

The PENELOPE code system. Specific features and recent improvements

Francesc Salvat¹

¹*Universitat de Barcelona, Facultat de Física (ECM and ICC), Diagonal 645, 08028 Barcelona, Spain
Electronic mail: francesc.salvat@ub.edu, Fax: +34 934021174*

Since its first release, back in 1996, the Monte Carlo code system PENELOPE has evolved into both a flexible and reliable tool for describing coupled electron-photon transport in complex material structures. The present article contains an overview of the physical interaction models, particle tracking methods, geometry tools, and variance-reduction techniques implemented in PENELOPE. Recent refinements aimed at improving the accuracy of the code, and its stability under variations of user-defined simulation parameters, are also described. These include the use of reliable cross sections for the ionization of inner atomic electron shells by electron/positron impact, a reformulation of the random-hinge method, and the use of fuzzy quadric surfaces in the description of the geometry.

KEYWORDS: Monte Carlo, electron interactions, photon interactions, class II tracking, constructive quadric geometry.

I. Introduction

PENELOPE¹ is a general-purpose code system for the Monte Carlo simulation of coupled electron-photon transport^(1–3) that simulates electron-photon showers in material systems consisting of homogeneous bodies with arbitrary chemical compositions within the energy range from 50 eV to 1 GeV. The first version of PENELOPE was released in 1996 and, since then, the code has evolved through a series of improvements in the physics interaction models, the sampling algorithms, the description of the geometry, and the available variance-reduction techniques.

The program is written in Fortran and it is structured as a set of linked subroutine packages with minimal input/output. The core of the code system is the package `penelope.f`, which generates electron-photon showers in homogeneous materials. The packages `rita.f` and `penvared.f` contain, respectively, generic tools for random sampling from single-variate distributions, and subroutines that implement various variance-reduction techniques. The code system also includes a flexible subroutine package, `pengeom.f`, for automatic tracking of particles within quadric geometries (*i.e.*, material systems consisting of homogeneous bodies limited by quadric surfaces), and subroutines for tracking electrons and positrons in static electromagnetic fields. The distribution package also contains tools for generating and displaying total cross sections and other interaction data for different materials, and the program `SHOWER`, which displays electron/photon showers.

The simulation routines are invoked from a steering main program, which guides the evolution of the tracks and keeps score of relevant quantities. In principle, the user should provide a main program adapted to his/her particular problem. However, many practical problems, involving a wide variety of radiation sources in arbitrary quadric geometries, can be tackled by

means of the generic main program `penmain.f`, which allows the user to define impact detectors and energy-deposition detectors for extracting detailed information from the simulation. The operation of `penmain` is controlled from an input text file. A second main program, `pencyl.f`, performs simulations in cylindrical structures, providing more detailed information on the transport process (numbers of interactions of various kinds, path length distributions, position maps of emerging particles, etc.) which may be useful for tuning the simulation parameters.

The PENELOPE manual⁽³⁾ gives a detailed description of the physical interaction models, the tracking algorithms, and the structure and operation of the code system. The complete distribution package and the manual of PENELOPE are available from the OECD Nuclear Energy Agency Data Bank (<http://www.nea.fr>) and from the Radiation Safety Information Computational Center (<http://www-rsicc.ornl.gov>).

PENELOPE has been applied to a wide variety of problems in dosimetry and microdosimetry, radiotherapy, radiation protection, nuclear spectroscopy, electron microscopy, electron probe microanalysis, etc. A comprehensive comparison of simulation results with experimental data available from the literature⁽⁴⁾ for electrons and photons with initial energies ranging from a few keV up to 1 GeV demonstrated the reliability of both the adopted interaction models and the tracking algorithm.

In the present article we give an overview of the simulation program, and we describe various improvements introduced in the latest versions of the code system. Section II contains a brief description of the interaction physics models adopted in the program, and of tools provided to obtain interaction characteristics of materials in numerical and graphical form. The random-hinge method, employed to generate random trajectories of electrons and positrons, is considered in Section III. In Section IV we give a brief description of the `pengeom.f` routines and associated tools for defining and visualizing quadric

¹The name is an acronym for PENetration and Energy LOSS of Positrons and Electrons.

geometries. The variance-reduction methods implemented in PENELOPE are presented in Section V. Finally, the generic main program `penmain.f` is presented in Section VI, where we give results from an illustrative example.

II. Interaction models

PENELOPE implements the most reliable interaction models available, limited only by the required generality of the code. These models combine results from first-principles calculations, semi-empirical formulas and evaluated databases. Each interaction mechanism is described by a differential cross section (DCS) which is either defined numerically or given by an analytical formula with parameters fitted to available theoretical or experimental information. Details on the physics models can be found in the code manual⁽³⁾ and in the review article by Salvat and Fernández-Varea.⁽⁵⁾

Most of the considered DCSs pertain to free atoms or single-element materials; the DCSs for compounds and mixtures are obtained by means of Bragg's additivity rule (the "molecular" DCS is set equal to the sum of DCSs of all the atoms in a molecule). However, inelastic collisions of electrons and positrons are modeled in terms of the mass density and the mean excitation energy I of the material and, therefore, the DCSs for these interactions account for the aggregation state of the material in an approximate way.

1. Photon interactions

The considered interactions of photons are the following:

P1) *Coherent (Rayleigh) scattering*, described by means of the Born DCS with atomic form factors and angle-independent effective anomalous scattering factors taken from the LLNL Evaluated Photon Data Library.⁽⁶⁾

P2) *Incoherent (Compton) scattering*, which is simulated using the DCS calculated from the relativistic impulse approximation with analytical one-electron Compton profiles.⁽⁷⁾ This approximation consistently accounts for the effect of binding and Doppler broadening. These are manifestations of the fact that atomic electrons are bound (energy transfers less than the binding energy are forbidden) and, therefore, they move with a certain velocity distribution (and "see" the incoming photon shifted in energy by the Doppler effect). Total atomic cross sections for Compton scattering differ from those obtained from the Waller-Hartree approximation,⁽⁶⁾ mostly because the latter neglects Doppler broadening.

P3) *Photoelectric absorption*. The photoeffect is described by using total atomic cross sections, and partial cross sections for the K shell and L, M, and N subshells of neutral atoms which were calculated by the author using conventional first-order perturbation theory. Atomic wave functions were represented as single Slater determinants built with one-electron orbitals that are solutions of the Dirac equation for the Dirac-Hartree-Fock-Slater self-consistent potential.^(8,9) These cross sections practically coincide with those in the LLNL Evaluated Photon Data Library,⁽⁶⁾ although they are tabulated in a denser grid of energies to describe the structure of the cross section near ab-

sorption edges. A screening normalization correction, initially proposed by Pratt,⁽¹⁰⁾ has been included in the current version of the code. The initial direction of photoelectrons is sampled from Sauter's K-shell hydrogenic DCS.⁽¹¹⁾

P4) *Electron-positron pair production*. The total cross sections for pair (and triplet) production were obtained from the xcom program of Berger *et al.*⁽¹²⁾ The initial kinetic energies of the produced particles are sampled from the Bethe-Heitler DCS for pair production, with exponential screening and Coulomb correction, empirically modified to improve its reliability for energies near the pair-production threshold.

The inverse mean free path (IMFP), also known as the attenuation coefficient, of photons of energy E is given by

$$\begin{aligned}\mu(E) &= N\sigma_{\text{tot}}(E) \\ &= N[\sigma_{\text{Ra}}(E) + \sigma_{\text{Co}}(E) + \sigma_{\text{ph}}(E) + \sigma_{\text{pp}}(E)],\end{aligned}\quad (1)$$

where N is the number of atoms or molecules per unit volume, and σ_{tot} is the total cross section per atom or molecule, which is the sum of contributions from Rayleigh (Ra) and Compton (Co) scattering, photoabsorption (ph) and pair production (pp).

The current version of the code allows the simulation of polarized photon beams, with the state of polarization described by means of the Stokes parameters.⁽³⁾ However, secondary photons (*i.e.*, characteristic x rays and bremsstrahlung photons emitted by electrons or positrons) are assumed to be unpolarized.

2. Electron and positron interactions

The considered interactions of electrons and positrons are:

E1) *Elastic collisions*. These are described by numerical DCSs obtained from Dirac partial-wave analysis for the electrostatic potential of the target atom obtained from Dirac-Fock atomic electron densities, with the exchange potential of Furness and McCarthy for electrons. These cross sections were calculated using the program ELSEPA.^(13,14)

E2) *Inelastic collisions*, which are simulated by means of the Born DCS obtained from the Sternheimer-Liljequist generalized oscillator strength model,^(15,16) including the density-effect correction. The excitation spectrum is modeled as a discrete set of delta oscillators. Excitations of each electron subshell are represented by a single oscillator, with strength equal to the number of electrons in the subshell and resonance energy determined by the subshell binding energy. The resonance energies are scaled so as to reproduce the mean excitation energy I recommended in the ICRU Report 37.⁽¹⁷⁾ Thus, collision stopping powers calculated from this model agree closely with the tabulations in the ICRU Report 37. Optionally, the DCS can be renormalized to reproduce the collision stopping power read from the input file.

E3) *Ionization of inner shells by impact of electrons and positrons*. We consider as inner atomic electron shells the K shell, and the L, M, and N subshells that have binding energies larger than about 50 eV. Because the Sternheimer-Liljequist generalized oscillator strength gives too coarse an approximation for inner shells, PENELOPE simulates inner-shell ionization

as a separate process (which does not alter the energy and the direction of the projectile particle) with numerical total cross sections calculated by Bote and Salvat⁽¹⁸⁾ using the distorted-wave (first) Born approximation with the Dirac-Hartree-Fock-Slater self-consistent potential. This approach has the advantage of yielding a nearly correct number of ionizations per unit path length, without altering the modeling of inelastic collisions, but at the expense of occasionally producing negative doses (x rays leave from sites where no energy was deposited).

In the latest version of the code we have changed strategy and consider ionizations of inner shells as real inelastic collisions. This is accomplished by setting the total cross section for collisions with electrons in each inner shell equal to the theoretical value obtained from the distorted-wave Born approximation, and renormalizing the total cross sections of outer subshells so as to keep the value of the collision stopping power unaltered.

E4) *Bremsstrahlung emission*. The energy of the emitted photon is sampled from numerical energy-loss spectra derived from the scaled cross-section tables of Seltzer and Berger,^(19,20) optionally renormalized to reproduce the radiative stopping power read from the input file. The intrinsic angular distribution of emitted photons is described by an analytical expression — an admixture of two “boosted” dipole distributions^{—(21)} with parameters determined by fitting the calculated partial-wave angular distributions of Kissel *et al.*⁽²²⁾

E5) *Positron annihilation*, which is simulated by using the Heitler DCS for in-flight two-photon annihilation with electrons at rest.

The distribution package includes an extensive database of DCSs and total cross sections, for all elements in the periodic system, from hydrogen to einsteinium, covering the energy range from 50 eV to 1 GeV. PENELOPE reads the interaction properties for each specific material from a formatted ASCII file, which should have been generated previously by running the program `material.f`. The user can access the most relevant information in a material data file by using the program `tables.f`, which reads material files and generates tables of interaction data (cross sections, mean free paths, stopping powers, ranges, ...) as functions of energy. These tables can be plotted, *e.g.*, by using GNUPLOT, a free plotting software. Figure 1 displays IMFPs of photons in lead obtained with the program `tables`.

The PENELOPE manual⁽³⁾ contains a detailed description of the sampling algorithms adopted to simulate the different interactions from the associated DCSs. Continuous distributions are sampled by means of the adaptive algorithm RITA (Rational Inverse Transform with Aliasing); Walker’s aliasing method is utilized to sample discrete distributions with large numbers of possible outcomes. These sampling methods are both robust and fast.

As mentioned above, PENELOPE can follow particles with energies down to 50 eV, which is the lower limit of the interval covered by the interaction database. However, the approximations underlying the interaction models are expected to be valid only for energies larger than about 1 keV. Consequently, simulation results for energies less than this value should be

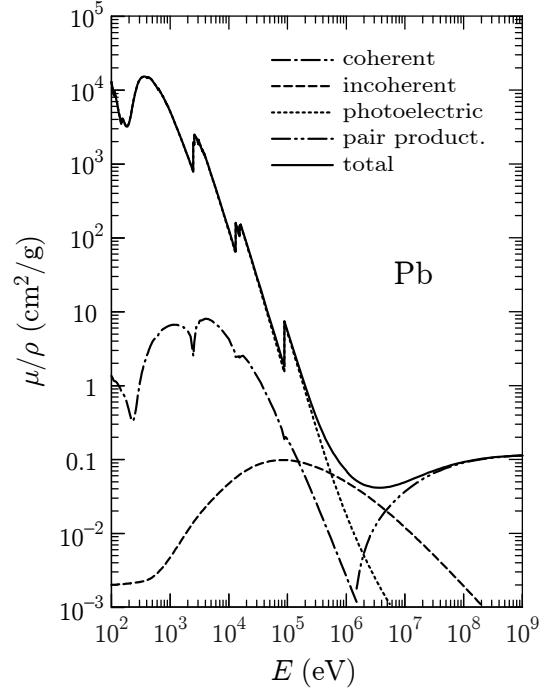


Figure 1: IMFPs of photons in lead, extracted from the PENELOPE database using the program `tables`.

regarded as semi-quantitative.

III. Tracking algorithm

Particle histories are simulated from the initial energy down to the absorption energies E_{abs} selected by the user, at which the particles are considered to be effectively absorbed in the material. In practically all interactions (the only exceptions are Rayleigh scattering of photons and elastic scattering of electrons and positrons) secondary particles are emitted as a direct result of the interaction, and also in the relaxation of atoms following inner-shell ionization. PENELOPE simulates the emission of characteristic x rays and Auger electrons with energies larger than E_{abs} that result from vacancies produced in inner shells of atoms by photoelectric absorption and Compton scattering of photons and by electron or positron impact. The relaxation of excited ions is followed until all vacancies have migrated to subshells with binding energies less than E_{abs} . The adopted transition probabilities, as well as the energies of Auger electrons, were extracted from the LLNL Evaluated Atomic Data Library.⁽²³⁾ The energies of K and L x-ray lines are taken from the review of Deslattes *et al.*,⁽²⁴⁾ while those of M and N lines are from Bearden’s compilation.⁽²⁵⁾

The simulation of photons follows the usual detailed procedure, where all interaction events in a photon history are simulated in chronological succession. That is, a photon of energy E starts from a certain position, $\mathbf{r}_0$, determined in accordance to the characteristic of the radiation source, moving in a direction defined by the unit vector $\hat{\mathbf{d}}_0$. The physics simulation routines determine the distance s to the next interaction, assuming the medium is infinite, by random sampling from the

exponential distribution,

$$p(s) = \mu(E) \exp[-s\mu(E)], \quad (2)$$

where $\mu(E)$ is the IMFP of the photon, Eq. (1). The program then propagates the photon the distance s along the ray, *i.e.*, to a position $\mathbf{r} = \mathbf{r}_0 + s\hat{\mathbf{d}}_0$, where the next interaction takes place. The physics routines are invoked again to sample the interaction type and to simulate the interaction from the corresponding DCSs. In Rayleigh and Compton scattering, the photon is absorbed by electrons in the medium and a second photon is emitted with energy E' (equal or less than E). When $E' > E_{\text{abs}}$, the surviving photon is followed by repeating these steps. Photoabsorption and pair production terminate the photon history. Each history is a sequence of a relatively small number of free flights and interactions, which can be simulated rapidly.

This simulation scheme can be readily adapted to real situations, where photons move within material structures, which may consist of several regions of different compositions separated by well-defined surfaces (interfaces). Of course, we must have the photon located, *i.e.*, we need to know the composition of the material at its position. The path length s to the next interaction is sampled using the IMFP in the current material and, if the ray segment does not leave the current material, the photon is moved a distance s and the interaction is simulated. When the ray segment intersects an interface, the photon is halted just after crossing the interface and tracking is resumed with the cross sections of the new material. Thus, to track photons in complex geometries, we only need to determine the distances at which a ray intersects the interfaces.

1. Random-hinge method

Unfortunately, detailed simulation of high-energy electrons and positrons is impractical because these particles undergo a very large number of interactions in the course of their slowing down². The most characteristic feature of PENELOPE is the use of a mixed (class II) scheme for electrons and positrons⁽²⁶⁾ that allows the generation of each particle trajectory with a relatively small number of computational steps. Hard interactions, with scattering angle θ or energy loss W larger than certain cutoff values, are simulated in a detailed way, *i.e.*, by random sampling from the corresponding restricted DCSs. The path length to the next hard interaction is sampled according to the energy-dependent mean free path for hard interactions. The combined effect of all soft interactions that occur along the trajectory segment between two consecutive hard interactions (step) is simulated as a single “artificial” soft event (a hinge) where the particle loses energy and changes its direction of motion. The energy loss and the angular deflection at the hinge are generated according to a multiple-scattering approach which leads to distributions for the energy-loss and the angular deflection (caused by multiple soft interactions) that have the correct means and variances.

In PENELOPE, the energy loss, the angular deflection and the lateral displacement due to the multiple soft collisions in a

step of length s are simulated by means of the random-hinge method,⁽²⁶⁾ which proceeds as follows:

- 1) The electron first moves a random distance τ , which is sampled uniformly in the interval $(0, s)$, in the initial direction.
- 2) The artificial event then takes place, in which the electron changes its direction of movement and its energy as prescribed by the multiple-scattering distributions.
- 3) Finally, the electron moves a distance $s - \tau$ in the new direction.

This tracking algorithm was originally designed to allow the code to operate as in detailed simulations, *i.e.*, the transported particle is moved in straight steps, and the energy and direction of movement change only through discrete interactions (hard interactions and hinges). The random-hinge method is well suited to simulate multiple-scattering processes in complex material structures consisting of homogeneous bodies with well-defined interfaces. In these geometries, when the track crosses an interface, see Fig. 2, we simply stop it at the crossing point, and resume the simulation in the new material. Because the distance τ from the beginning of the step to the hinge is distributed uniformly in $(0, s)$, the particle reaches the interface with nearly correct average energy and direction (see Ref.⁽³⁾).

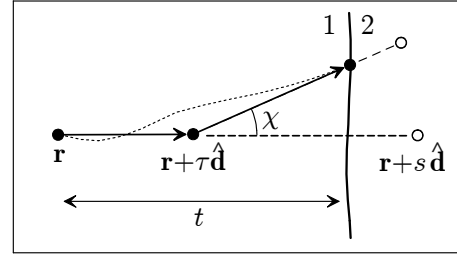


Figure 2: Interface crossing with the random-hinge method.

The random-hinge method works well when the average deflection and the average lateral displacement due to soft interactions along a step is small. In spite of its simplicity, this method competes in accuracy and speed with other, much more sophisticated transport algorithms (see Ref.,⁽²⁷⁾ and references therein). It seems that the randomness of the hinge position leads to correlations between the angular deflection and the displacement that are close to the actual correlations.

It has been observed recently that simulations of electron transport in multilayers of various materials become sensitive to user parameters when step lengths are comparable to the layer thickness because the random-hinge method overestimates energy straggling in steps that end at an interface. In the next version of the code this deficiency will be mitigated by considering that the energy loss due to soft interactions is deposited continuously and uniformly along the step.

2. Simulation parameters

A key feature in the practical implementation of the random-hinge method is the definition of the *energy-dependent* cutoffs. In PENELOPE these cutoffs are determined by a small set of user parameters, independent of the energy of the particle and which may take different values for the various materials in the geom-

²On average, a high-energy electron or positron loses in each interaction an energy of the order of a few tens of eV.

etry. These user parameters have a direct impact on both the accuracy and the efficiency of the simulation.

Energy-loss events are classified by introducing cutoff values W_{cc} and W_{cr} , and considering inelastic collisions (in) with energy loss $W < W_{cc}$ and emission of bremsstrahlung photons (br) with $W < W_{cr}$ as soft interactions. The mean free paths $\lambda_{in}^{(h)}$ and $\lambda_{br}^{(h)}$ between hard inelastic collisions and hard radiative events are then given by

$$\frac{1}{\lambda_{in}^{(h)}} = \mathcal{N} \int_{W_{cc}}^E \frac{d\sigma_{in}}{dW} dW \quad (3)$$

and

$$\frac{1}{\lambda_{br}^{(h)}} = \mathcal{N} \int_{W_{cr}}^E \frac{d\sigma_{br}}{dW} dW, \quad (4)$$

where the integrands are the corresponding energy-loss DCSs.

The cutoff deflection angle θ_c , which separates hard and soft elastic collisions, is defined by two user parameters, C_1 and C_2 , which typically should be given small values, between 0 and 0.05. These two parameters are used to fix the mean free path between hard elastic events, $\lambda_{el}^{(h)}$, defined by

$$\frac{1}{\lambda_{el}^{(h)}} = \mathcal{N} \int_{\theta_c}^{\pi} \frac{d\sigma_{el}}{d\Omega} 2\pi \sin \theta d\theta, \quad (5)$$

where the integrand is the DCS for elastic scattering per unit solid angle and θ is the polar scattering angle. In PENELOPE we set

$$\lambda_{el}^{(h)} = \max \left\{ \lambda_{el}, \min \left[C_1 \lambda_{el,1}, C_2 \frac{E}{S} \right] \right\} \quad (6)$$

where $\lambda_{el,1}$ is the first transport mean free path, defined by

$$\frac{1}{\lambda_{el,1}} = \mathcal{N} \int_{\theta_c}^{\pi} (1 - \cos \theta) \frac{d\sigma_{el}}{d\Omega} 2\pi \sin \theta d\theta, \quad (7)$$

and

$$S = \mathcal{N} \int_0^E W \left(\frac{d\sigma_{in}}{dW} + \frac{d\sigma_{br}}{dW} \right) dW \quad (8)$$

is the stopping power, *i.e.*, the average energy loss per unit path length due to both inelastic collisions and bremsstrahlung emission. It is worth recalling that $\lambda_{el}^{(h)}$ is equal to the average length of a step between hard elastic collisions. The average angular deflection of the electron trajectory at the end of a step of length $\lambda_{el}^{(h)}$ can be evaluated by using Lewis' multiple-scattering theory⁽²⁸⁾ which, ignoring energy losses along the step, gives

$$1 - \langle \cos \theta \rangle = 1 - \exp \left(- \frac{\lambda_{el}^{(h)}}{\lambda_{el,1}} \right) \simeq \frac{\lambda_{el}^{(h)}}{\lambda_{el,1}} \lesssim C_1. \quad (9)$$

That is, C_1 sets an upper limit for the average angular deflection at the end of the step. On the other hand, the average energy loss at the end of the step is

$$\langle E - E_{final} \rangle \simeq \lambda_{el}^{(h)} S \lesssim C_2 E, \quad (10)$$

so that C_2 limits the average fractional energy loss along the step. Evidently, an increase of C_1 or C_2 , leads to increased

values of both the mean free path between hard events, $\lambda_{el}^{(h)}$, and the cutoff angle, θ_c defined by Eq. (5), in certain energy ranges.

It should be noted that C_1 and C_2 act on different energy domains. This is illustrated in Fig. 3, where the lengths λ_{el} , $\lambda_{el,1}$ and E/S for electrons in lead are represented as functions of the kinetic energy. The mean free path $\lambda_{el}^{(h)}$ for hard elastic events, determined from the prescription (6) with $C_1 = C_2 = 0.05$ is also plotted. For low energies, $\lambda_{el}^{(h)} = \lambda_{el}$ and the simulation is purely detailed ($\theta_c = 0$). For intermediate energies, $\lambda_{el}^{(h)} = C_1 \lambda_{el,1}$, whereas $\lambda_{el}^{(h)} = C_2 E/S$ in the high-energy domain. From Fig. 3 it is clear that increasing the value of C_2 does not have any effect on the simulation of electron tracks with initial energies that are less than ~ 10 MeV.

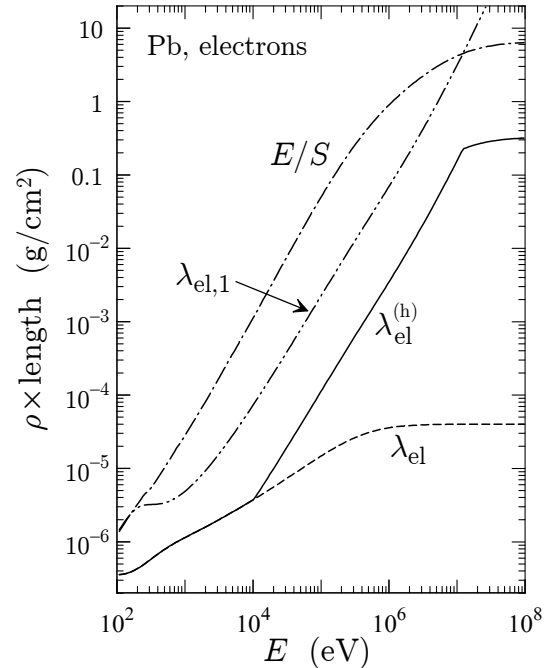


Figure 3: Elastic mean free path λ_{el} , first transport mean free path $\lambda_{el,1}$ and $E/S(E)$ for electrons in lead. The solid line represents the mean free path between hard elastic events $\lambda_{el}^{(h)}$ obtained from Eq. (6) with $C_1 = C_2 = 0.05$.

It is worth mentioning that the step length to the next interaction does not follow the usual exponential distribution because the IMFP for hard interactions depends on the energy of the particle, which decreases with the traveled path length due to the soft interactions. An associated difficulty arises from the energy dependence of the quantities that define the multiple-scattering distributions of the energy loss and the angular deflection at the hinge. PENELOPE utilizes an elaborate strategy to account for variations of the particle energy along a step, which rests on the requirement that the energy loss due to soft events along the step is bounded. This requirement is met by introducing a maximum step length, s_{max} , which is set equal to four times the mean free path between hard events (only 2 per cent of the sampled steps have lengths that exceed this value). The program effectively limits the step length by truncating longer steps with a delta interaction, which does nothing but stop the particle and the program flow. Incidentally, limiting the step

length is also required for controlling the accuracy of simulated trajectories of electrons and positrons in electromagnetic fields.

As already mentioned, the energy loss and the angular deflection due to soft interactions along a step are sampled from artificial distributions with the correct first and second moments (see Ref.⁽³⁾). The details of these distributions are irrelevant when the particle undergoes about ten or more interactions because the distributions of the accumulated energy loss and angular deflection become nearly Gaussian and, consequently, they are determined by the first and second moments of the artificial distributions. When the particle is in a thin region, it is advisable to use a small value of $s_{\max}$ to make sure that the number of hinges within the material is sufficient to smear away the unphysical details of the adopted artificial distributions.

Summarizing, the mixed simulation scheme is defined by the user parameters E_{abs} , C_1 , C_2 , W_{cc} , W_{br} , and $s_{\max}$ which determine the computer time needed to simulate each electron or positron track. The simulation gets slower when any of these parameters is reduced. Simulation results are found to be practically insensitive to the user parameters, provided only their values are not too large. As a rule of thumb, C_1 and C_2 should be smaller than about 0.05 (values larger than 0.2 are not allowed). The cutoff energies W_{cc} and W_{br} have a visible effect on simulated energy spectra; they should be smaller than the energy bin width. Our recommendation is to set these cutoff energies equal to about 1 keV (or one hundredth of the initial energy of primary particles, whichever is the smallest) because simulation speed remains practically constant when they are given larger values. The parameter $s_{\max}$ is defined internally and only needs to be modified in the case of thin bodies; a value of the order of one tenth of the body thickness is usually adequate.

It is worth noting that when the parameters C_1 , C_2 , and W_{cc} are set to zero, our class II scheme produces a purely detailed (*i.e.*, nominally exact) simulation of elastic and inelastic collisions. Bremsstrahlung emission cannot be handled in a strictly detailed way because the DCS diverges at zero photon energy; PENELOPE performs an almost detailed simulation of radiative events by setting $W_{\text{br}} = 10$ eV, and disregarding the emission of photons with lower energies. An obvious advantage of class II schemes is that we can numerically verify their accuracy and stability under variations of the user parameters by comparing simulation results with those of a detailed simulation.

3. The program SHOWER

Monte Carlo simulation has proven to be a valuable tool for education. In the past, radiation physics used to be considered as a tough subject, mostly because high-energy radiation is well outside the realm of daily experience. Nowadays, by simply running a transport simulation code on a personal computer we can learn more than from tens of obscure empirical formulas and numerical tables, and eventually “understand” many aspects of radiation transport (those for which we have run the simulation code and “digested” the results).

The PENELOPE package includes an executable binary file named SHOWER that generates electron-photon showers within a slab, of one of 280 pre-defined materials and displays them

(projected) on the computer screen plane. The current version operates only under Microsoft Windows. The program is self-explanatory, and requires only a small amount of information from the user, which is entered from the keyboard, in response to prompts from the program. Electron, photon and positron tracks are displayed in different colors and intensities that vary with the energy of the particle. We should mention that the maximum number of showers that can be plotted in a single shot is limited to 50 because the screen may become too cluttered. Generating this small number of showers takes a short time, of the order of a few seconds, even on modest personal computers (provided only that absorption energies are sensibly chosen).

Once on the graphical screen, the view plane can be rotated about the horizontal screen axis by typing “r” and the rotation angle in degrees; the screen plane can also be rotated progressively, by 15 deg steps, by pressing the “enter” key repeatedly. Entering the single-character command “n” erases the screen and displays a new bunch of showers. Observation of single showers projected on a revolving plane gives a truly three-dimensional perspective of the transport process.

IV. The geometry package PENGEO

The subroutine package `pengeom.f` provides a complete set of tools to handle geometrical aspects in Monte Carlo simulations of radiation transport. PENGEO automatically tracks particles through the material structure, independently of the details of the physics models adopted to describe the interactions. The definition of the geometry and the numerical tracking algorithm are tailored to optimize simulation speed.

The PENGEO subroutines are designed for detailed simulations of photon and electron transport, where all individual interactions are simulated sequentially, and for mixed (class II) simulations of charged particles, where the effect of soft interactions is described by the random-hinge method. These simulation schemes find no difficulties with interface crossings because we only need to calculate intersections of particle rays with interfaces. This is at variance with condensed (class I) schemes, which require computing the distance of the particle to the nearest interface (see, *e.g.*, Ref.⁽²⁹⁾), a much more involved operation. In addition, physical and geometrical operations are effectively decoupled using both detailed and class II schemes.

Given a function $F(\mathbf{r})$, assumed to be continuous and differentiable, the equation $F(\mathbf{r}) = 0$ defines a surface in implicit form. A surface divides the space into two exclusive regions that are identified by the sign of $F(\mathbf{r})$, the *surface side pointer*, SP. A point with coordinates $\mathbf{r} = (x, y, z)$ is said to be inside the surface if $F(\mathbf{r}) \leq 0$ (SP = -1), and outside it if $F(\mathbf{r}) > 0$ (SP = +1). The surface itself [*i.e.*, the set of points such that $F(\mathbf{r}) = 0$] is the boundary of the two regions. Note that the equation $-F(\mathbf{r}) = 0$ defines the same surface, but with the inside replaced by the outside.

To reduce the numerical work needed to calculate interface crossings, it is convenient to use surfaces expressed by simple analytical functions. In the PENGEO package, all limiting sur-

faces are assumed to be quadrics given by the implicit equation

$$F(\mathbf{r}) = A_{xx}x^2 + A_{xy}xy + A_{xz}xz + A_{yy}y^2 + A_{yz}yz + A_{zz}z^2 + A_x x + A_y y + A_z z + A_0 = 0, \quad (11)$$

which includes planes, spheres, cylinders, cones, ellipsoids, paraboloids, hyperboloids, etc. Quadric surfaces are flexible enough to model many man-made material structures and to approximate more general surfaces.

The material systems considered in PENGEO M consist of a number of homogeneous bodies B_k determined by their limiting surfaces S_i and compositions (materials). Each surface S_i is specified by giving its equation $F_i(\mathbf{r}) = 0$, and the volume of elementary bodies can be defined simply by giving the side pointers $SP_i(B_k)$ of all the surfaces S_i that limit that volume. However, this is not always possible or convenient for the user. It is more expedient to assume that bodies are defined in “ascending” exclusive order so that previously defined bodies effectively delimit the new ones; this implies that overlaps between different bodies are not permitted. Figure 4 shows a simple geometry consisting of 2 bodies B_k ($k = 1, 2$), a sphere with a cylindrical inset, limited by four surfaces S_i ($i = 1$ to 4). In this case, the cylinder is defined first (as the volume limited by surfaces 1, 2, and 3) so that it automatically limits the volume of the hollow sphere (limited by surface 4, and the cylinder). It is impossible to define the hollow sphere as a single body by means of only its limiting surfaces because surfaces 1 and 2 of the cylinder do not limit body 2 (the hollow sphere) in all their extensions.

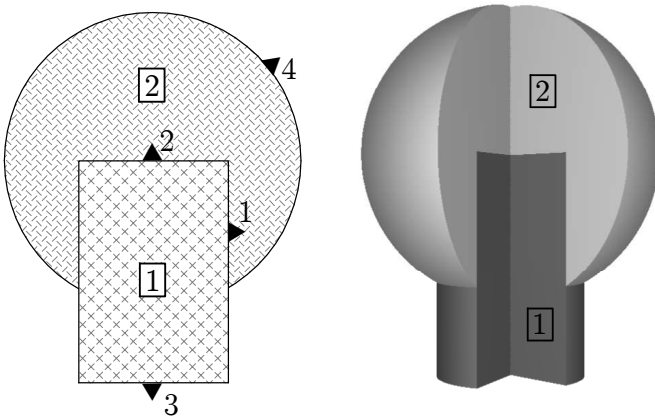


Figure 4: Left: Schematic representation of a simple geometry, a sphere with a cylindrical inset. The solid triangles indicate the outside of the surfaces ($SP = +1$); number labels in squares indicate bodies. Right: Three-dimensional view with an excluded wedge to show the inner structure.

The “location” of a particle is specified by giving its *position coordinates* $\mathbf{r}$ and the body B_k where it is moving. Given the initial position of a particle, $\mathbf{r}_0$, to determine the body that contains it we should calculate the SPs of all limiting surfaces and explore the set of bodies in ascending order to find the one with the right SPs. For complex systems, with a large number of limiting surfaces, this blind search may be quite lengthy. The searching work can be reduced by simply disregarding those

elements of the geometry that cannot be reached in a single step (*e.g.*, bodies that are “screened” by other bodies or too far apart). In PENGEO M this is accomplished by grouping sets of bodies together to form *modules*.

A module is defined as a connected volume, limited only by quadric surfaces that contains one or several bodies. A module can also contain other modules, which will be referred to as *submodules* of the first. The volume of a module is filled with a homogeneous material, which automatically fills the cavities of the module (*i.e.*, volumes that do not correspond to a body or to a submodule); these filled cavities are considered as a single new body. For simplicity, modules are required to satisfy the following conditions: 1) the bodies and submodules of a module must be completely contained within the parent module (*i.e.*, it is not allowed to have portions of bodies or submodules that lie outside the module) and 2) a submodule of a module cannot overlap with other submodules and bodies of the same module (this is necessary to make sure that a particle can only enter or leave a module through its limiting surfaces).

In practical simulations with finite geometries, the tracking of a particle should be discontinued when it leaves the material system. In PENGEO M this is done automatically by assuming that the complete system is contained within a single convex module, the enclosure, which comprises the whole system. It is also convenient (but not necessary) to require that the enclosure has a finite volume, so that all rays starting from any point within the volume of the enclosure do intersect one of its limiting surfaces at a finite distance. When an enclosure is not defined by the user (*i.e.*, when the geometry consists of several separate modules and/or bodies that are not inside a module), PENGEO M defines the enclosure as a large sphere of 10^7 cm radius, centered at the origin of coordinates. It is assumed that there is a perfect vacuum outside the enclosure and in any inner volume that is not a body or a filled module. If the geometry definition contains bodies that extend beyond the enclosure, they are truncated and only the parts inside the enclosure are retained. Hence, particles that leave the enclosure will never return to the material system.

The PENGEO M routines consider each module as the mother of its bodies and submodules, and as a daughter of the module that contains it. We thus have a kind of genealogical tree with various generations of modules and bodies. The first generation reduces to the enclosure (which is the only motherless module). The members of the second generation are bodies and modules that are daughters of the enclosure. The n -th generation consists of modules and bodies whose mothers belong to the $(n - 1)$ -th generation. Each module is defined by its limiting surfaces (which determine the border with the external world) and those of their descendants (which determine the module’s internal structure); this is not true for bodies (childless members of the tree), which can be limited either by surfaces, by other sister bodies or by a combination of both. A well-ramified tree of modules simplifies the location of particles because when a particle leaves a module it can only enter one of its sister modules and bodies or move to the mother module.

When a particle starts its motion, the geometry routines should first locate it, and subsequently move it in straight flights, of length prescribed by the physics routines, until the trajectory

intersects one of the limiting surfaces of the current body. A particle that starts from a point $\mathbf{r}_0$ moving in a direction defined by the unit vector $\hat{\mathbf{d}}$ intersects a surface $F(\mathbf{r}) = 0$ after traveling a *positive* distance s , which is a solution of the master equation

$$F(\mathbf{r}_0 + s\hat{\mathbf{d}}) = 0. \quad (12)$$

In the case of quadric surfaces, the function on the left-hand side becomes a second-degree polynomial in s , and the distances to the intersections are obtained by simply solving a quadratic equation.

When a trajectory segment intersects a limiting surface, the program moves the particle a distance s , and places it “at the surface”. Even with quadric surfaces, we can find numerical ambiguities when locating a particle that is close to a surface: because of the limited accuracy of floating-point numbers in a computer, we may be unable to assert with confidence whether the particle is inside or outside the surface. Indeed, similar ambiguities are found whenever a particle crosses an interface; due to round-off errors, the value of the surface function at the calculated intersection point may have either sign, although in this case we do know that the particle has just passed the interface. To get rid of these kinds of ambiguities, all surfaces are treated as fuzzy, that is, a surface swells or shrinks very slightly when the particle crosses it so as to ensure that the particle is at the correct side of the surface. Of course, mathematical surfaces are not altered; instead, we define the SP of a given surface at the particle position $\mathbf{r}_0$ by considering the relative motion of the particle with respect to the surface. Notice that to apply this strategy we must know the direction of motion $\hat{\mathbf{d}}$ of the particle.

The current version of PENGEOOM can describe very complicated systems with up to 5,000 bodies and 10,000 limiting surfaces, including nanometric structures. When the tree of modules is well ramified, the speed of the geometry operations is largely independent of the complexity of the whole material system.

The geometry is defined from a formatted input file, to be provided by the user. A pair of computer programs named GVIEW2D and GVIEW3D have been written to visualize the geometry and to help the user to debug the definition file. These codes generate two- and three-dimensional 24-bit color images of the system using specific graphics routines. The executable codes included in the distribution package run on personal computers under Microsoft Windows. The most characteristic (and useful) feature of these viewers is that displayed pictures are generated by using the PENGEOOM subroutines and, therefore, what we see on the screen is what is passed to the simulation code. Running these programs, errors and inconsistencies in the geometry definition file that would affect the results of actual simulations are readily identified.

Although the preparation of geometry definition files is a routine task, the manual editing of a long formatted file may become very tedious. To facilitate the work, we have developed a Java program, named PENGEOOM.JAR, which provides a graphical user interface (Fig. 5), from where we can edit the geometry definition files and run the two- and three-dimensional viewers, which open separate graphics windows. The rendering of the geometry files is performed by using the same algorithms as in

the Windows programs GVIEW2D and GVIEW3D. The Java program PENGEOOM.JAR and the Fortran routines PENGEOOM constitute a complete geometry toolbox that can be utilized in any Monte Carlo code performing detailed or class II simulation. These tools will be made available as a separate package.

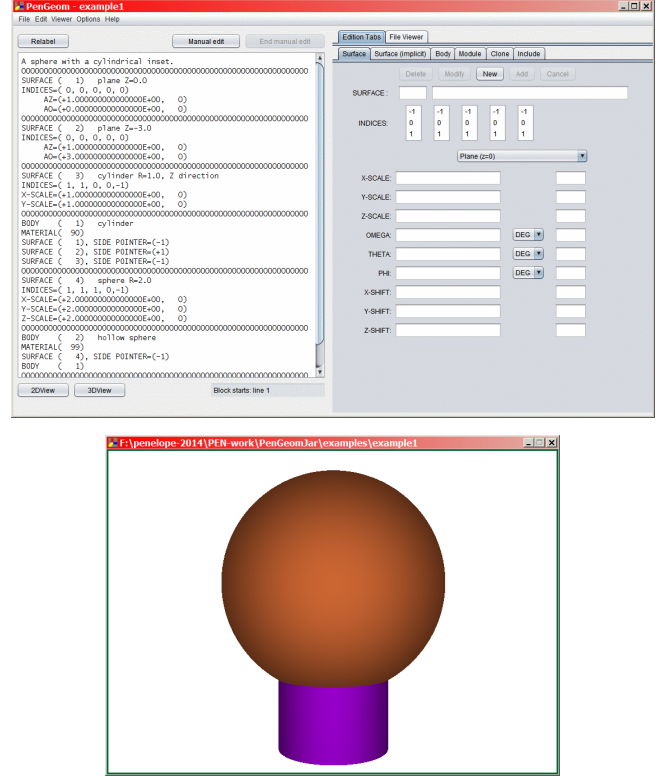


Figure 5: Main window of the GUI PENGEOOM.JAR, and three-dimensional viewer.

V. Variance reduction

Although all results from Monte Carlo calculations are affected by statistical uncertainties, these can be evaluated systematically provided only that the number of contributions to the scores is large enough.^(3,30) When this is the case, the central limit theorem implies that simulation results are normally distributed and, consequently, the variance (or the standard deviation) provides a consistent measure of the uncertainty. It is worth noting that the simulation of each shower (*i.e.*, of the complete histories of a primary particle from the radiation source *and* the set of secondary particles that are released in the interactions) can be regarded as an individual sampling process of each quantity of interest Q (*e.g.*, the average energy deposited into the sensitive volume of a radiation detector, per simulated shower).

The considered quantity is evaluated by scoring the contribution q_j of each individual shower, and its square q_j^2 . After simulating a sufficiently large number N of showers, the average value $\bar{Q}$ and the variance var_Q of the sampled population

are (see Section 1.3 of the PENELOPE manual)

$$\bar{Q} = \frac{1}{N} \sum_{j=1}^N q_j, \quad (13)$$

and

$$\text{var}_Q = \frac{1}{N} \sum_{j=1}^N q_j^2 - \left[\frac{1}{N} \sum_{j=1}^N q_j \right]^2. \quad (14)$$

By default, the example main programs `penmain.f` and `pency1.f` deliver simulation results with 3σ uncertainty intervals, $\bar{Q} \pm 3\sigma_Q$, where $\sigma_Q = \sqrt{\text{var}_Q/N}$ is the standard deviation (or standard error) of $\bar{Q}$.

As a measure of the effectiveness of a Monte Carlo algorithm to compute the quantity Q , it is common to use the efficiency ϵ_Q , which is defined by

$$\epsilon_Q = \left(\frac{\bar{Q}}{\sigma_Q} \right)^2 \frac{1}{T}, \quad (15)$$

where T is the computing time (or any other measure of the calculation effort) needed to obtain the simulation result. In the limit of large N , σ_Q^2 and T are proportional to N^{-1} and N , respectively, and hence ϵ_Q is a constant (*i.e.*, it is independent of N). In practice, the efficiency ϵ_Q varies with N because of statistical fluctuations; the magnitude of these fluctuations decreases when N increases and eventually tends to zero.

In a formal sense, a Monte Carlo simulation is equivalent to the calculation of the integral defining the expectation value of the quantity of interest,

$$\bar{Q} = \int q(x) p(x) dx, \quad (16)$$

where $p(x)$ represents the probability density function of the variables x which determine the score q . Note that equality of the Monte Carlo estimate and the exact value of the integral is reached asymptotically, *i.e.*, when N , the number of samplings of q , tends to infinity. Similarly, the variance corresponds to the following integral

$$\text{var}_Q = \int [q(x) - \bar{Q}]^2 p(x) dx = \int q^2(x) p(x) dx - \bar{Q}^2. \quad (17)$$

The so-called variance-reduction methods are techniques that aim to optimize the *efficiency* of the simulation by re-arranging the integrand in Eq. (16). In radiation problems, this is frequently accomplished by assigning a weight w to each particle, which is suitably manipulated as the particle history progresses. We can write

$$\bar{Q} = \int wq(x) \frac{p(x)}{w} dx = \int q'(x) p'(x) dx, \quad (18)$$

with $q' = wq$ and $p' = p/w$. The variance then becomes

$$\text{var}'_Q = \int q'^2(x) p'(x) dx - \bar{Q}^2 = \int wq^2(x) p(x) dx - \bar{Q}^2. \quad (19)$$

Thus, with an adequate choice of weights, we could make $\text{var}'_Q < \text{var}_Q$ and, consequently, increase the efficiency of the

simulation of the quantity of interest; the weighted simulation would then attain the desired accuracy with less computer time. It should be noted that an efficiency gain in the evaluation of Q normally implies a loss of efficiency for other quantities that may be evaluated in the same run.

The source package `penvared.f` contains subroutines to automatically apply various elementary variance-reduction techniques which may modify the number and weights of the particles in a shower. It is assumed that primary particles are assigned a weight equal to unity; secondary particles inherit the weight w of the parent particle. The considered techniques are safe, that is, they keep the simulation results unbiased. However, care must be exercised to prevent the particle weights from becoming much smaller or much larger than the average weight. Particles with small weights will contribute very little (even though it takes time to simulate them). Particles with very large weights damage the statistics; they may produce variance “bombs”.

The considered techniques are:

- 1) *Particle splitting*. The current particle is split into $S (> 1)$ identical particles with weights equal to w/S .
- 2) *Russian roulette*. The particle is killed with probability $\mathcal{K} (< 1)$; if it survives, its weight is increased by a factor $(1 - \mathcal{K})^{-1}$. Splitting and Russian roulette are normally used together to favor the flux of radiation towards the region of interest and inhibit the radiation the leaves that region.
- 3) *Interaction forcing*. This technique consists of artificially inserting “forced” interactions of selected kinds randomly along the particle trajectory. This is accomplished by replacing the inverse mean free path λ_i^{-1} of an interaction by a larger value, $\mathcal{F} \lambda_i^{-1}$, where $\mathcal{F} (> 1)$ is the forcing factor specified by the user. To keep the simulation unbiased, interactions are allowed to affect the state of the transported particle only with probability $\mathcal{F}^{-1}$ and, at the same time, the deposited energy and the secondary particles generated in forced interactions are assigned a weight equal to $\mathcal{F}^{-1} w$. Note that we can force different types of interactions, with different forcing factors. Furthermore, forcing factors can be modified in the course of the simulation of a particle (whenever this does not interfere with the running sequence of class II simulation).
- 4) *Bremsstrahlung splitting*. In the simulation of each radiative event, a “normal” bremsstrahlung photon is emitted with energy W and the polar emission angle θ . The technique of bremsstrahlung splitting consists of splitting the normal photon into a number $\mathcal{B} (> 1)$ of photons, all them with the energy W and polar emission angle θ of the normal photon, weight equal to $\mathcal{B}^{-1} w$, and azimuthal emission angles sampled uniformly in the interval from zero to 2π . This method is computationally simple (a single DO loop) and very effective because each “split” photon is obtained by applying a rotation to the direction of the normal photon, which is much easier than performing the sampling of a normal photon.
- 5) *X-ray splitting*. Each x ray (emitted in the relaxation of ions with vacancies in inner shells caused by photoelectric absorption or electron/positron impact ionization) is split into a number $\mathcal{X} (> 1)$ of x rays with the same energy and random

directions distributed isotropically.

6) *Delta scattering of photons.* This technique, also known as the Woodcock method,^(31–33) is applicable only to photons. It takes advantage of the high penetration of photons to simplify the tracking of these particles through material systems with complex geometries. Photons are transported freely across the system using an augmented inverse mean free path, Λ^{-1} , which is larger than the actual IMFPs in all the materials crossed by a trajectory ray. The event at the end of each free flight may be either a real interaction or a delta interaction, which has no effect on the photon. Delta interactions occur with probability $1 - \lambda^{-1}/\Lambda^{-1}$, where λ^{-1} is the actual IMFP in the current material. This procedure avoids the need for computing intersections of particle rays with interfaces at the expense of having to determine which material is at the end of each free flight. Hence delta scattering will improve the efficiency only for those geometries where locating a particle (*i.e.*, finding the material at its current position) is faster than normal tracking.

Although the variance-reduction routines operate automatically and are robust, they should be invoked with care. The effects on the efficiency of the simulation are not always easy to predict. It is therefore advisable to perform tentative runs with different values of the variance-reduction parameters to check the efficiency gain (or loss!).

A situation where straight application of these variance-reduction techniques gives an enormous gain in efficiency is found in studies of x-ray emission from solids irradiated by electron beams. Simulations of this problem tend to be very lengthy because interactions that produce photons (inner-shell ionization and bremsstrahlung emission) occur with very small probabilities. Figure 6 displays energy spectra of photons emitted from a 50- μm thick copper foil irradiated by a 40-keV electron beam impinging normally on the foil. The detector is assumed to collect all photons that emerge from the irradiated surface with energies larger than 1 keV. The simulations were run on an Intel Core i7-3520M processor; the upper plot in Fig. 6 is the result from an analogue simulation run for 20 minutes. The lower plot was obtained from a 10-minute run using interaction forcing ($\mathcal{F} = 2000$ for inner-shell ionization by electron impact, and $\mathcal{F} = 200$ for bremsstrahlung emission), bremsstrahlung splitting ($\mathcal{B} = 2$) and x-ray splitting ($\mathcal{X} = 2$); the efficiency gain is evident.

VI. The main program penmain

Penmain is a generic main program that performs simulations of electron-photon transport in complex material structures. The program is devised to allow occasional users to employ PENELOPE without having to write their main program. The geometry of the material system is described by means of the PENGEO routines. Optionally, the program allows the use of various variance-reduction techniques (interaction forcing, Bremsstrahlung splitting, and x-ray splitting) in a systematic way. The operation of penmain is completely controlled from the input data file. Although it is impossible to cover all possible situations with a “closed” program, penmain is flexible enough to solve a broad class of practical problems.

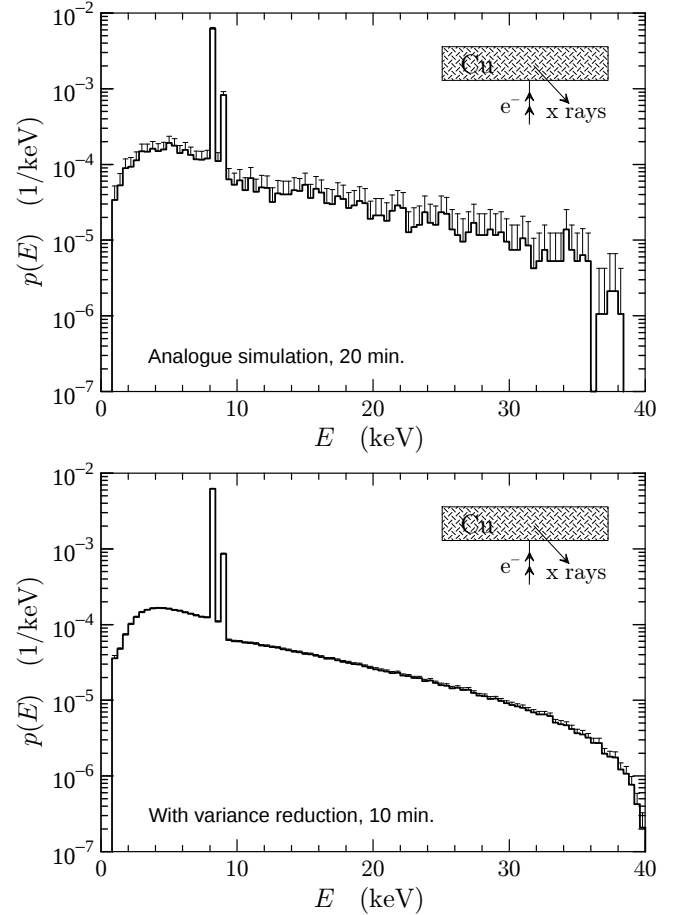


Figure 6: Results from simulations of x-ray emission from a Cu foil irradiated with a 40-keV electron beam. Error bars represent statistical uncertainties (only the positive side of the bars is displayed).

In the default mode, penmain assumes that primary particles of a given kind are emitted from a point or an extended source, either with fixed energy or with a specified (histogram-like) energy spectrum. The initial direction of the primary particles is sampled uniformly within a circle on the unit sphere (conical beam), or within a “rectangular” window on the unit sphere (rectangular beam). Alternatively, the program can read the initial state variables of “primary” particles from pre-calculated phase-space files. This option is useful for splitting the simulation of complex problems into several consecutive stages. The program also admits radiation sources defined by the user, although this requires some editing work.

Penmain provides global simulation results such as the energy and angular distributions of particles that emerge from the material system, the average energy deposited in each body, etc. To generate more specific information, the user can define impact detectors and energy-deposition detectors. Each detector consists of a set of active (non-void) bodies, which must have been defined as parts of the geometry. The output spectrum from an impact detector is the energy distribution of particles that entered any of the active bodies coming from a body that is not active (*i.e.*, that is not part of the detector). Optionally, for each impact detector, the program can generate a phase-space file where the state variables of particles at the detector entrance

are recorded, and it can also tally the average distribution of fluence with respect to energy of electrons, positrons and photons within the active volume of the detector. The output spectrum of an energy-deposition detector is the distribution of absorbed energy (per shower) in the active bodies. The program also offers the option of tallying a dose map on an orthogonal mesh defined by the user.

A report on the global simulation, including relevant input data as well as some partial results, is written in two output files named `penmain.dat` and `penmain-res.dat`. The calculated continuous distributions (histograms) are written in separate files, whose names have the extension “.dat”. These files are in a format suited for direct visualization with a plotting program. We use `GNUPLOT` (version 4.2), which is robust and flexible, available for different platforms, and is free.

Optionally, at the end of a run, the program can write the totality of simulation results in a dump file. The program can also write simulation results in a dump file at regular time intervals. A dump/resume option allows the user to stop the simulation at any time, and to resume it from the last dumping point in a completely consistent way. Although `penmain` is designed to be run on a single processor, we have recently included an auxiliary program that adds the values of the counters in a number of dump files generated from independent runs of `penmain`. This program allows the user to run the same problem on several processors (using different seeds of the random number generator to make the results statistically independent), and then combine the results to produce a single set of output files, with accumulated statistics. It works as an effective “poor man’s” parallelization device.

1. Example

As an example of tough calculation with `penmain` we consider a beam of 1-MeV photons impinging on a material structure consisting of a layer of 1 cm of water, followed by 2 cm of air, and 5 cm of water. Primary photons are assumed to come from a point source at 2 cm from the lower surface of the block with directions uniformly distributed within the solid angle of a cone with central axis normal to the surface and a semi-aperture of 5 degrees.

To ensure a proper description of electron transport near water-air interfaces, two 1-mm layers of water next to these interfaces were assigned a different water material. In this material and in the inner air we used conservative values of the simulation parameters $C_1 = C_2 = 0.01$ and $s_{\max} = 0.02$ cm, to have a faithful description of the transport of electrons near physical interfaces. For normal water we set $C_1 = C_2 = 0.05$ and $s_{\max} = 0.05$.

We applied interaction forcing for Compton scattering and photoelectric absorption with $\mathcal{F} = 85$ in air, and $\mathcal{F} = 5$ in water. Simulation results were obtained from a single run of `PENMAIN` in which about 3×10^8 primary photons were simulated; the simulation lasted for about 12 hours on an Intel Core i7-3520M processor. Figure 7 displays the simulated depth-dose distribution (*i.e.*, the energy deposited per unit mass thickness and per primary photon as a function of depth z). Figure 8 shows the absorbed dose distribution on a plane that contains

the central axis of the beam. Since the mean-free path of photons in air is quite large, of the order of 130 m, the use of heavy interaction forcing is required to obtain the dose in the air volume with acceptable accuracy.

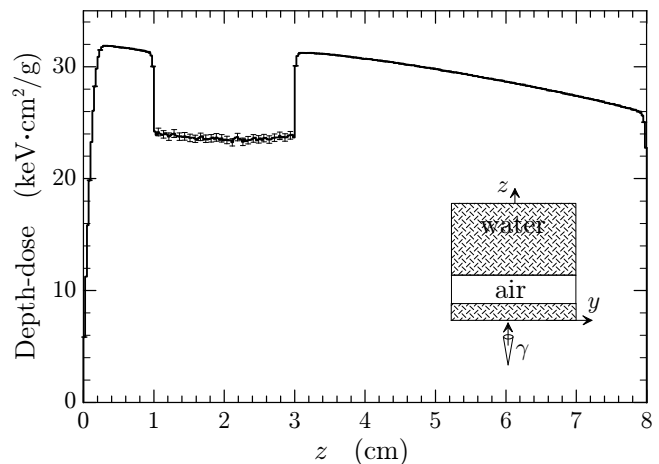


Figure 7: Depth-dose distribution in a composite water-air phantom irradiated with 1-MeV photons. The error bars represent statistical uncertainties.

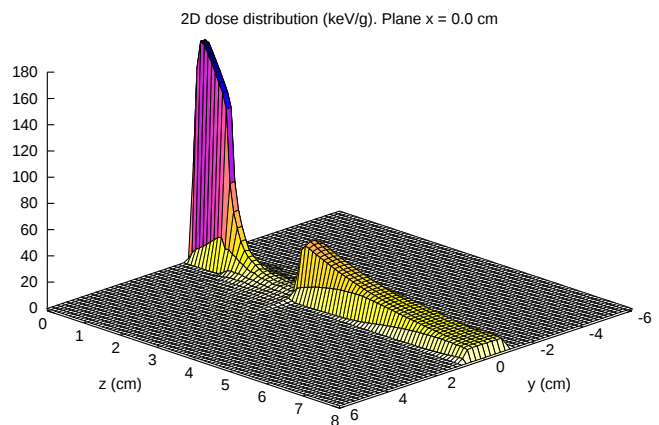


Figure 8: Surface plot of the absorbed dose distribution (per primary photon) at a plane perpendicular to the beam, generated with GNUPLOT.

Acknowledgments

I am deeply indebted to J.M. Fernández-Varea and J. Sempau, for their long-lasting collaboration in the development of the physics and the programming of `PENELOPE`. I am also grateful to A. Lallena, J. Almansa, and F. Salvat-Pujol for developing the `PENGEOM.JAR` program. Last, but not least, I would like to express my gratitude to the staff of the NEA Data Bank, especially to J.M. Galán, C. Lebetunelle, C. Rocher-Thomas and V. Kebedjis, for their help in distributing the code system and for kindly organizing training courses on `PENELOPE`.

Financial support from the Spanish Ministerio de Ciencia e Innovación and ERDF (project no. FPA2009-14091-C02-01) and from the Generalitat de Catalunya (grant SGR 2009-276) is gratefully acknowledged.

References

- 1) 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**, 31–46 (1995).
- 2) J. Sempau, E. Acosta, J. Baró, J. M. Fernández-Varea, and F. Salvat, "An algorithm for Monte Carlo simulation of coupled electron-photon transport," *Nucl. Instrum. Meth. B*, **13**, 377–390 (1997).
- 3) F. Salvat, J. M. Fernández-Varea, and J. Sempau, *PENELOPE-2011: A code System for Monte Carlo Simulation of Electron and Photon Transport*, OECD/NEA Data Bank, Issy-les-Moulineaux, France (2011). Available in PDF format from <http://www.nea.fr/lists/penelope.html>.
- 4) J. Sempau, J. M. Fernández-Varea, E. Acosta, and F. Salvat, "Experimental benchmarks of the Monte Carlo code PENELOPE," *Nucl. Instrum. Meth. B*, **207**, 107–123 (2003).
- 5) F. Salvat and J. M. Fernández-Varea, "Overview of physical interaction models for photon and electron transport used in Monte Carlo codes," *Metrologia*, **46**, S112–S138 (2009).
- 6) D. E. Cullen, J. H. Hubbell, and L. Kissel, "EPDL97 The Evaluated Data Library, '97 Version," UCRL-50400, Lawrence Livermore National Laboratory (1997).
- 7) D. Brusa, G. Stutz, J. A. Riveros, J. M. Fernández-Varea, and F. Salvat, "Fast sampling algorithm for the simulation of photon Compton scattering," *Nucl. Instrum. Meth. A*, **379**, 167–175 (1996).
- 8) D. A. Liberman, D. T. Cromer, and J. T. Waber, "Self-Consistent-field Dirac wave functions for atoms and ions. I. Comparison with previous calculations," *Phys. Rev.*, **137**, A27–A34 (1965).
- 9) D. Liberman, D. T. Cromer, and J. T. Waber, "Relativistic self-consistent field program for atoms and ions," *Comput. Phys. Commun.*, **2**, 107–113 (1971).
- 10) R. H. Pratt and H. K. Tseng, "Behavior of electron wave functions near the atomic nucleus and normalization screening theory in the atomic photoeffect," *Phys. Rev. A*, **5**, 1063–1072 (1972).
- 11) F. Sauter, "Über den atomaren Photoeffekt in der K-Schale nach der relativistischen Wellenmechanik Diracs," *Ann. Phys.*, **11**, 454–488 (1931).
- 12) M. J. Berger et al., "XCOM: Photon Cross Sections Database," Institute of Standards and Technology (2005, available from <http://physics.nist.gov/PhysRefData/Xcom/Text/XCOM.html>).
- 13) F. Salvat, A. Jablonski, and C. J. Powell, "ELSEPA—Dirac partial-wave calculation of elastic scattering of electrons and positrons by atoms, positive ions and molecules," *Comput. Phys. Commun.*, **165**, 157–190 (2005).
- 14) ICRU Report 77, *Elastic Scattering of Electrons and Positrons*, ICRU, Bethesda, MD (2007).
- 15) D. Liljequist, "A simple calculation of inelastic mean free path and stopping power for 50 eV – 50 keV electrons in solids," *J. Phys. D: Appl. Phys.*, **16**, 1567–1582 (1981).
- 16) R. M. Sternheimer, "The density effect for the ionization loss in various materials," *Phys. Rev.*, **88**, 851–859 (1952).
- 17) ICRU Report 37, *Stopping Powers for Electrons and Positrons*, ICRU, Bethesda, MD (1984).
- 18) D. Bote and F. Salvat, "Calculations of inner-shell ionization by electron impact with the distorted-wave and plane-wave Born approximations," *Phys. Rev. A*, **77**, 042701 (2008).
- 19) S. M. Seltzer and M. J. Berger, "Bremsstrahlung spectra from electron interactions with screened atomic nuclei and orbital electrons," *Nucl. Instrum. Meth. B*, **12**, 95–134 (1985).
- 20) S. M. Seltzer and M. J. Berger, "Bremsstrahlung energy spectra from electrons with kinetic energy 1 keV–10 GeV incident on screened nuclei and orbital electrons of neutral atoms with $Z = 1–100$," *At. Data Nucl. Data Tables*, **35**, 345–418 (1986).
- 21) E. Acosta, X. Llovet, and F. Salvat, "Monte Carlo simulation of bremsstrahlung emission by electrons," *Appl. Phys. Lett.*, **80**, 3228–3230 (2002).
- 22) L. Kissel, C. A. Quarles, and R. H. Pratt, "Shape functions for atomic-field bremsstrahlung from electrons of kinetic energy 1–500 keV on selected neutral atoms $1 \leq Z \leq 92$," *At. Data Nucl. Data Tables*, **28**, 381–460 (1983).
- 23) S. T. Perkins et al., "Tables and Graphs of Atomic Subshell and Relaxation Data Derived from the LLNL Evaluated Atomic Data Library (EADL), $Z = 1–100$," UCRL-ID-50400, Lawrence Livermore National Laboratory (1991).
- 24) R. D. Deslattes et al., "X-ray transition energies: new approach to a comprehensive evaluation," *Rev. Mod. Phys.*, **75**, 36–99 (2003).
- 25) J. A. Bearden, "X-ray wavelengths," *Rev. Mod. Phys.*, **39**, 78–124 (1967).
- 26) J. M. Fernández-Varea, R. Mayol, J. Baró, and F. Salvat, "On the theory and simulation of multiple elastic scattering of electrons," *Nucl. Instrum. Meth. B*, **73**, 447–473 (1993).
- 27) A. F. Bielajew and F. Salvat, "Improved electron transport mechanics in the PENELOPE Monte Carlo model," *Nucl. Instrum. Meth. B*, **173**, 332–343 (2001).
- 28) H. W. Lewis, "Multiple scattering in an infinite medium," *Phys. Rev.*, **78**, 526–529 (1950).
- 29) A. F. Bielajew, "HOWFAR and HOWNEAR: Geometry modeling for Monte Carlo particle transport," PIRS-0341, National Research Council of Canada (1995).
- 30) M. H. Kalos and P. A. Whitlock, *Monte Carlo Methods*, Wiley-VCH, Weinheim (2008).
- 31) E. Woodcock, T. Murphy, P. Hemmings, and S. Longworth, "Techniques used in the GEM code for Monte Carlo neutronics calculations in reactors and other systems of complex geometry," *Proc. Conf. on Applications of Computing Methods to Reactor Problems*, ANL-7050, pp 557–579, Argonne National Laboratories (1965).
- 32) W. A. Coleman, "Mathematical verification of a certain Monte Carlo sampling technique and applications of the technique to radiation transport problems," *Nucl. Sci. and Engineering*, **32**, 76–81 (1968).
- 33) J. Sempau, S. Wilderman, and A. F. Bielajew, "DPM, a fast, accurate Monte Carlo code optimized for photon and electron radiotherapy treatment planning dose calculations," *Phys. Med. Biol.*, **45**, 2263–2291 (2000).