

PENEPMA, a Monte Carlo code for the simulation of x-ray emission spectra using PENELOPE

Xavier Llovet¹ and Francesc Salvat²

¹ Serveis Científico-Tècnics, Universitat de Barcelona.
Lluís Solé i Sabarís, 1-3, 08028 Barcelona, Spain.

² Facultat de Física (ECM), Universitat de Barcelona.
Diagonal 647, 08028 Barcelona, Spain.

Abstract

The computer program PENEPMA performs Monte Carlo simulations of x-ray emission spectra from complex material structures irradiated by monoenergetic electron beams. The program uses the general-purpose Monte Carlo code system PENELOPE (version 2008) for electron and photon transport, which implements the most elaborate interaction models available for arbitrary materials. The structure and operation of PENEPMA are similar to those of the generic main program PENMAIN, which is part of the PENELOPE distribution package. PENEPMA is designed to allow occasional users to simulate EPMA experiments and x-ray generators without having to write a PENELOPE main program. The user defines the details of his or her experiment (electron-beam characteristics, geometrical structure and composition of the target, photon detectors) and the simulation control parameters by editing the input file. The program delivers energy spectra of photons that enter the various detectors, as well as characteristic line intensities, with fluorescence contributions given separately. Optionally, PENEPMA can generate a three-dimensional distribution of x-ray generation.

KEYWORDS: X-ray spectra. Electron probe microanalysis. Electron-photon showers. Monte Carlo simulation. PENELOPE

Date: 21 July, 2008

Contents

1	Introduction	1
2	Interaction models and simulation algorithm	2
3	The program PENEPMa	5
3.1	Simulation efficiency and interaction forcing	9
3.2	Detector response function	10
4	Structure of the PENEPMa input file	11
5	Selecting the simulation parameters	16
6	Installation	17
7	An example	18
	Appendix. Geometries of interest in EPMA	27
	References	35

1 Introduction

The generation of x rays from solid samples under electron bombardment is the basis of electron probe microanalysis (EPMA), a widely used technique for materials analysis. In principle, the elemental composition of the probed volume can be inferred from the measured characteristic x-ray intensities. In practice, the accuracy of quantitative analyses is limited severely by approximations used to describe the sample geometrical structure and the different interaction mechanisms that determine the transport of electrons within the sample, the generation of x rays, and the transport of these from the generation site to the detector. A similar problem is encountered in the design and optimisation of medical and industrial x-ray generators.

The most reliable description of radiation transport in matter is provided by Monte Carlo simulation methods. The reliability of these methods stems from their ability to incorporate realistic physical interaction models and from the ease with which they can handle complex geometries. In a Monte Carlo simulation, we can not only describe the penetration and slowing down of the primary electrons, but also the generation and transport of secondary radiations (electrons and photons). Several groups have developed Monte Carlo codes for EPMA applications, but some of them involve approximations (particularly in the description of fluorescence corrections) which limit their accuracy.

The present simulation code is based on PENELOPE¹ (Salvat *et al.*, 2006), a general-purpose Monte Carlo code system for the simulation of coupled electron-photon transport in arbitrary materials. PENELOPE covers the energy range from 1 GeV down to, nominally, 50 eV. The physical interaction models implemented in the code are based on the most reliable information available at present, limited only by the required generality of the code. These models combine results from first-principles calculations, semi-empirical models and evaluated data bases. It should be borne in mind that although PENELOPE can run particles down to 50 eV, the interaction cross sections for energies below ~ 1 keV may be affected by sizeable uncertainties; the results for these energies should be considered as semi-quantitative. PENELOPE incorporates a flexible geometry package called PENGEOM that permits automatic tracking of particles in complex geometries consisting of homogeneous bodies limited by quadric surfaces. The PENELOPE code system is distributed by the OECD/NEA Data Bank (<http://www.nea.fr>). The distribution package includes a report (Salvat *et al.*, 2006) that provides detailed information on the physical models and random sampling algorithms adopted in PENELOPE, on the PENGEOM geometry package, and on the structure and operation of the simulation routines. In the following, this report will be referred to as “the PENELOPE manual”.

PENELOPE is coded as a set of FORTRAN subroutines, which perform the random sampling of interactions and the tracking of particles². In principle, the user should provide a main steering program to follow the particle histories through the material

¹PENELOPE is an acronym for “Penetration and ENergy LOss of Positrons and Electrons”.

²For brevity, we use the term “particle” to refer to either electrons, positrons or photons.

structure and to keep score of quantities of interest. To facilitate the application of PENELOPE to EPMA, we have written a dedicated main program called PENEPMMA, which performs simulation of x-ray spectra and calculates quantities of interest. The operation of PENEPMMA is completely controlled from input data files and, therefore, occasional users are spared from having to do any programming work.

The present report is organised as follows. In Section 2, we briefly present the interaction models and simulation algorithms implemented in PENELOPE (version 2006). Section 3 describes general features of the PENEPMMA program, with instructions on efficiency optimisation and the convolution of Monte Carlo spectra with the detector response function. The operation of PENEPMMA is controlled through the input data file, the structure of this file and the action of the different options are described in Section 4. In Section 5 we give practical indications to set the electron simulation parameters. The installation of the code system is described in Section 6. A practical calculation is presented in Section 7; the simulation results illustrate the capabilities of PENEPMMA as an auxiliary tool for quantitative EPMA. Finally, the Appendix contains examples of geometry definition files corresponding to geometrical structures that may be found in EPMA samples.

2 Interaction models and simulation algorithm

PENELOPE combines numerical and analytical total and differential cross sections (DCS) to describe the different interaction mechanisms. These cross sections are the result from approximate physical models and, therefore, are affected by systematic uncertainties. In particular, the DCSs in the database correspond to interactions with free, isolated atoms. DCSs for compounds and mixtures are derived by using Bragg's additivity rule, *i.e.*, the molecular DCS is approximated by the sum of atomic DCSs of the atoms in a molecule. Aggregation effects are therefore disregarded. These effects are known to be appreciable, *e.g.*, for elastic scattering of electrons with energies less than about 1 keV. Aggregation also affects the photoelectric cross section, causing extended x-ray absorption fine structure effects. On the other hand, the adopted x-ray energies and transition probabilities of excited atoms with inner-shell vacancies pertain to free atoms with a single vacancy. Nonetheless, the structure of the code is flexible enough to enable the use of alternative, more elaborate physical models.

The interaction mechanisms considered in PENELOPE-2006, and the corresponding DCSs, are the following:

- Elastic scattering of electrons and positrons: numerical DCSs obtained from Dirac partial-wave analysis for the electrostatic potential derived 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 (Salvat *et al.*, 2005).
- Inelastic collisions of electrons and positrons: Born DCS obtained from the Sternheimer-Liljequist generalised oscillator strength model (Liljequist, 1983; Sternheimer, 1952), including the density-effect correction. Optionally, the DCS can be renormalised to

reproduce the collision stopping power read from the input file.

- Electron impact ionisation: numerical total cross sections for ionisation of atomic electron shells obtained from a combination of the distorted-wave Born approximation and the plane-wave Born approximation. The database was computed using the theory and numerical methods described by Bote and Salvat (2008). These ionization cross sections are much more accurate than those used in older versions of PENEPMA, especially in the energy range of interest in EPMA.
- Bremsstrahlung emission by electrons and positrons: the energy of the emitted photons is sampled from numerical energy-loss spectra derived from the scaled cross-section tables of Seltzer and Berger (1985), optionally renormalised 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—(Acosta *et al.*, 2002) with parameters determined by fitting the benchmark partial-wave shape functions of Kissel, Quarles and Pratt (1983).
- Positron annihilation: Heitler DCS for two-photon annihilation in flight.
- Coherent (Rayleigh) scattering of photons: Born DCS with an analytical atomic form factor.
- Incoherent (Compton) scattering of photons: DCS calculated using the relativistic impulse approximation with analytical one-electron Compton profiles (Brusa *et al.*, 1996).
- Photoelectric absorption of photons: total atomic cross sections and partial cross sections for the K-shell and L- and M- subshells from the LLNL Evaluated Photon Data Library (Cullen *et al.*, 1989). The initial direction of photoelectrons is sampled from Sauter’s K-shell hydrogenic DCS.
- Electron-positron pair production: total cross sections obtained from the XCOM program of Berger and Hubbell (1987). The initial kinetic energies of the produced particles are sampled from the Bethe–Heitler DCS, with exponential screening and Coulomb correction, empirically modified to improve its reliability for energies near the pair-production threshold.

Particle tracks are simulated from the initial energy down to the absorption energies selected by the user (see Section 5), at which particles are considered to be effectively absorbed in the medium. Secondary particles emitted with initial energy larger than the corresponding absorption energy are simulated after completion of each primary track. Secondary particles are produced in direct interactions (inelastic collisions, bremsstrahlung emission, positron annihilation, Compton scattering, photoelectric absorption and pair production) and as radiation (characteristic x rays and Auger electrons) following inner-shell ionisation. PENELOPE simulates the emission of characteristic x rays and Auger electrons that result from vacancies produced in K-shells and L- and M-subshells by photoelectric absorption and Compton scattering of photons and by electron or positron impact. The relaxation of these vacancies is followed until all vacancies have migrated to N and outer shells. The adopted transition probabilities have been extracted from the LLNL Evaluated Atomic Data Library (Perkins *et al.*,

1991).

The simulation of photon tracks follows the usual detailed procedure, *i.e.*, all the interaction events in a photon history are simulated in chronological order. Electrons can also be simulated in detail, or alternatively, by using a mixed (class II) simulation algorithm (Salvat *et al.*, 2006). Mixed simulation algorithms are useful when the electrons undergo a very large number of interactions and detailed simulation becomes impractical (*e.g.*, for primary electron energies above ~ 100 keV). In a mixed algorithm, the user defines cut-off values of the polar scattering angle (θ_c) and the energy loss (W_c) in single interactions. Hard interactions, with scattering angle $\theta \geq \theta_c$ or energy loss $W \geq W_c$, are simulated individually according to the adopted single-scattering DCSs. These interactions are responsible for large lateral displacements or large energy-loss fluctuations, which detailed simulation describes faithfully. The rest of interactions, with $\theta < \theta_c$ and $W < W_c$, are classified as soft interactions; they have a mild effect on the electron trajectories and are simulated by using simple, but still fairly accurate, multiple scattering approximations. With a judicious selection of the cut-off parameters θ_c and W_c , mixed simulation algorithms yield reliable results in moderate computation times, even for electrons with very high energies.

The straight simulation of EPMA spectra may be very inefficient, even when using mixed simulation algorithms for electron transport. The reason is that the two processes leading to the emission of x-rays (inner-shell ionisation and bremsstrahlung emission) occur with exceedingly small probabilities. That is, on average, we need to generate many electron histories to obtain one emerging photon. Moreover, as the solid angle subtended by the detector is usually very small, only a tiny fraction of the emerging photons hits the detector and contributes to the spectrum. The method adopted in PENEPA to cope with this difficulty is to use the variance-reduction technique of interaction forcing (also known as the method of weights), which consists of artificially increasing the probability of inner-shell ionisation and bremsstrahlung emission and, at the same time, assigning appropriate statistical weights to the generated secondary particles in such a way that the simulation results remain unbiased. Interaction forcing is also advantageous in studies of fluorescence corrections; in this case the interactions that should be forced are photoelectric absorption and Compton scattering of photons. PENELOPE includes a flexible implementation of interaction forcing, which allows x-ray spectra to be computed in reasonable times (of the order of 1 hour, on modest personal computers).

Notice that PENELOPE tracks primary electrons *and* all generated particles (electrons and photons), that is, it generates the whole electron-photon shower produced by each primary electron. Therefore, fluorescence effects are consistently described.

The practical usefulness of Monte Carlo simulation lies in its ability to handle complex geometrical structures. PENELOPE contains the subroutine package PENGEO which automatically tracks particles through arbitrary material systems consisting of homogeneous bodies limited by quadric surfaces. Limiting surfaces can be specified either through their implicit equation, $f(\mathbf{r}) = 0$, or by means of their reduced form and a few simple geometrical transformations. A quadric surface divides the space into two exclusive regions; the region where $f(\mathbf{r}) < 0$ ($f(\mathbf{r}) > 0$) is assigned the side pointer -1

(+1). A body is defined by giving its limiting quadric surfaces and corresponding side pointers. Previously defined bodies can be used to define new bodies. To speed up the geometry operations, the bodies of the material system may be grouped into modules (connected volumes, limited by quadric surfaces that contain one or several bodies), which in turn can form part of larger modules. This modular structure permits a substantial reduction of the work of the geometry routines, and allows very complicated structures to be considered (the current version of PENGEO can handle systems with up to 9,999 surfaces and 5,000 bodies and modules). Repetitive structures can also be easily defined by using the “clone” operation.

3 The program PENEPMA

As mentioned above, PENEPMA is a dedicated program that performs simulations of x-ray emission in arbitrary quadric geometries. The geometry of the material system is described by means of the subroutine package PENGEO. The operation of PENEPMA is completely controlled from the input data file.

In the simulation, all position and direction vectors are referred to a fixed orthogonal frame, the laboratory frame, which is selected by the user. Lengths and energies are given in cm and eV, respectively. Occasionally, directions (unit vectors, $\hat{\mathbf{d}}$) are specified by giving the polar and azimuthal angles, θ and ϕ , respectively. Notice that θ , the angle between the direction vector $\hat{\mathbf{d}}$ and the z axis, may take values between 0 and 180° (which correspond to the directions of the positive and negative z axis, respectively). The azimuthal angle ϕ is the angle between the projection of $\hat{\mathbf{d}}$ on the xy -plane and the x axis and can take values between 0 and 360° . PENEPMA assumes that primary electrons are emitted from a point source, at a certain position, $\mathbf{r}_0 = (x_0, y_0, z_0)$, with fixed energy. The initial direction of primary electrons is sampled isotropically within a cone of given semi-aperture α and central axis in the direction (θ_0, ϕ_0) (see Fig. 1). Notice that $\alpha = 0$ defines a parallel pencil beam, and $\alpha = 180^\circ$ corresponds to an isotropic source.

The geometry of the target is defined by means of an input file, which must be written following the PENGEO file structure and format rules (see Chapter 5 of the PENELOPE manual). Before attempting simulation, the geometry file should be viewed and debugged by running the viewing programs GVIEW2D and GVIEW3D. These programs issue a geometry report that contains indications on possible inconsistencies in the geometry definition file, and on the adequacy of its modular structure.

Material data files for all materials present in the target should be generated by running the pre-processing program MATERIAL, and concatenated in a single file, in the order assumed in the geometry definition file. Notice that if materials are not properly ordered, simulation results will not correspond to the case you have in mind.

The code assumes the experimental arrangement sketched in Fig. 2. In principle, there are no constraints on the shape and position of the sample. However, for the sake of simplicity, we shall assume that the upper surface of the sample is flat and coincides

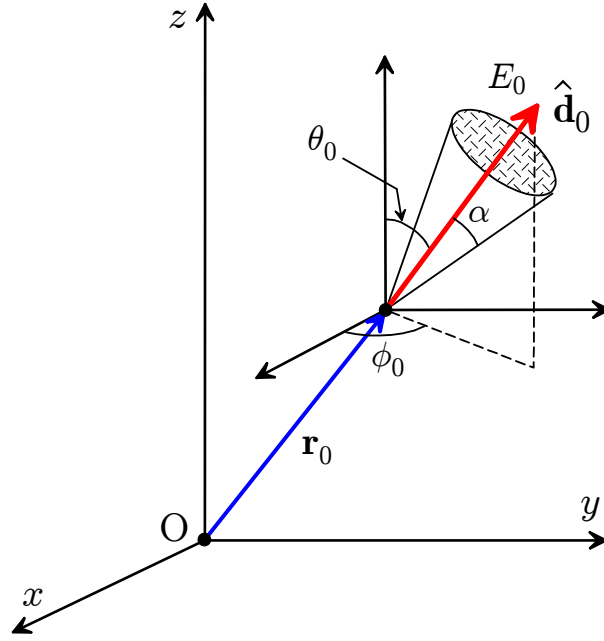


Figure 1: Definition of the primary electron beam. Electrons start off from the position $\mathbf{r}_0$. The beam axis is specified by giving its polar and azimuthal angles, θ_0 and ϕ_0 , and the semi-aperture α , all in degrees.

with the $z = 0$ plane. The primary electron beam is defined as indicated above. It is the responsibility of the user to make sure that primary electrons do impinge on the sample (otherwise, you will find that the simulation goes really fast: electrons simply leave the geometry enclosure without interacting). To define a conventional EPMA experiment (see Fig. 2), you can place the source on a point on the positive z axis, $\mathbf{r}_0 = (0, 0, z_0)$, with $z_0 > 0$, and set $\theta_0 = 0^\circ$, $\phi_0 = 0$ (electrons fly downwards along the z axis) and $\alpha = 0$ (parallel beam). To define a divergent electron beam, set α equal to the semi-aperture angle, and assign a value consistent with the desired spot size to z_0 .

The user can define a number of ideal photon detectors (up to 25). Each detector covers a solid angle corresponding to a “square”, $(\theta_1, \theta_2) \times (\phi_1, \phi_2)$, on the surface of a sphere of large radius and centred at the origin of the reference frame (see Fig. 2). This particular shape of the active surface of a detector is useful for normalising the simulated spectrum. The program simultaneously generates the energy spectra of x rays that reach each detector (the simulation time is practically independent of the number of detectors used). It should be noted that in actual EPMA measurements, x-ray detectors cover a very small solid angle about a direction (θ_d, ϕ_d) with a take-off angle (relative to the sample surface) of about 30-40 deg. Simulations with this configuration would be very inefficient, because of the small collection solid angle. In cases where the flux of emerging photons has axial symmetry about the z -axis (*e.g.*, for plane-surface samples that are homogeneous within the interaction volume, and at normal incidence), the efficiency of the simulation can be increased by using an annular detector (with $\phi_1 = 0$ and $\phi_2 = 360^\circ$) that covers a finite interval of polar emission angles θ .

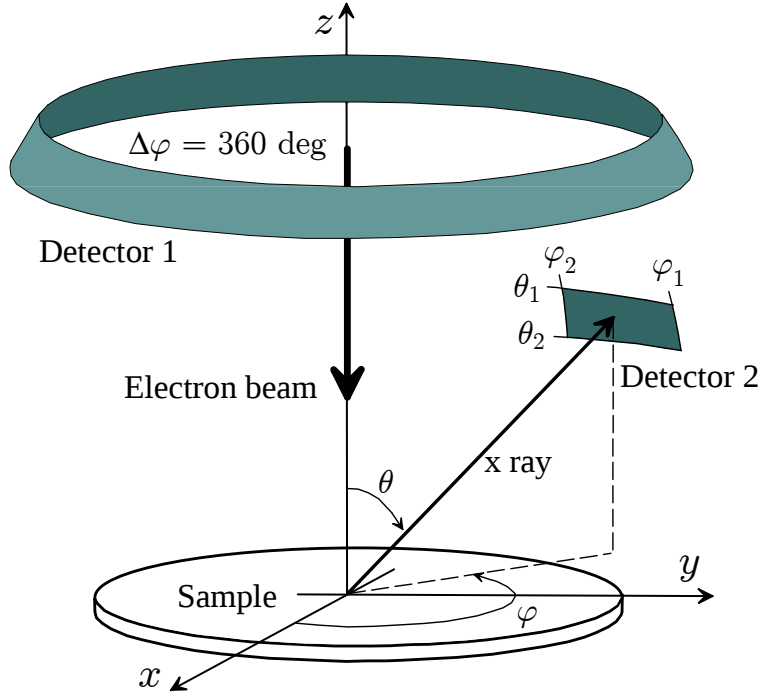


Figure 2: Geometry of the experimental set-up assumed in PENEPMA. Each detector is defined by specifying the limiting values of the polar and azimuthal angles, $(\theta_1, \theta_2) \times (\phi_1, \phi_2)$. The annular detector 1 is useful to get fast simulation results in cases where the x-ray flux is axially symmetric.

The program PENEPMA generates detailed information about many quantities of physical interest, and a variety of continuous distributions (depending on the options selected in the input file). The results are issued in a number of output files, whose names start with “pe-” (for PenEpma) and have the extension “.dat”. Each calculated continuous distribution is written in a separate file (as a histogram), with a heading describing its content and in a format ready for visualisation with a plotting program. The following is the list of possible output files generated by PENEPMA:

- The file **penepma.dat** contains a global report on the simulation, which includes relevant input data as well as some calculated generic quantities (such as fractions of primary electrons that are transmitted, backscattered and absorbed, secondary particle generation probabilities, the average energy deposited within each body of the sample, and the average energy of photons that reach each detector).
- The files **pe-angle-el.dat** and **pe-angle-ph.dat** contain angular distributions $p(\theta, \phi)$ of emerging electrons and photons, respectively. Angular distributions, $p(\theta)$, depending only on the polar angle (*i.e.*, integrated over ϕ), are written in the files **pe-anel.dat** and **pe-anph.dat**.
- The files **pe-energy-el-trans.dat** and **pe-energy-el-back.dat** give the energy distributions of electrons that are “transmitted” and “backscattered”. Here, these terms are used to designate particles that leave the sample moving upwards (with $\theta < 90^\circ$) and downwards (with $\theta > 90^\circ$), respectively. This convention agrees with the usual

meaning of these terms only when the source is placed below the sample.

- The files `pe-energy-ph-trans.dat` and `pe-energy-ph-back.dat` contain the energy distributions of transmitted and backscattered photons (*i.e.*, photons that leave the sample moving upwards and downwards, respectively).
- The global photon energy spectrum (including characteristic peaks and bremsstrahlung background) from detector number `N` is written in a file named `pe-spect-N.dat`. The output spectrum is given in absolute units, *i.e.*, as the probability density for detecting a photon per unit photon energy, per unit solid angle and per incident electron.
- Since PENEPMMA keeps track of the interaction mechanism that originates each secondary particle; it is able to generate the energy spectrum of characteristic x rays that reach a detector, *i.e.*, with the bremsstrahlung background removed. The program also tallies the fluorescence contribution to each characteristic peak (*i.e.*, photons that are not generated from direct interactions of primary electrons). The file `pe-charact-N.dat` contains the characteristic spectrum from detector number `N` and the fluorescence contribution, both in absolute units.
- Optionally, for each photon detector, `penepma` can generate a *phase-space file* where the state variables of photons that reach the detector are recorded. The output phase-space file has the name `pe-psf-N.dat`, where `N` is the detector number. Phase-space files are useful, *e.g.*, to check the effectiveness of interaction forcing and to identify the mechanisms that originated the detected photons.
- The program also offers the option of scoring the spatial distribution of x-ray emission within the volume of a parallelepiped (the scoring box) with a Cartesian mesh, and for a given interval of x-ray energies. The file `pe-3d-xrays.dat` contains the 3D distribution of x-ray emission (probability of x-ray emission per unit volume and per incident electron). The files `pe-x-xrays.dat`, `pe-y-xrays.dat` and `pe-z-xrays.dat` give the distribution of x-ray emission along straight lines parallel to the coordinate axes and that intersect at the centre of the scoring box. The file `pe-depth-xrays.dat` contains the conventional depth distribution of x-ray generation (the $\Phi(\rho z)$ function), limited to the volume of the scoring box.

Both PENELOPE and PENGEOM generate additional files, which are useful to verify the correctness of input data. Thus, PENELOPE writes a file named `pe-material.dat` that contains a listing of material characteristics, relaxation data and interaction data for the materials present in the target; this file is a readable version of the input material definition file. The files `pe-geometry.rep` and `pengeom-tree.rep` are generated by PENGEOM. The first is a duplicate of the input geometry definition file (error messages from PENGEOM are written to this file); the file `pengeom-tree.rep` gives the structure of the modular tree, with indications regarding possible optimisation of the geometry definition. If our description of the problem under consideration suffers from inconsistencies (*e.g.*, incorrect ordering of materials in the PENELOPE material data file), the information needed for identifying and resolving them should be sought in these three output files. When the problem is properly defined, these files can be ignored.

PENEPMMA is designed to be run on single-processor computers. The dump/resume option allows us to stop the simulation at any time, and to resume it from the last

dumping point in a completely consistent way. Optionally, at the end of a run, the program can write the contents of all counters to a dump file; using this option, a simulation with poor statistics can be resumed from exactly the same point where it was stopped (without losing the work done in the previous run). In addition, the program can generate the output and dump files at specified intervals of time; this option allows the user to inspect the results as they are being computed, and to stop the simulation when the required statistical accuracy has been reached. In this last case, make sure that execution is not stopped while the output files are being written to the hard disc; otherwise, the results will be lost.

3.1 Simulation efficiency and interaction forcing

Radiation transport codes usually calculate multiple quantities Q in a single run. Each quantity is evaluated by scoring the contribution q_i of each individual shower, and its square q_i^2 . After simulating a sufficiently large number N of showers, the average value and the standard deviation of Q of the sampled population are evaluated as (see Section 1.3 of the PENELOPE manual)

$$\bar{Q} = \frac{1}{N} \sum_{i=1}^N q_i, \quad \sigma_Q = \sqrt{\frac{1}{N} \sum_{i=1}^N q_i^2 - \left[\frac{1}{N} \sum_{i=1}^N q_i \right]^2}. \quad (1)$$

Assuming that the number N of simulated showers is large enough for the central limit theorem to apply, the “error” interval $\bar{Q} \pm n\sigma_Q$ contains the exact average value $\langle Q \rangle$ with a probability of 68.3% if $n = 1$, 95.4% if $n = 2$ and 99.7% if $n = 3$ (3σ rule). PENEPMA computes and delivers the statistical uncertainties (3σ) of all evaluated quantities and distributions. This implies that the probability of “missing the real value” is 0.003.

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

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

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 is a constant (*i.e.*, it is independent of N). In practice, the efficiency ϵ varies with N because of statistical fluctuations; the magnitude of these fluctuations decreases when N increases and eventually tends to zero.

The analogue simulation of x-ray spectra is very inefficient. Methods to improve the efficiency of Monte Carlo simulation are usually referred to as variance-reduction methods (see Section 1.6 of the PENELOPE manual). In PENEPMA we use the variance-reduction technique of interaction forcing. In this approach, the cross section of a mechanism of interest is multiplied by a “forcing factor” **FORCER** (larger than unity), and the generated secondary particles are given a statistical weight equal to that of the parent particle divided by **FORCER**. That is, we generate **FORCER** times more secondary

particles, but their total statistical weight is the same as in analogue simulation (without interaction forcing).

As indicated above, the interaction mechanisms that need to be forced to optimise the simulation of x-ray spectra are electron bremsstrahlung emission and inner-shell ionisation by electrons, photon Compton scattering and photoelectric absorption. The effect of interaction forcing on the global efficiency is not easy to predict, because each spectrum consists of multiple counters (one for each energy channel), which may not be uniformly affected. The program PENEPMMA gives the simulation efficiency for the average energy of photons that reach each detector. This single quantity provides a measure of the energy flux at a detector and can be used to judge the effectiveness of interaction forcing. Please try and do tentative runs with different **FORCER** values and check the efficiency gain (or loss!).

3.2 Detector response function

As the simulated spectrum $D_{\text{MC}}(E)$ corresponds to an ideal detector, characteristic lines are narrow, *i.e.*, each line is fully contained in a single channel of the output histogram. To obtain a realistic spectrum, which can be compared, *e.g.*, to an experimental energy-dispersive spectrum, we must introduce the effect of the efficiency and the finite energy resolution of the detector. The efficiency $\epsilon_d(E)$ is the probability that a photon of energy E incident on the detector window is effectively detected, that is, counted. The energy resolution function $R(E, E_{\text{ch}})$ gives the probability (normalised to unity) that a photon of energy E that is effectively detected produces a count in the energy channel corresponding to energy E_{ch} . The efficiency and the energy resolution of a particular detector should be determined experimentally. The product $\epsilon_d(E) R(E, E_{\text{ch}})$ is known as the response function of the detector. Thus, in the ideal case where N_E photons of energy E enter the detector, the “measured” spectrum is

$$D(E_{\text{ch}}) = N_E \epsilon_d(E) R(E, E_{\text{ch}}). \quad (3)$$

Given a simulated spectrum $D_{\text{MC}}(E)$, we can calculate the corresponding “measured” spectrum $D_{\text{MC}}(E)$ as

$$D(E_{\text{ch}}) = \int_0^\infty [D_{\text{MC}}(E) \epsilon_d(E)] R(E, E_{\text{ch}}) dE. \quad (4)$$

The PENEPMMA distribution package contains the Fortran source file `convolg.f` of a program that performs this calculation. In this program, it is assumed that the efficiency of the detector is equal to unity and the energy resolution function is a Gaussian characterised by its full width at half maximum (FWHM),

$$R(E, E_{\text{ch}}) = \frac{1}{\sqrt{2\pi} \sigma(E)} \exp \left[-\frac{1}{2} \left(\frac{E - E_{\text{ch}}}{\sigma(E)} \right)^2 \right], \quad \sigma = 0.42466 \times \text{FWHM}(E). \quad (5)$$

The last function in the source file `convolg.f`, `FWHM(E)`, gives the FWHM of the resolution function as a function of E for a particular SiLi detector. To define the FWHM

of a different detector, the user has to edit the source file and modify this function. We can account for the actual efficiency of the detector by simply multiplying the simulated spectrum $D_{MC}(E)$ by the efficiency $\epsilon_d(E)$.

4 Structure of the PENEPMA input file

PENEPMA operates similarly to the example main program PENMAIN, which is included in the PENELOPE distribution package, and is described in the manual. As indicated above, the program reads data from an input file and outputs the results in a number of files with fixed names³. The program also uses a geometry file and a material file that define the geometrical structure and composition of the sample (see the PENELOPE manual). These files are declared in the PENEPMA input file; they must be placed in a directory that is accessible from the working directory where the executable file is run.

Each line in the PENEPMA input data file consists of a 6-character keyword (columns 1–6) and a blank (column 7), followed either by numerical data (in free format) or by a character string, which start at the 8-th column. Keywords are explicitly used/verified by the program (which is case sensitive!). Notice also that the order of the data lines is important. The keyword “-----” (6 blanks, which we have denoted by “-”) indicates comment lines and can be placed anywhere in the file. The program ignores any text following the first blank after the last numerical datum, or after the character string, in each line (thus, in the input file layout given below, the comments in square brackets are ignored). Lines with certain keywords (*e.g.*, “PDANGL” and “PDENER”) can appear an arbitrary number of times, limited only by the allocated amount of memory. The program assigns default values to many input variables; lines that declare default values may be removed from the input file. The execution of the program is aborted when an incorrect input datum is found. The conflicting quantity usually appears in the last line of the output file. If the trouble is with arrays having dimensions smaller than required, the program indicates how the problem can be solved (this may require editing the source file, be careful).

The structure of the `penepma` input file is the following (the 72-column rulers are just for visual aid, they do not form part of the input file).

```

.....1.....2.....3.....4.....5.....6.....7..
TITLE  Title of the job, up to 120 characters.
      . (the dot prevents editors from removing trailing blanks)
      >>>>>> Electron beam definition.
SENERG SEO                                [Energy of the electron beam]
SPOSIT SX0,SY0,SZ0                        [Coordinates of the electron source]
SDIREC STHETA,SPHI                       [Direction angles of the beam axis, in deg]
SAPERT SALPHA                             [Beam aperture, in deg]
```

³Warning: PENEPMA overwrites older output files that are left in the working directory. You should save all result files on a separate directory before rerunning a program.

```

.
>>>>>>> Material data and simulation parameters.
          Up to 10 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 of the sample.
GEOMFN geo-filename.ext          [Geometry definition file, 20 chars]
DSMAX IBODY,DSMAX(IBODY)         [IB, maximum step length (cm) in body IB]
.
>>>>>>> Interaction forcing.
IFORCE KB,KPAR,ICOL,FORCER,WLOW,WHIG [KB,KPAR,ICOL,FORCER,WLOW,WHIG] *
.
>>>>>>> Emerging particles. Energy and angular distributions.
NBE EMIN,EMAX,NBE                [E-interval and no. of energy bins]
NBTH NBTH                        [No. of bins for the polar angle THETA]
NBPH NBPH                        [No. of bins for the azimuthal angle PHI]
.
>>>>>>> Photon detectors (up to 25 different detectors).
IPSF=0, the psf is not created.
IPSF=1, a psf is created.
IPSF>1, a psf is created, but contains only state variables
        of detected photons that have ILB(4)=IPSF (used for
        studying angular distributions of x rays).
PDANGL THETA1,THETA2,PHI1,PHI2,IPSF [Angular window, in deg, IPSF] &*
PDENER EDEL,EDEU,NCHE              [Energy window, no. of channels] &*
XRORIG xorigID.dat                 [Map of emission sites of detected x-rays] &*
.
>>>>>>> Spatial distribution of events.
GRIDX XL,XU                       [X coordinates of the box vertices]
GRIDY YL,YU                       [Y coordinates of the box vertices]
GRIDZ ZL,ZU                       [Z coordinates of the box vertices]
GRIDBN NX,NY,NZ                   [Numbers of bins]
XRAYE EMIN,EMAX                   [Energy interval where x-rays are tallied]
.
>>>>>>> Job properties.
RESUME dumpfile1.dat              [Resume from this dump file, 20 chars]
DUMPTO dumpfile2.dat              [Generate this dump file, 20 chars]
DUMPP DUMPP                       [Dumping period, in sec]
.
NSIMSH DSHN                       [Desired number of simulated showers]
RSEED ISEED1,ISEED2               [Seeds of the random-number generator]
TIME TIMEA                        [Allotted simulation time, in sec]
.....1.....2.....3.....4.....5.....6.....7..

```

The following listing describes the function of each of the keywords, the accompanying

data and their default values.

TITLE ... Title of the job (up to 120 characters).

-- Default: none (the input file must start with this line)

The **TITLE** string is used to mark dump files. To prevent improper use of incorrect resuming files, change the title each time you modify basic parameters of your problem. The code will then be able to identify the inconsistency and it will stop after writing an error message.

• Electron beam definition

SENERG ... Initial energy $SE0$ of the electron beam.

-- Default: $SE0=1.0E6$

SPOSIT ... Coordinates (x_0, y_0, z_0) of the electron source.

-- Defaults: $SX0=SY0=0.0$, $SZ0=-1.0E15$

SDIREC ... Polar angle θ_0 and azimuthal angle ϕ_0 of the electron beam axis direction, in deg.

-- Defaults: $STHETA=0.0$, $SPHI=0.0$

SAPERT ... Angular aperture α of the electron beam, in deg.

-- Default: $SALPHA=0.0$

Notice that the default beam is a pencil beam that moves downwards along the z -axis.

• Material data and simulation parameters

Each material is defined by introducing the following *two* lines;

MFNAME ... Name of a PENELOPE input material data file (up to 20 characters). This file must be generated in advance by running the program **MATERIAL**.

-- Default: $NMAT=1$

MSIMPA ... Set of simulation parameters for the current material; absorption energies, $EABS(1:3,M)$, elastic-scattering parameters, $C1(M)$ and $C2(M)$, and cutoff energy losses for inelastic collisions and bremsstrahlung emission, $WCC(M)$ and $WCR(M)$.

-- Defaults: $EABS(1,M)=EABS(3,M)=0.01*SE0$, $EABS(2,M)=0.001*SE0$
 $C1(M)=C2(M)=0.05$, $WCC(M)=EABS(1,M)$, $WCR(M)=EABS(2,M)$

Note that we must specify a separate material file name and a set of simulation parameters for each material. The label (material number) assigned by PENELOPE to each material is determined by the ordering of the material definitions in the input file. That is, the first, second, ... materials are assigned the labels 1, 2, ... These labels are also used in the geometry definition file.

• Geometry definition

GEOMFN ... PENGEOm's geometry definition file name (a string of up to 20 characters).

-- Default: none

The geometry definition file can be debugged and visualised with the viewers **GVIEW2D** and **GVIEW3D**. Notice that bodies are referenced by the internal label

assigned by PENGEO; this label can be identified by running the viewer GVIEW2D or by inspecting the output geometry file `pe-geometry.rep`, which is written by PENGEO.

DSMAX ... Maximum step length DSMAX(KB) (in cm) of electrons and positrons in body KB. This parameter is important only for thin bodies; it should be given a value of the order of one tenth of the body thickness or less. Insert one line for each thin body in the geometrical structure.

-- Default: DSMAX=1.0E20 (no step length control)

• Interaction forcing

IFORCE ... Activates forcing of interactions of type ICOL of particles KPAR in body KB. FORCER is the forcing factor, $\mathcal{F}$, which must be larger than unity (see Section 6.1.3 of the manual). The values WLOW and WHIG are the limits of the weight window where interaction forcing is applied.

-- Default: no interaction forcing

TRICK: a negative input value of FORCER, $-\mathcal{F}$, is assumed to mean that each particle should interact, on average and approximately, $\mathcal{F}$ times in a path length equal to the range of the particle with $E = \text{SE0}$. This is very useful, *e.g.*, to generate x-ray spectra.

• Distributions of emerging particles

NBE ... Limits, EMIN and EMAX, of the interval where energy distributions are tallied and number of energy bins.

-- Defaults: EMIN=0.0, EMAX=SE0, NBE=100

NBTH ... Number of bins for the polar angle θ .

-- Default: NBTH=90

NBPH ... Number of bins for the azimuthal angle ϕ .

-- Default: NBPH=1 (azimuthal average).

NOTE: In the output files, the terms “transmitted” and “backscattered” are used to denote particles that leave the sample moving upwards ($W > 0$) and downwards ($W < 0$), respectively. Notice that this agrees with the usual meaning of these terms only when electrons impinge on the sample from below (*i.e.*, with $W > 0$).

• Photon detectors

PDANGL ... Starts the definition of a new photon detector. Up to 25 different detectors can be considered.

THETA1, THETA2 and PHI1, PHI2 are the limits of the angular intervals covered by the detector, in degrees.

The integer flag IPSF serves to activate the creation of a phase-space file, which contains the state variables of all particles that enter the detector. Use this option with care, because phase-space files may grow very fast.

IPSF=0; no phase-space file is created.

IPSF=1; a phase-space file is created.

IPSF>1; a psf is created, but contains only state variables of detected photons

that have ILB(4)=IPSF (used for studying angular distributions of characteristic x rays).

NOTE: Generating the psf is useful for tuning interaction forcing, which requires knowing the weights of detected particles.

-- Default: THETA1=35, THETA2=45, PHI1=0, PHI2=360, IPSF=0

PDENER ... Energy limits of the detector

EDEL and EDEU are the lower and upper limits of the energy window covered by the detector. NCHE is the number of channels in the output energy spectrum from the detector ($\leq 1,000$).

-- Default: EDEL=0, EDU=E0, NCHE=100

XRORIG ... This line in the input file activates the generation of a file with the position coordinates of the emission sites of the photons that reach the detector. The file name must be explicitly defined. Notice that the file may grow very fast, so use this option only in short runs. The output file is overwritten when a simulation is resumed.

-- Default: NONE

• Spatial distribution of x-ray emission

The program can generate the space distribution of x-ray emission within a parallelepiped (the scoring box), whose edges are parallel to the axes of the laboratory frame. This box is defined by giving the coordinates of its vertices. The distribution of x-ray generation events is tallied using a uniform orthogonal grid with NX, NY and NZ (.LE. 100) bins along the directions of the coordinate axes.

GRIDX ... *x*-coordinates of the vertices of the scoring box.

-- Default: none

GRIDY ... *y*-coordinates of the vertices of the scoring box.

-- Default: none

GRIDZ ... *z*-coordinates of the vertices of the scoring box.

-- Default: none

GRIDBN ... Numbers of bins NX, NY, and NZ in the *x*, *y* and *z* directions, respectively.

-- Defaults: NX=1, NY=1, NZ=1

XRAYE ... Limits EMIN and EMAX of the energy interval where x rays are scored.

-- Defaults: none

• Job properties

RESUME ... The program will read the dump file `dumpfile1.dat` (up to 20 characters) and resume the simulation from the point where it was left. Use this option very, *very* carefully. Make sure that the input data file is fully consistent with the file used to generate the dump file.

-- Default: off

DUMPTO ... Generate a dump file named `dumpfile2.dat` (name given by the user, up to 20 characters) after completing the simulation run. This allows the simulation

to be resumed to improve statistics.

-- Default: off

NOTE: If the file `dumpfile2.dat` already exists, it is overwritten.

DUMPP ... When the **DUMPTO** option is activated, simulation results are written in the output files every **DUMPP** seconds. This option is useful to check the progress of long simulations. It also allows the program to be run with a long execution time, and to be stopped when the required statistical uncertainty has been reached.

-- Default: **DUMPP**=1.0E15

NSIMSH ... Desired number of simulated showers.

-- Default: **DSHN**=2.0E9

RSEED ... Seeds of the random-number generator.

-- Default: **ISEED1**=1; **ISEED2**=2

TIME ... Allotted simulation time, in sec.

-- Default: **TIMEA**=100.0E0

5 Selecting the simulation parameters

The speed and accuracy of **PENEPMA** is determined by the values of the simulation parameters **EABS(1:3)**, **C1**, **C2**, **WCC**, **WCR**. These parameters are selected by the user for each material in the simulated sample⁴. The parameter **DSMAX**, which is defined by the user for each body, is important only when electrons are transported through very thin bodies. Here we summarise the rules for assigning “safe” values to these parameters (for further details, consult the **PENELOPE** manual).

The electron and photon absorption energies **EABS(1,M)** and **EABS(2,M)** are generally determined by the characteristics of the experiment. In the case that we are interested in determining only the intensity of characteristic x rays, and to speed up the simulation process, we can assign a value slightly lower than the energy of the x rays of interest to **EABS(1,M)**. At the same time **EABS(2,M)** can be given a value slightly lower than that of the x rays, to avoid transporting photons that are not able to contribute to the line of interest. The x-ray spectrum is truncated at an energy equal to **EABS(2,M)**.

The parameters **C1** and **C2** control the simulation of elastic scattering and the tracking algorithm for electrons and positrons. Their allowed values are limited to the interval $[0, 0.2]$. If **C1**=**C2**=0, the simulation of electron histories is performed in detailed mode, *i.e.*, interaction by interaction. If the values of these parameters are not zero, the mixed simulation algorithm is switched on, and the simulation becomes faster. In this case, our recommended practice is to set **C1**=**C2**=0.05, which is fairly conservative. Before increasing the value of any of these parameters, it is advisable to perform short test simulations to verify that with the augmented parameter value the results remain essentially unaltered (and that the simulation runs faster; if there is no gain in speed, keep the conservative values).

⁴To specify simulation parameters for a single body we can simply assign a specific material to this body (the material file may contain multiple definitions of the same material).

As mentioned above, the parameter `DSMAX` has an effect only for very thin bodies and when electrons are tracked using mixed simulation. This parameter defines the maximum step length, and it should be assigned a value of the order of one tenth of the characteristic thickness of the body where electrons move. This ensures that, on average, there will be more than ~ 10 soft events (hinges) along a typical electron track through that body, which is enough to “wash out” the details of the artificial distributions used to sample these events. For thick bodies, it is not necessary to limit the length of the steps, and we can set `DSMAX` = 10^{20} cm (the default value), or some other very large value, to switch off the external step-length control.

`WCC` and `WCR` are the cut-off energy losses for inelastic collisions and bremsstrahlung emission. These parameters have a very weak influence on the accuracy of the results provided only that they are both smaller than the width of the bins used to tally energy distributions. When electron energy distributions are of no interest, our recommendation is to set these cut-off energies equal to one hundredth of the initial energy of primary electrons. Note that, for the sake of consistency, `WCC` should not be larger than the absorption energy of electrons in the material, `EABS(1,M)` (otherwise, we would lose secondary electrons that have energies larger than `EABS(1,M)`). Similarly, `WCR` should be less than the photon absorption energy `EABS(2,M)` (because we want to keep track of bremsstrahlung photons down to that energy).

6 Installation

PENEPMA is distributed as a single ZIP compressed file named `penepma06.zip`, which contains the Fortran source file of the program, some examples and the program `CONVOLG`. The present manual is also included. To install PENEPMA on your computer, inflate (unzip) the file `penepma06.zip` and copy its contents to your hard disk. To obtain the executable binary of PENEPMA, compile and link `penepma.f` with `penelope.f`, `pengeom.f`, `penvared.f` and `timer.f` (these files are in the directory `fsource` of `PENLOPE`). Notice that `penelope.f`, `pengeom.f`, `penvared.f` and `timer.f` are declared through an include statement inside `penepma.f` and do not have to be listed in the compilation command; still, all the associated modules must be in the same directory as the main program `penepma.f`.

As indicated above, `penepma` generates multiple files with simulated probability distributions. Each output file has a heading describing its contents, which is in a format suited for visualisation with a plotting program. We use `GNUPLOT`, which is small in size, available for various platforms (including Linux and Windows) and is free; this software can be downloaded from the distribution sites listed at the Gnuplot Central site, <http://www.gnuplot.info>. PENEPMA is accompanied by `GNUPLOT` scripts that, if `GNUPLOT` has been installed on your computer, can be used to display the probability distributions calculated by the simulation programs. For instance, after running `penepma.exe` you can visualise the simulated photon spectra by simply 1) copying the file `spectra.gnu` from the directory `./penepma/programs` to the directory that contains the results and 2) entering the command “`wgnuplot spectra.gnu`” (in Linux,

“gnuplot spectra.gnu”) or clicking the icon of the script. Note that, depending on the options selected in the PENEPMA input data file, certain distributions may not be generated; in this case, the corresponding GNUPLOT script will not work (nothing will be displayed).

The program PENEPMA and the simulation routines are written in Fortran 77 language, but use a few extensions that are included in most compilers; all these extensions are part of Fortran 95. To generate the executable binary files of your simulation programs you need to have a Fortran compiler installed on your computer. For Win32 (Windows 9x/NT/2000/XP), a number of free compilers are available. We recommend the old-fashioned, but compact and reliable, G77 Fortran 77 compiler⁵ or the Fortran 95 compiler GFORTRAN⁶, both from the Free Software Foundation. Another free compiler, available for different platforms is G95⁷.

It is advisable to open a new directory for each case you are going to simulate. This directory must contain the PENEPMA executable file, the PENEPMA input file, the geometry definition file and the material data file. To visualise the simulated distributions, you will also need to copy the relevant GNUPLOT scripts in the working directory.

7 An example

The example considered here corresponds to an EPMA measurement of a copper-iron sample in the form of a couple (see Fig. 3). The PENEPMA input file, `epma2.tex`, and the associated geometry definition file (`epma2.geo`), are shown below; these files are included in the PENEPMA distribution package.

In this case, the electron beam has an energy of 15 keV and impinges normally on the sample in the direction of the negative z -axis ($\theta = 180^\circ$, see Fig. 1); we assume that the beam has a negligible spot size. Electrons hit the sample surface at the point $x = 0.1 \mu\text{m}$, $y = z = 0$, *i.e.*, on the Cu side and at $0.1 \mu\text{m}$ from the interface. Although this example does not have axial symmetry, we have defined nine annular photon detectors ($\phi_1 = 0$, $\phi_2 = 360^\circ$). The first covers the whole upper hemisphere, *i.e.*, it counts all photons emitted from the sample surface. The other detectors (second to ninth) cover 10° -wide polar-angle intervals. It should be noted that the simulation efficiency of a detector (which is measured by the relative statistical uncertainty of the spectrum) increases with the subtended solid angle; small detectors usually require long simulation times.

- penepma input file (`epma2.in`) for the example described in the text.

```
.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7..
TITLE  A CU-Fe couple
```

⁵<http://www.geocities.com/Athens/Olympus/5564/> At this site you can also find C.G. Page's book, *Professional Programmer's Guide to Fortran 77*, in pdf format.

⁶<http://gcc.gnu.org/wiki/GFortran>

⁷<http://g95.org/>

```

.
>>>>>>> Electron beam definition.
SENERG 15e3 [Energy of the electron beam, in eV]
SPOSIT 1e-5 0 1 [Coordinates of the electron source]
SDIREC 180 0 [Direction angles of the beam axis, in deg]
SAPERT 0 [Beam aperture, in deg]
.
>>>>>>> Material data and simulation parameters.
Up to 10 materials; 2 lines for each material.
MFNAME Cu.mat [Material file, up to 20 chars] &*
MSIMPA 1e3 1e3 1e3 0.2 0.2 1e3 1e3 [EABS(1:3),C1,C2,WCC,WCR] &*
MFNAME Fe.mat [Material file, up to 20 chars] &*
MSIMPA 1e3 1e3 1e3 0.2 0.2 1e3 1e3 [EABS(1:3),C1,C2,WCC,WCR] &*
.
>>>>>>> Geometry of the sample.
GEOMFN epma2.geo [Geometry definition file, 20 chars]
DSMAX 1 1.5e-4 [IB, Maximum step length (cm) in body IB]
DSMAX 2 1.5e-4 [IB, Maximum step length (cm) in body IB]
.
>>>>>>> Interaction forcing.
IFORCE 1 1 4 -4 0.1 1.0 [KB,KPAR,ICOL,FORCER,WLOW,WHIG]
IFORCE 1 1 5 -400 0.1 1.0 [KB,KPAR,ICOL,FORCER,WLOW,WHIG]
IFORCE 1 2 2 -10 1e-4 1.0 [KB,KPAR,ICOL,FORCER,WLOW,WHIG]
IFORCE 1 2 3 -10 1e-4 1.0 [KB,KPAR,ICOL,FORCER,WLOW,WHIG]
IFORCE 2 1 4 -4 0.1 1.0 [KB,KPAR,ICOL,FORCER,WLOW,WHIG]
IFORCE 2 1 5 -400 0.1 1.0 [KB,KPAR,ICOL,FORCER,WLOW,WHIG]
IFORCE 2 2 2 -10 1e-4 1.0 [KB,KPAR,ICOL,FORCER,WLOW,WHIG]
IFORCE 2 2 3 -10 1e-4 1.0 [KB,KPAR,ICOL,FORCER,WLOW,WHIG]
.
>>>>>>> Emerging particles. Energy and angular distributions.
NBE 1e3 15e3 250 [E-interval and no. of energy bins]
NBTH 45 [No. of bins for the polar angle THETA]
NBPH 30 [No. of bins for the azimuthal angle PHI]
.
>>>>>>> Photon detectors (up to 10 different detectors).
IPSF=0, do not create a phase-space file.
IPSF=1, creates a phase-space file.
PDANGL 0 90 0 360 0 [Angular window, in deg, IPSF]
PDENER 0 15e3 1000 [Energy window, no. of channels]
.
PDANGL 5 15 0 360 0 [Angular window, in deg, IPSF]
.
PDANGL 15 25 0 360 0 [Angular window, in deg, IPSF]
.
PDANGL 25 35 0 360 0 [Angular window, in deg, IPSF]
.

```

```

PDANGL 35 45 0 360 0          [Angular window, in deg, IPSF]
.
PDANGL 45 55 0 360 0          [Angular window, in deg, IPSF]
.
PDANGL 55 65 0 360 0          [Angular window, in deg, IPSF]
.
PDANGL 65 75 0 360 0          [Angular window, in deg, IPSF]
.
PDANGL 75 85 0 360 0          [Angular window, in deg, IPSF]
.
>>>>>>> Spatial distribution of events.
GRIDX  -1e-5 5e-5             [X coordinates of the box vertices]
GRIDY  -3e-5 3e-5             [Y coordinates of the box vertices]
GRIDZ  -6e-5 0.               [Z coordinates of the box vertices]
GRIDBN 60 60 60               [Numbers of bins]
XRAYE  8.01e3 8.05e3          [Energy interval where x-rays are tallied]
. K-L2,L3 of Cu
.
>>>>>>> Job properties
RESUME dump1.dmp              [Resume from this dump file, 20 chars]
DUMPTO dump1.dmp              [Generate this dump file, 20 chars]
DUMPP  600                    [Dumping period, in sec]
.
NSIMSH 2.0e9                  [Desired number of simulated showers]
TIME 2.0e5                    [Allotted simulation time, in sec]
....+....1....+....2....+....3....+....4....+....5....+....6....+....7..

```

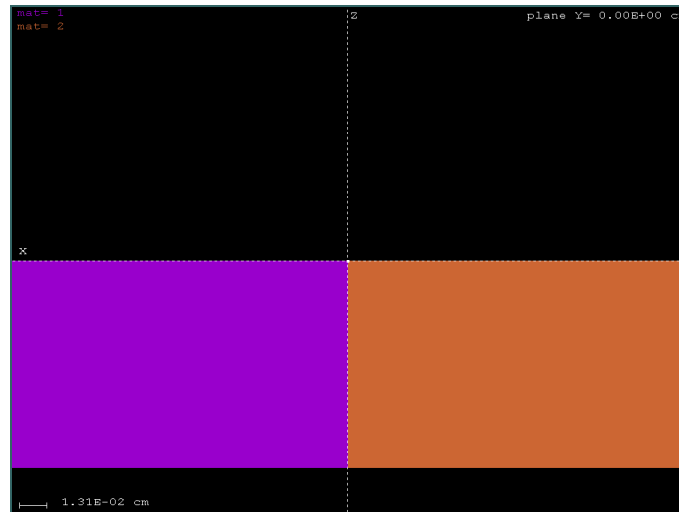


Figure 3: Geometry of the Fe-Cu couple. Notice that the x -axis is pointing towards the left.

- Geometry definition file used in the example (`epma2.geo`).

Figure 4 displays global and characteristic x-ray spectra from the first detector defined in `epma2.in` (which covers the whole upper hemisphere). These x-ray spectra were tallied using 1000 energy channels, which correspond to an energy resolution of 15 eV/channel. As mentioned above, the simulation yields the energy distribution of photons reaching the detector in absolute units, namely the “number of counts” in each

energy bin is the probability of producing a photon that is detected and contributes to that energy bin per incident electron, per unit energy and per unit solid angle. To obtain the intensity of a characteristic line, the value of the intensity at the corresponding energy bin has to be multiplied by the channel width. The spectrum measured by a real detector can be calculated by convolving the simulated (global) spectrum with the energy-response function of the detector using the auxiliary program CONVOLG (see Section 3.2). Figure 5 shows the measured spectrum for the first detector of this example, calculated using the energy resolution function of a typical Si(Li) detector.

Figure 6 shows the angular distribution of photons that leave the sample with any energy. We see that for θ less than about 60 deg, the distribution varies very slowly with the angles; this implies that the counting rate from a small detector would be fairly independent of its position. For θ larger than about 60 deg, the photon intensity decreases when θ increases, due to the increasing effect of photon attenuation within the sample.

PENEPMA generates the space distribution of generation sites of x rays, with energies in a given interval, within a parallelepiped (the scoring box) whose edges are parallel to the axes of the laboratory frame. In the present example the scoring box is $(-0.1 \mu\text{m}, 0.5 \mu\text{m}) \times (-0.3 \mu\text{m}, 0.3 \mu\text{m}) \times (-0.6 \mu\text{m}, 0 \mu\text{m})$. The selected energy interval extends from 8.01 keV to 8.05 keV, which contains the energy of the Cu $K\alpha$ line. It should be noted that the program scores all photons generated with initial energy in this energy interval, including bremsstrahlung photons. Figure 7 shows the x-ray distribution on the plane $z = -0.2 \mu\text{m}$. We see that, at this depth, the electron beam has spread considerably. The few events observed in the Fe part ($x < 0$) of the metal couple are very likely due to high-energy bremsstrahlung photons that are produced in the Cu part and undergo Compton scattering in Fe.

The program also calculates the depth-distribution of x-ray generation (within the scoring box and the selected energy interval). The depth-distribution of characteristic x rays is a quantity of primary importance for conventional quantitative EPMA. However, the depth distribution given by PENEPMA has dimensions of cm^{-1} and it is normalised to the total number of emitted x-rays per incident electron, while the depth-distribution used in conventional EPMA is dimensionless and its integral does not correspond to the total number of emitted x-rays. Figure 8 shows the depth-distribution of x-rays with energies in the considered interval (8.01 to 8.05 keV).

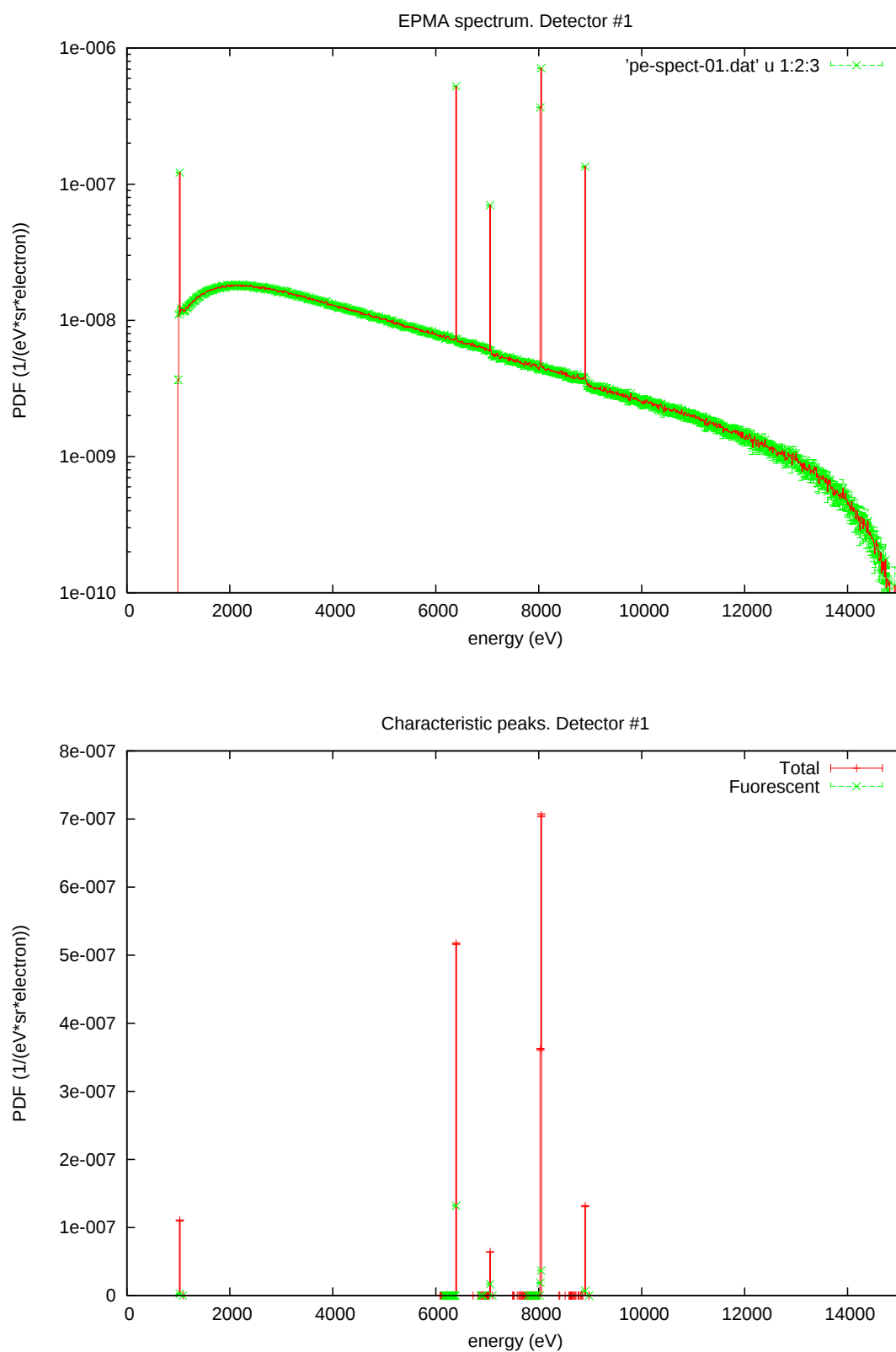


Figure 4: Simulated global x-ray energy spectrum (top) and characteristic x-ray spectrum (bottom) from the Fe-Cu couple. Both spectra were recorded with detector number 1 in the `epma2.in` file. Notice that the characteristic spectrum also displays the contributions from fluorescence photons (crosses).

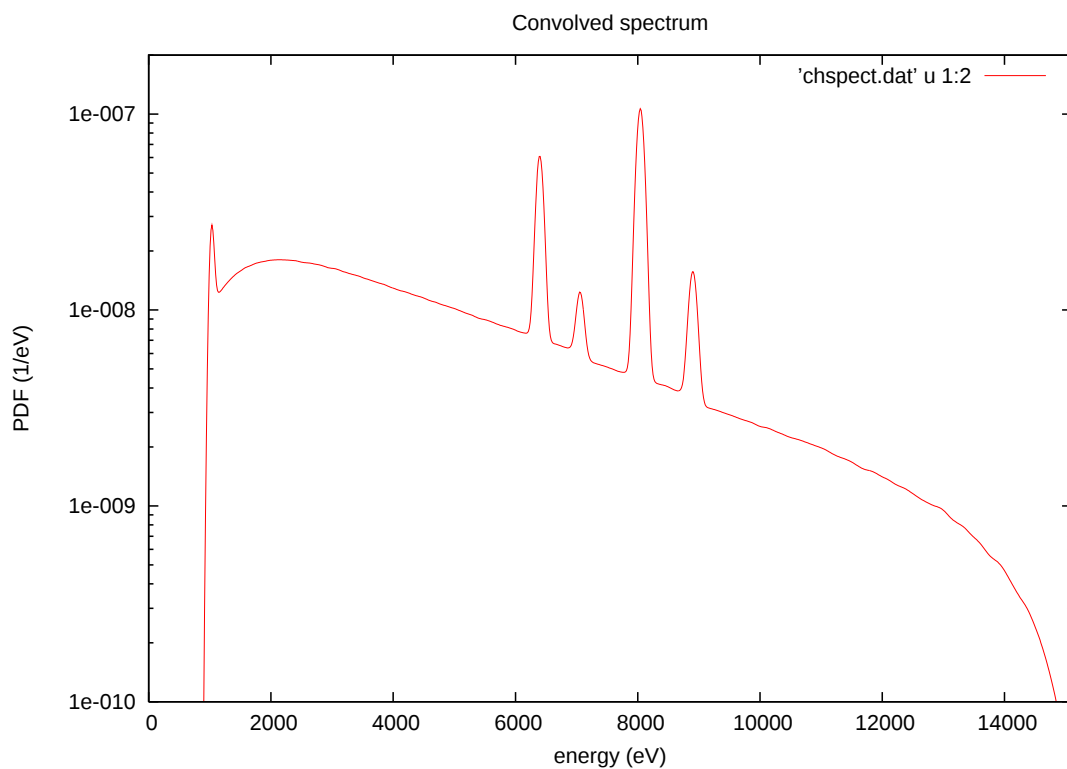


Figure 5: Measured spectrum from detector number 1 (see the PENEPMA input file `epma2.in`) calculated by convolving the simulated spectrum with the energy resolution function of a Si(Li) detector.

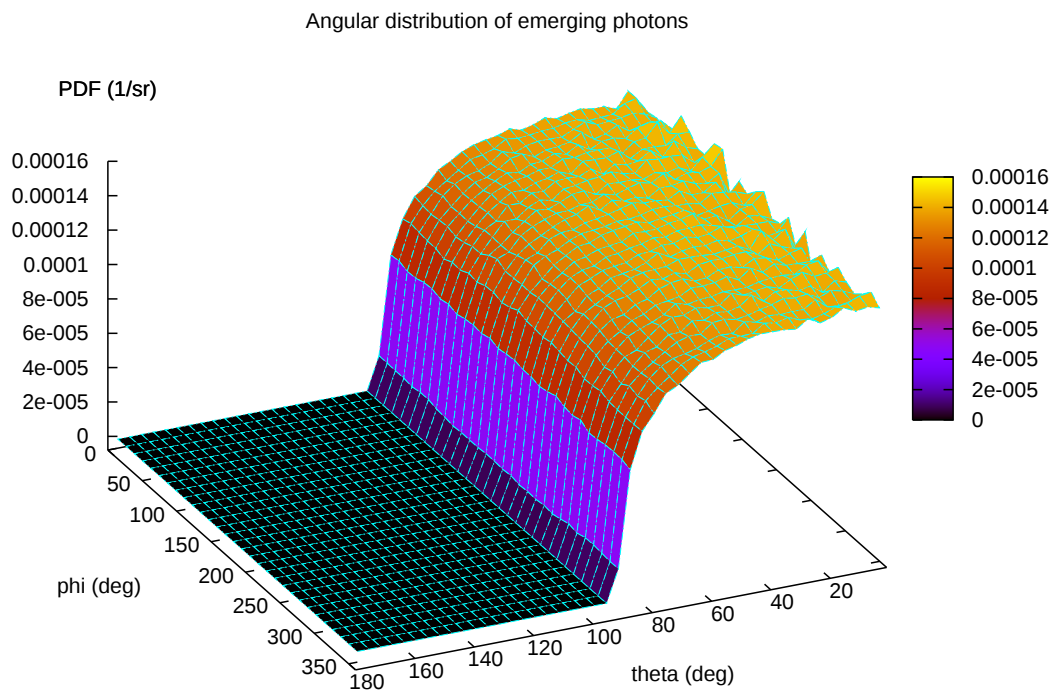


Figure 6: Angular distribution of photons emitted from the sample with any energy.

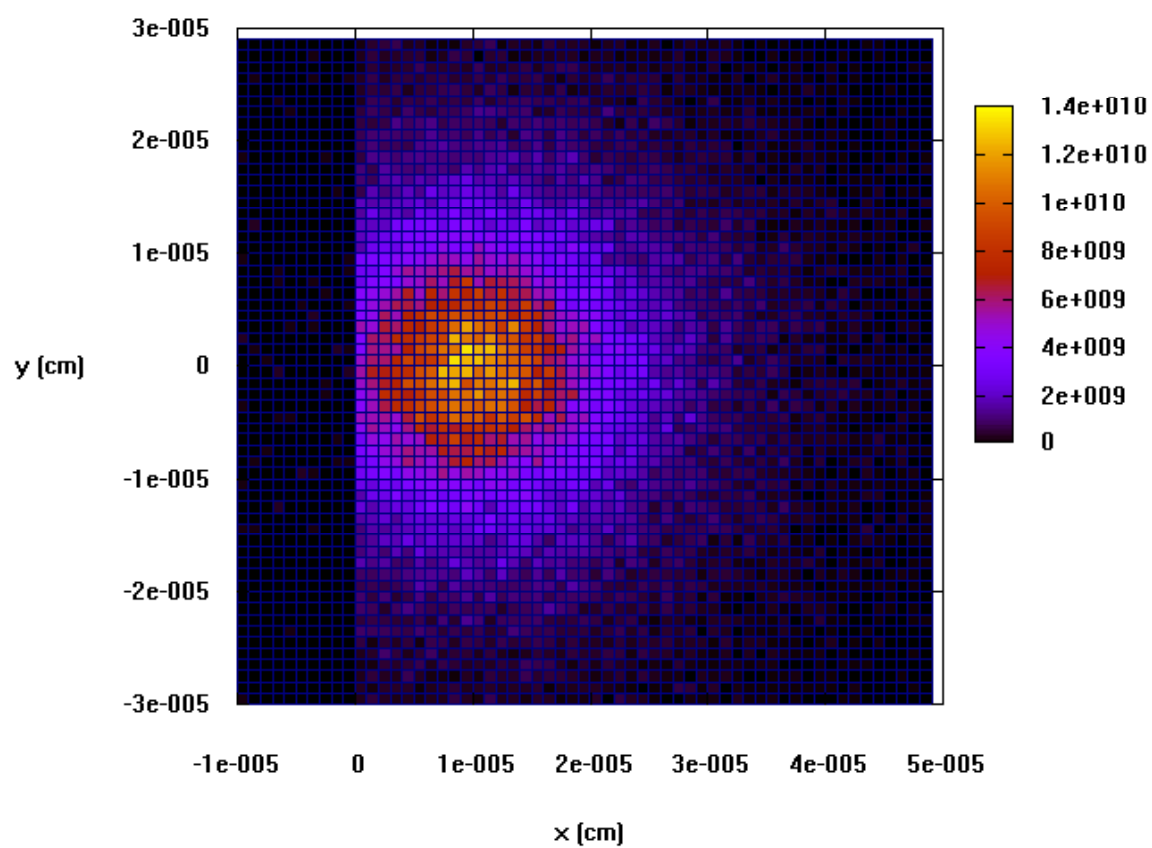


Figure 7: Spatial distribution of x-ray generation in the plane $z = -0.2 \mu\text{m}$, restricted to x rays with initial energy in the interval 8.01–8.05 keV.

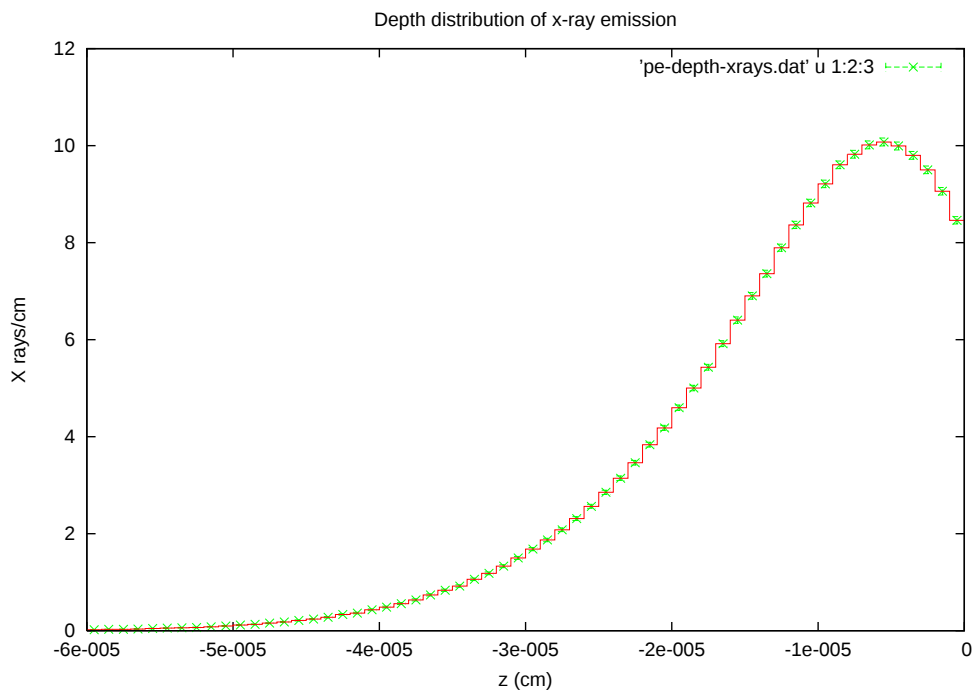


Figure 8: Depth-distribution of x-ray generation, restricted to x rays with initial energy in the interval 8.01–8.05 keV.

Because of the flexibility of the geometry package, PENEPMA can simulate samples with very complicated geometrical structures. The definition of the geometry is routine work and, by following simple rules regarding the structure of the modular tree (see Chapter 5 of the PENELOPE manual), the simulation speed can be made practically independent of the geometrical complexity of the sample. Here we give three examples of geometry definition files for two-material samples, which are similar to geometrical structures found in EPMA samples. These example files are included in the PENEPMA distribution package.

- Particles embedded in a matrix (file `ex1.geo`, see Fig. 9).

1-micron-diameter particles embedded in a matrix

[illegible]

[illegible]

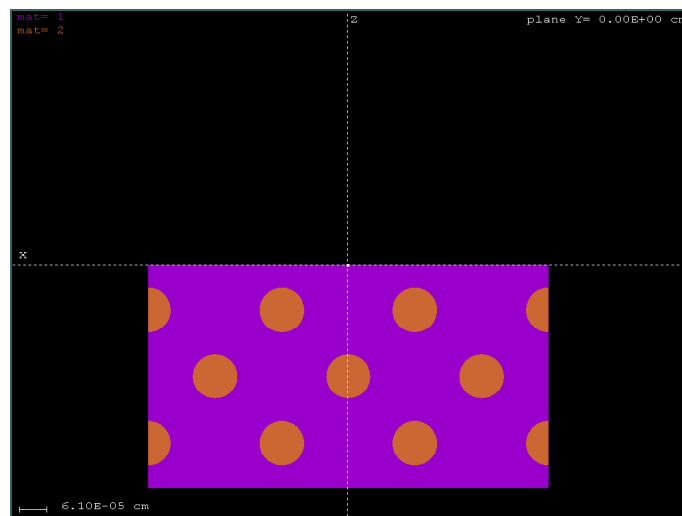


Figure 9: Geometry example `ex1.geo`; particles embedded in a matrix.

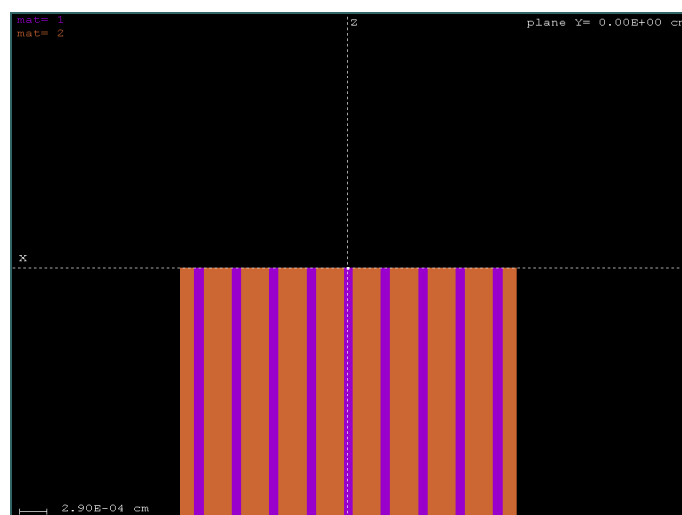
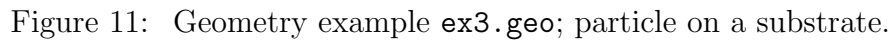


Figure 10: Geometry example `ex2.geo`; lamellae structure.

[illegible]



-+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7.....

[illegible]

- Perkins S.T., D.E. Cullen, M.H. Chen, J.H. Hubbell, J. Rathkopf and J. Scofield (1991), ‘Tables and graphs of atomic subshell and relaxation data derived from the LLNL evaluated atomic data library (EADL), $Z = 1-100$ ’, *Report UCRL-50400* vol. 30 (Lawrence Livermore National Laboratory, Livermore, CA).
- Seltzer S.M. and M.J. Berger (1985), “Bremsstrahlung spectra from electron interactions with screened atomic nuclei and orbital electrons”, *Nucl. Instrum. Meth. B* **12**, 95–134.
- Sternheimer R.M. (1952), “The density effect for the ionization loss in various materials”, *Phys. Rev.* **88**, 851–859.
- Salvat F., A. Jablonski and C.J. Powell (2005), “ELSEPA–Dirac partial-wave calculation of elastic scattering of electrons and positrons by atoms, positive ions and molecules”, *Comput. Phys. Commun.* **165**, 157–190.
- Salvat F., J.M. Fernández-Varea and J. Sempau (2006), “PENELOPE-2006: A code system for Monte Carlo simulation of electron and photon transport”, OECD/NEA Data Bank, Issy-les-Moulineaux, France.
Available in pdf format from <http://www.nea.fr/html/dbprog/penelope.pdf>
-