCR]
       >>>>>>>> Geometry definition file.
GEOMFN sphere.geo                   [Geometry definition file, 20 chars]
PARINP 1 3.0e-4                           radius [Replacement parameter]
DSMAX  1 5.0e-7                     [KB, maximum step length in body KB]
       >>>>>>>> Dose distribution in a sphere.
GRIDR  3.0e-4 150              [Radius of the dose volume, no. of bins]
       >>>>>>>> Job properties
RESUME dump1.dat                 [Resume from this dump file, 20 chars]
DUMPTO dump1.dat                    [Generate this dump file, 20 chars]
DUMPP  60                                        [Dumping period, in sec]
NSIMSH 5.0e6                      [Desired number of simulated showers]
END
....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
```

Practical simulation

Let's assume that your problem can be solved by using **penmain**. You should do the following:

- Prepare your geometry definition file. Verify that the geometry is properly defined using the viewing-debugging programs **gview2d** and **gview3d**. Using **PenGeomJar** may ease the task
- Run the program **material** (or **tables**) to generate the material data files of all the materials present in your geometry
- Edit the **penmain** input file. Enter the different materials in the order assumed in the geometry definition file
- Assign tentative values to the simulation parameters for each material
- Perform preliminary short simulations, and check that the results do make sense. Consider optimizing the simulation parameters
- Consider the possibility of applying variance reduction techniques
- Run the simulation. Make sure that statistical uncertainties are small enough to trust the results