

PENELOPE-2006: A Code System for Monte Carlo Simulation of Electron and Photon Transport

Workshop Proceedings
Barcelona, Spain
4-7 July 2006



N U C L E A R • E N E R G Y • A G E N C Y

**PENELOPE-2006: A Code System
for Monte Carlo Simulation of
Electron and Photon Transport**

**Workshop Proceedings
Barcelona, Spain
4-7 July 2006**

Francesc Salvat, José M. Fernández-Varea, Josep Sempau
*Facultat de Física (ECM)
Universitat de Barcelona
Spain*

© OECD 2006
NEA No. 6222

NUCLEAR ENERGY AGENCY
ORGANISATION FOR ECONOMIC CO-OPERATION AND DEVELOPMENT

ORGANISATION FOR ECONOMIC CO-OPERATION AND DEVELOPMENT

The OECD is a unique forum where the governments of 30 democracies work together to address the economic, social and environmental challenges of globalisation. The OECD is also at the forefront of efforts to understand and to help governments respond to new developments and concerns, such as corporate governance, the information economy and the challenges of an ageing population. The Organisation provides a setting where governments can compare policy experiences, seek answers to common problems, identify good practice and work to co-ordinate domestic and international policies.

The OECD member countries are: Australia, Austria, Belgium, Canada, the Czech Republic, Denmark, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Japan, Korea, Luxembourg, Mexico, the Netherlands, New Zealand, Norway, Poland, Portugal, the Slovak Republic, Spain, Sweden, Switzerland, Turkey, the United Kingdom and the United States. The Commission of the European Communities takes part in the work of the OECD.

OECD Publishing disseminates widely the results of the Organisation's statistics gathering and research on economic, social and environmental issues, as well as the conventions, guidelines and standards agreed by its members.

* * *

This work is published on the responsibility of the Secretary-General of the OECD. The opinions expressed and arguments employed herein do not necessarily reflect the official views of the Organisation or of the governments of its member countries.

NUCLEAR ENERGY AGENCY

The OECD Nuclear Energy Agency (NEA) was established on 1st February 1958 under the name of the OEEC European Nuclear Energy Agency. It received its present designation on 20th April 1972, when Japan became its first non-European full member. NEA membership today consists of 28 OECD member countries: Australia, Austria, Belgium, Canada, the Czech Republic, Denmark, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Japan, Luxembourg, Mexico, the Netherlands, Norway, Portugal, Republic of Korea, the Slovak Republic, Spain, Sweden, Switzerland, Turkey, the United Kingdom and the United States. The Commission of the European Communities also takes part in the work of the Agency.

The mission of the NEA is:

- to assist its member countries in maintaining and further developing, through international co-operation, the scientific, technological and legal bases required for a safe, environmentally friendly and economical use of nuclear energy for peaceful purposes, as well as
- to provide authoritative assessments and to forge common understandings on key issues, as input to government decisions on nuclear energy policy and to broader OECD policy analyses in areas such as energy and sustainable development.

Specific areas of competence of the NEA include safety and regulation of nuclear activities, radioactive waste management, radiological protection, nuclear science, economic and technical analyses of the nuclear fuel cycle, nuclear law and liability, and public information. The NEA Data Bank provides nuclear data and computer program services for participating countries.

In these and related tasks, the NEA works in close collaboration with the International Atomic Energy Agency in Vienna, with which it has a Co-operation Agreement, as well as with other international organisations in the nuclear field.

© OECD 2006

No reproduction, copy, transmission or translation of this publication may be made without written permission. Applications should be sent to OECD Publishing: rights@oecd.org or by fax (+33-1) 45 24 13 91. Permission to photocopy a portion of this work should be addressed to the Centre Français d'exploitation du droit de Copie, 20 rue des Grands-Augustins, 75006 Paris, France (contact@cfcopies.com).

FOREWORD

The OECD/NEA Data Bank was established to promote effective sharing of data and software developed in member countries in the field of nuclear technology and radiation physics applications. It operates a Computer Program Service (CPS) related to nuclear energy applications. The software library collects, compiles and verifies programs in an appropriate computer environment, ensuring that the computer program package is complete and adequately documented. Internationally agreed quality-assurance methods are used in the verification process.

In order to obtain good results in modelling the behaviour of technological systems, two conditions must be fulfilled:

1. Good quality and validated computer codes and associated basic data libraries should be used.
2. Modelling should be performed by a qualified user of such codes.

One subject to which special effort has been devoted in recent years is radiation transport. Workshops and training courses including the use of computer codes have been organised in the field of neutral particle transport for codes using both deterministic and stochastic methods. The area of charged particle transport, and in particular electron-photon transport, has received increased attention for a number of technological and medical applications.

A new computer code was released to the NEA Data Bank for general distribution in 2001: “PENELOPE, A Code System for Monte Carlo Simulation of Electron and Photon Transport” developed by Francesc Salvat, José M. Fernández-Varea, Eduardo Acosta and Josep Sempau. A first workshop/tutorial was held at the NEA Data Bank in November 2001. This code began to be used very widely by radiation physicists, and users requested that a second PENELOPE workshop with hands-on training be organised. The NEA Nuclear Science Committee endorsed this request while the authors agreed to teach a course covering the physics behind the code and to demonstrate, with corresponding exercises, how it can be used for practical applications. Courses have been organised on an annual basis. New versions of the code have also been presented containing improved physics models and algorithms.

These proceedings contain the corresponding manual and teaching notes of the PENELOPE-2006 workshop and training course, held on 4-7 July 2006 in Barcelona, Spain.

Abstract

The computer code system PENELOPE (version 2006) performs Monte Carlo simulation of coupled electron-photon transport in arbitrary materials for a wide energy range, from a few hundred eV to about 1 GeV. Photon transport is simulated by means of the standard, detailed simulation scheme. Electron and positron histories are generated on the basis of a mixed procedure, which combines detailed simulation of hard events with condensed simulation of soft interactions. A geometry package called PENGEOPE permits the generation of random electron-photon showers in material systems consisting of homogeneous bodies limited by quadric surfaces, i.e. planes, spheres, cylinders, etc. This report is intended not only to serve as a manual of the PENELOPE code system, but also to provide the user with the necessary information to understand the details of the Monte Carlo algorithm.

Keywords: *Radiation transport, electron-photon showers, Monte Carlo simulation, sampling algorithms, quadric geometry*

Symbols and numerical values of constants frequently used in the text (Mohr and Taylor, 2005)

Quantity	Symbol	Value
Avogadro's number	N_A	$6.0221415 \times 10^{23} \text{ mol}^{-1}$
Velocity of light in vacuum	c	$2.99792458 \times 10^8 \text{ m s}^{-1}$
Reduced Planck's constant	$\hbar = h/(2\pi)$	$6.58211915 \times 10^{-16} \text{ eV s}$
Electron charge	e	$1.60217653 \times 10^{-19} \text{ C}$
Electron mass	m_e	$9.1093826 \times 10^{-31} \text{ kg}$
Electron rest energy	$m_e c^2$	510.998918 keV
Classical electron radius	$r_e = e^2/(m_e c^2)$	$2.817940325 \times 10^{-15} \text{ m}$
Fine-structure constant	$\alpha = e^2/(\hbar c)$	$1/137.03599911$
Bohr radius	$a_0 = \hbar^2/(m_e e^2)$	$0.5291772108 \times 10^{-10} \text{ m}$
Hartree energy	$E_h = e^2/a_0$	27.2113845 eV

TABLE OF CONTENTS

Foreword	iii
Preface	ix
1. Monte Carlo simulation. Basic concepts	1
1.1 Elements of probability theory	2
1.1.1 Two-dimensional random variables	5
1.2 Random-sampling methods	6
1.2.1 Random-number generator	6
1.2.2 Inverse-transform method	8
1.2.2.1 Examples	9
1.2.3 Discrete distributions	10
1.2.3.1 Walker's aliasing method	12
1.2.4 Numerical inverse transform for continuous PDFs	13
1.2.4.1 Determining the interpolation grid	15
1.2.4.2 Sampling algorithm	16
1.2.5 Rejection methods	18
1.2.6 Two-dimensional variables. Composition methods	20
1.2.6.1 Examples	21
1.3 Monte Carlo integration	23
1.3.1 Monte Carlo vs. numerical quadrature	26
1.4 Simulation of radiation transport	29
1.4.1 Interaction cross sections	29
1.4.2 Mean free path	31
1.4.3 Scattering model and probability distributions	32
1.4.4 Generation of random track	34
1.4.5 Particle transport as a Markov process	36
1.5 Statistical averages and uncertainties	38
1.6 Variance reduction	41
1.6.1 Interaction forcing	42
1.6.2 Splitting and Russian roulette	43
1.6.3 Other methods	44

2. Photon interactions.....	45
2.1 Coherent (Rayleigh) scattering	46
2.1.1 Simulation of coherent scattering events	49
2.2 Photoelectric effect	50
2.2.1 Simulation of photoelectron emission	52
2.2.1.1 Initial direction of photoelectrons	53
2.3 Incoherent (Compton) scattering.....	54
2.3.1 Analytical Compton profiles.....	60
2.3.2 Simulation of incoherent scattering events.....	62
2.4 Electron-positron pair production	65
2.4.1 Simulation of pair-production events	70
2.4.1.1 Angular distribution of the produced particles	72
2.4.1.2 Compound materials	72
2.5 Attenuation coefficients	73
2.6 Atomic relaxation.....	76
3. Electron and positron interactions.....	81
3.1 Elastic collisions	82
3.1.1 Partial-wave cross sections	86
3.1.1.1 Simulation of single-scattering events	90
3.1.2 The modified Wentzel (MW) model	92
3.1.2.1 Simulation of single elastic events with the MW model.....	96
3.2 Inelastic collisions.....	97
3.2.1 GOS model	101
3.2.2 Differential cross sections.....	105
3.2.2.1 DCS for close collisions of electrons.....	106
3.2.2.2 DCS for close collisions of positrons.....	107
3.2.3 Integrated cross sections.....	108
3.2.4 Stopping power of high-energy electrons and positrons.....	113
3.2.5 Simulation of hard inelastic collisions.....	115
3.2.5.1 Hard distant interactions	115
3.2.5.2 Hard close collisions of electrons	116
3.2.5.3 Hard close collisions of positrons	118
3.2.5.4 Secondary electron emission.....	119
3.2.6 Ionisation of inner shells.....	120

3.3	Bremsstrahlung emission	123
3.3.1	The energy-loss scaled DCS	124
3.3.2	Integrated cross sections	127
3.3.2.1	CSDA range	127
3.3.3	Angular distribution of emitted photons	129
3.3.4	Simulation of hard radiative events	131
3.3.4.1	Sampling of the photon energy	133
3.3.4.2	Angular distribution of emitted photons	134
3.4	Positron annihilation	135
3.4.1	Generation of emitted photons	136
4.	Electron/positron transport mechanics	139
4.1	Elastic scattering	140
4.1.1	Multiple elastic scattering theory	140
4.1.2	Mixed simulation of elastic scattering	141
4.1.2.1	Angular deflections in soft scattering events	145
4.1.3	Simulation of soft events	146
4.2	Soft energy losses	149
4.2.1	Energy dependence of the soft DCS	153
4.3	Combined scattering and energy loss	155
4.3.1	Variation of $\lambda_T^{(h)}$ with energy	157
4.3.2	Scattering by atomic electrons	160
4.3.3	Bielajew's alternate random hinge	163
4.4	Generation of random tracks	163
4.4.1	Stability of the simulation algorithm	166
5.	Constructive quadric geometry	169
5.1	Rotations and translations	171
5.2	Quadric surfaces	173
5.3	Constructive quadric geometry	176
5.4	Geometry-definition file	179
5.5	The subroutine package PENGEOm	184
5.5.1	Impact detectors	188
5.6	Debugging and viewing the geometry	189
5.7	A short tutorial	191

6. Structure and operation of the code system	197
6.1 PENELOPE.....	198
6.1.1 Database and input material data file.....	199
6.1.2 Structure of the main program	207
6.1.3 Variance reduction.....	216
6.2 Examples of main programs	218
6.2.1 Program penslab.....	219
6.2.1.1 Structure of the input file	221
6.2.2 Program pencyl	224
6.2.2.1 Structure of the input file	226
6.2.2.2 Example	230
6.2.3 Program penmain.....	231
6.2.3.1 Structure of the input file	233
6.3 Selecting the simulation parameters.....	239
6.4 The code SHOWER.....	244
6.5 Installation.....	244
A. Collision kinematics.....	249
A.1 Two-body reactions.....	250
A.1.1 Elastic scattering.....	252
A.2 Inelastic collisions of charged particles	253
B. Numerical tools	257
B.1 Cubic spline interpolation	257
B.2 Numerical quadrature.....	261
B.2.1 Gauss integration	261
B.2.2 Adaptive bipartition	262
C. Electron/positron transport in electromagnetic fields.....	263
C.1 Tracking particles in vacuum	264
C.1.1 Uniform electric fields.....	266
C.1.2 Uniform magnetic fields.....	268
C.2 Exact tracking in homogeneous magnetic fields.....	269
Bibliography.....	273

PREFACE

Radiation transport in matter has been a subject of intense work since the beginning of the 20th century. Nowadays, we know that high-energy photons, electrons and positrons penetrating matter suffer multiple interactions by which energy is transferred to the atoms and molecules of the material and secondary particles are produced¹. By repeated interaction with the medium, a high-energy particle originates a cascade of particles which is usually referred to as a shower. In each interaction, the energy of the particle is reduced and further particles may be generated so that the evolution of the shower represents an effective degradation in energy. As time goes on, the initial energy is progressively deposited into the medium, while that remaining is shared by an increasingly larger number of particles.

A reliable description of shower evolution is required in a number of fields. Thus, knowledge of radiation transport properties is needed for quantitative analysis in surface electron spectroscopies (Jablonski, 1987; Tofterup, 1986), positron surface spectroscopy (Schultz and Lynn, 1988), electron microscopy (Reimer, 1985), electron energy loss spectroscopy (Reimer *et al.*, 1992), electron probe microanalysis (Heinrich and Newbury, 1991), etc. Detailed information on shower evolution is also required for the design and quantitative use of radiation detectors (Titus, 1970; Berger and Seltzer, 1972). A field where radiation transport studies play an important sociological role is that of radiation dosimetry and radiotherapy (Andreo, 1991).

The study of radiation transport problems was initially attempted on the basis of the Boltzmann transport equation. However, this procedure comes up against considerable difficulties when applied to limited geometries, with the result that numerical methods based on the transport equation have only had certain success in simple geometries, mainly for unlimited and semi-infinite media (see, *e.g.*, Zheng-Ming and Brahme, 1993). At the end of the 1950s, with the availability of computers, Monte Carlo simulation methods were developed as a powerful alternative to deal with transport problems. Basically, the evolution of an electron-photon shower is of a random nature, so that this is a process that is particularly amenable to Monte Carlo simulation. Detailed simulation, where all the interactions experienced by a particle are simulated in chronological succession, is exact, *i.e.*, it yields the same results as the rigorous solution of the transport equation (apart from the inherent statistical uncertainties).

To our knowledge, the first numerical Monte Carlo simulation of photon transport is that of Hayward and Hubbell (1954) who generated 67 photon histories using a desk calculator. The simulation of photon transport is straightforward since the mean number of events in each history is fairly small. Indeed, the photon is effectively absorbed after a single photoelectric or pair-production interaction or after a few Compton interactions (say, of the order of 10). With present-day computational facilities, detailed simulation of photon transport is a simple routine task.

The simulation of electron and positron transport is much more difficult than that of photons. The main reason is that the average energy loss of an electron in a single interaction is very small (of the order of a few tens of eV). As a consequence, high-energy electrons suffer a large number of interactions before being effectively absorbed in the medium. In practice, detailed simulation is feasible only when the average number of collisions per track is not too large (say, up to a few hundred). Experimental

¹ In this report, the term particle will be used to designate either photons, electrons or positrons.

situations which are amenable to detailed simulation are those involving either electron sources with low initial kinetic energies (up to about 100 keV) or special geometries such as electron beams impinging on thin foils. For larger initial energies, and thick geometries, the average number of collisions experienced by an electron until it is effectively stopped becomes very large, and detailed simulation is very inefficient.

For high-energy electrons and positrons, most of the Monte Carlo codes currently available [*e.g.*, ETRAN (Berger and Seltzer, 1988), ITS3 (Halbleib *et al.*, 1992), EGS4 (Nelson *et al.*, 1985), GEANT3 (Brun *et al.*, 1986), EGSnrc (Kawrakow and Rogers, 2001), MCNP (X-5 Monte Carlo Team, 2003), GEANT4 (Agostinelli *et al.*, 2003; Allison *et al.*, 2006), FLUKA (Ferrari *et al.*, 2005), EGS5 (Hirayama *et al.*, 2005), ...] have recourse to multiple-scattering theories which allow the simulation of the global effect of a large number of events in a track segment of a given length (step). Following Berger (1963), these simulation procedures will be referred to as “condensed” Monte Carlo methods. The multiple-scattering theories implemented in condensed simulation algorithms are only approximate and may lead to systematic errors, which can be made evident by the dependence of the simulation results on the adopted step length (Bielajew and Rogers, 1987). To analyse their magnitude, one can perform simulations of the same arrangement with different step lengths. The results are usually found to stabilise when the step length is reduced, while computation time increases rapidly, roughly in proportion to the inverse of the step length. Thus, for each particular problem, one must reach a certain compromise between available computer time and attainable accuracy. It is also worth noting that, owing to the nature of certain multiple-scattering theories and/or to the particular way they are implemented in the simulation code, the use of very short step lengths may introduce spurious effects in the simulation results. For instance, the multiple-elastic-scattering theory of Molière (1948), which is the model used in EGS4-based codes, is not applicable to step lengths shorter than a few times the elastic mean free path (see, *e.g.*, Fernández-Varea *et al.*, 1993b) and multiple elastic scattering has to be switched off when the step length becomes smaller than this value. As a consequence, stabilisation for short step lengths does not necessarily imply that simulation results are correct. Condensed schemes also have difficulties in generating particle tracks in the vicinity of an interface, *i.e.*, a surface separating two media of different compositions. When the particle moves near an interface, the step length must be kept smaller than the *minimum* distance to the interface so as to make sure that the step is completely contained in the initial medium (Bielajew and Rogers, 1987). This may complicate the code considerably, even for relatively simple geometries.

In the present report, we describe the 2006 version of PENELOPE, a Monte Carlo algorithm and computer code for the simulation of coupled electron-photon transport. The name is an acronym that stands for PENetration and Energy LOss of Positrons and Electrons (photon simulation was introduced later). The simulation algorithm is based on a scattering model that combines numerical databases with analytical cross section models for the different interaction mechanisms and is applicable to energies (kinetic energies in the case of electrons and positrons) from a few hundred eV to ~1 GeV. Photon transport is simulated by means of the conventional detailed method. The simulation of electron and positron transport is performed by means of a mixed procedure. Hard interactions, with scattering angle θ or energy loss W greater than pre-selected cutoff values θ_c and W_c , are simulated in detail. Soft interactions, with scattering angle or energy loss less than the corresponding cutoffs, are described by means of multiple-scattering approaches. This simulation scheme handles lateral displacements and interface crossing appropriately and provides a consistent description of energy straggling. The simulation is stable under variations of the cutoffs θ_c , W_c and these can be made quite large, thus speeding up the calculation considerably, without altering the results. A characteristic feature of our code is that the most delicate parts of the simulation are handled internally; electrons, positrons and photons are simulated by calling the same subroutines. Thus, from the users’ point of view, PENELOPE makes the practical simulation of electrons and positrons as simple as that of photons (although simulating a charged particle may take a longer time).

The present version of PENELOPE is the result of continued evolution from the first version, which was released in 1996. It incorporates substantial changes and additions to the previous versions (1996, 2000, 2001, 2003 and 2005), which aim at improving both reliability and generality of the code system. As for the physics, elastic scattering of electrons and positrons (with energies up to 100 MeV) is now described by using a numerical database of differential cross sections, which was generated using the relativistic partial-wave code ELSEPA (Salvat *et al.*, 2005). The ionisation of K, L and M shells by photoelectric absorption and by electron or positron impact is described from the corresponding partial cross sections, and fluorescence radiation from vacancies in K, L and M shells is followed. Random sampling from numerical distributions is now performed by using the RITA algorithm (Rational Inverse Transform with Aliasing, described in Section 1.2.4). The distribution package includes three examples of main programs: `penslab` (which simulates electron-photon transport in a slab), `pencyl` (for transport in cylindrical geometries), and `penmain` (for generic quadric geometries).

This report is intended not only to serve as a manual of the simulation package, but also to provide the user with the necessary information to understand the details of the Monte Carlo algorithm. In Chapter 1 we give a brief survey of random sampling methods and an elementary introduction to Monte Carlo simulation of radiation transport. The cross sections adopted in PENELOPE to describe particle interactions, and the associated sampling techniques, are presented in Chapters 2 and 3². Chapter 4 is devoted to mixed simulation methods for electron and positron transport. In Chapter 5, a relatively simple, but effective, method to handle simulation in quadric geometries is presented. The Fortran 77 simulation package PENELOPE, the example main programs, and other complementary tools are described in Chapter 6, which also provides instructions to operate them. Information on relativistic kinematics and numerical methods is given in Appendices A and B. Finally, Appendix C is devoted to simulation of electron/positron transport under external, static electric and magnetic fields. The Fortran source files of PENELOPE (and the auxiliary programs and subroutine packages), the database, various complementary tools, and the code documentation are supplied on a ZIP-compressed file, which is distributed by the NEA Data Bank³ and the RSICC⁴. The code is also available from the authors, but we would appreciate it if users did try to get the code from these institutions.

In the course of our Monte Carlo research, we have had the good fortune of obtaining much help from numerous friends and colleagues. Since the mid 1980's, we have benefited from discussions with D. Liljequist, which gave shape to our first algorithm for simulation of electrons and positrons. We are particularly grateful to A. Riveros for his enthusiastic and friendly support over the years, and for guiding us into the field of microanalysis and x-ray simulation. Stimulating collaboration with A.F. Bielajew led to substantial improvements in the electron transport mechanics and in the code organisation. We are deeply indebted to J.H. Hubbell and D.E. Cullen for kindly providing us with updated information on photon interaction and atomic relaxation data. Thanks are also due to S.M. Seltzer for sending us his bremsstrahlung energy-loss database. We are especially indebted to P. Andreo for many comments and suggestions, which have been of much help in improving the code, and for providing a preliminary version of the tutorial. Many subtleties of the manual were clarified thanks to the helpful advice of A. Lallena. A. Sánchez-Reyes and E. García-Toraño were the first external users of the code system; they suffered the inconveniences of using continuously changing preliminary versions of the code without complaining too much. L. Sorbier and C. Campos contributed to improve the description of x-ray emission in the 2003 version. Our most sincere appreciation to the members of our research group; X. Llovet, Ll. Brualla, D. Bote, A. Badal, and F. Al-Dweri. They not only chased bugs through the

² In these Chapters, and in other parts of the text, the CGS Gaussian system of units is adopted.

³ OECD Nuclear Energy Agency Data Bank. Le Seine Saint-Germain, 12 Bd. des Iles. 92130 Issy-les-Moulineaux, France. E-mail: nea@nea.fr; <http://www.nea.fr>

⁴ Radiation Safety Information Computational Center. PO Box 2008, Oak Ridge, TN 37831-6362, USA. E-mail: pdcc@ornl.gov; <http://www-rsicc.ornl.gov>

programs and documentation, but also suggested many practical improvements. Last but not least, we are also indebted to J. Baró and E. Acosta, who took part in the development of previous versions of the code.

Finally, we would like to thank the staff of the NEA Data Bank, particularly E. Sartori, J. Galán and C. Rocher, for their help in distributing the code and for kindly organising the training courses on PENELOPE.

Partial support from the Fondo de Investigación Sanitaria (Ministerio de Sanidad y Consumo, Spain), project no. 03/0676, is gratefully acknowledged. Various parts of the code system were developed within the framework of the European Integrated Project MAESTRO (Methods and Advanced Equipment for Simulation and Treatment in Radiation Oncology), which is granted by the Commission of the European Communities (contract no. LSHC-CT-2004-503564).

Barcelona, May 2006

Chapter 1

Monte Carlo simulation. Basic concepts

The name “Monte Carlo” was coined in the 1940s by scientists working on the nuclear-weapon project in Los Alamos to designate a class of numerical methods based on the use of random numbers. Nowadays, Monte Carlo methods are widely used to solve complex physical and mathematical problems (James, 1980; Rubinstein, 1981; Kalos and Whitlock, 1986), particularly those involving multiple independent variables where more conventional numerical methods would demand formidable amounts of memory and computer time. The book by Kalos and Whitlock (1986) gives a readable survey of Monte Carlo techniques, including simple applications in radiation transport, statistical physics and many-body quantum theory.

In Monte Carlo simulation of radiation transport, the history (track) of a particle is viewed as a random sequence of free flights that end with an interaction event where the particle changes its direction of movement, loses energy and, occasionally, produces secondary particles. The Monte Carlo simulation of a given experimental arrangement (*e.g.*, an electron beam, coming from an accelerator and impinging on a water phantom) consists of the numerical generation of random histories. To simulate these histories we need an “interaction model”, *i.e.*, a set of differential cross sections (DCS) for the relevant interaction mechanisms. The DCSs determine the probability distribution functions (PDF) of the random variables that characterise a track; 1) free path between successive interaction events, 2) type of interaction taking place and 3) energy loss and angular deflection in a particular event (and initial state of emitted secondary particles, if any). Once these PDFs are known, random histories can be generated by using appropriate sampling methods. If the number of generated histories is large enough, quantitative information on the transport process may be obtained by simply averaging over the simulated histories.

The Monte Carlo method yields the same information as the solution of the Boltzmann transport equation, with the same interaction model, but is easier to implement (Berger, 1963). In particular, the simulation of radiation transport in complex geome-

tries is straightforward, while even the simplest finite geometries (*e.g.*, thin foils) are very difficult to be dealt with by the transport equation. The main drawback of the Monte Carlo method lies in its random nature: all the results are affected by statistical uncertainties, which can be reduced at the expense of increasing the sampled population and, hence, the computation time. Under special circumstances, the statistical uncertainties may be lowered by using variance-reduction techniques (Rubinstein, 1981; Bielajew and Rogers, 1988).

This Chapter contains a general introduction to Monte Carlo methods and their application to radiation transport. We start with a brief review of basic concepts in probability theory, which is followed by a description of generic random sampling methods and algorithms. In Section 1.3 we consider the calculation of multidimensional integrals by Monte Carlo methods and we derive general formulas for the evaluation of statistical uncertainties. In Section 1.4 we present the essentials of detailed Monte Carlo algorithms for the simulation of radiation transport in matter. The last Sections of this Chapter are devoted to the evaluation of statistical uncertainties and the use of variance-reduction techniques in radiation transport studies.

1.1 Elements of probability theory

The essential characteristic of Monte Carlo simulation is the use of random numbers and random variables. A random variable is a quantity that results from a repeatable process and whose actual values (realisations) cannot be predicted with certainty. In the real world, randomness originates either from uncontrolled factors (as occurs, *e.g.*, in games of chance) or from the quantum nature of microscopic systems and processes (*e.g.*, nuclear disintegration and radiation interactions). As a familiar example, assume that we throw two dice in a box; the sum of points on their upper faces is a discrete random variable, which can take the values 2 to 12, while the distance x between the dice is a continuous random variable, which varies between zero (dice in contact) and a maximum value determined by the dimensions of the box. On a computer, random variables are generated by means of numerical transformations of random numbers (see below).

Let x be a continuous random variable that takes values in the interval $x_{\min} \leq x \leq x_{\max}$. To measure the likelihood of obtaining x in an interval (a, b) we use the probability $P\{x|a < x < b\}$, defined as the ratio n/N of the number n of values of x that fall within that interval and the total number N of generated x -values, in the limit $N \rightarrow \infty$. The probability of obtaining x in a differential interval of length dx about x_1 can be expressed as

$$P\{x|x_1 < x < x_1 + dx\} = p(x_1) dx, \quad (1.1)$$

where $p(x)$ is the PDF of x . Since 1) negative probabilities have no meaning and 2) the obtained value of x must be somewhere in $(x_{\min}, x_{\max})$, the PDF must be definite

positive and normalised to unity, *i.e.*,

$$p(x) \geq 0 \quad \text{and} \quad \int_{x_{\min}}^{x_{\max}} p(x) \, dx = 1. \quad (1.2)$$

Any “function” that satisfies these two conditions can be interpreted as a PDF. In Monte Carlo simulation we shall frequently use the uniform distribution,

$$U_{x_{\min}, x_{\max}}(x) \equiv \begin{cases} 1/(x_{\max} - x_{\min}) & \text{if } x_{\min} \leq x \leq x_{\max}, \\ 0 & \text{otherwise,} \end{cases} \quad (1.3)$$

which is discontinuous. The definition (1.2) also includes singular distributions such as the Dirac delta, $\delta(x - x_0)$, which is defined by the property

$$\int_a^b f(x) \delta(x - x_0) \, dx = \begin{cases} f(x_0) & \text{if } a < x_0 < b, \\ 0 & \text{if } x_0 < a \text{ or } x_0 > b \end{cases} \quad (1.4)$$

for any function $f(x)$ that is continuous at x_0 . An equivalent, more intuitive definition is the following,

$$\delta(x - x_0) \equiv \lim_{\Delta \rightarrow 0} U_{x_0 - \Delta, x_0 + \Delta}(x), \quad (1.4')$$

which represents the delta distribution as the zero-width limit of a sequence of uniform distributions centred at the point x_0 . Hence, the Dirac distribution describes a single-valued discrete random variable (*i.e.*, a constant). The PDF of a random variable x that takes the discrete values $x = x_1, x_2, \dots$ with point probabilities $p_1, p_2, \dots$ can be expressed as a mixture of delta distributions,

$$p(x) = \sum_i p_i \delta(x - x_i). \quad (1.5)$$

Discrete distributions can thus be regarded as particular forms of continuous distributions.

Given a continuous random variable x , the cumulative distribution function of x is defined by

$$\mathcal{P}(x) \equiv \int_{x_{\min}}^x p(x') \, dx'. \quad (1.6)$$

This is a non-decreasing function of x that varies from $\mathcal{P}(x_{\min}) = 0$ to $\mathcal{P}(x_{\max}) = 1$. In the case of a discrete PDF of the form (1.5), $\mathcal{P}(x)$ is a step function. Notice that the probability $P\{x | a < x < b\}$ of having x in the interval (a, b) is

$$P\{x | a < x < b\} = \int_a^b p(x) \, dx = \mathcal{P}(b) - \mathcal{P}(a), \quad (1.7)$$

and that $p(x) = d\mathcal{P}(x)/dx$.

The n -th moment of $p(x)$ is defined as

$$\langle x^n \rangle \equiv \int_{x_{\min}}^{x_{\max}} x^n p(x) dx. \quad (1.8)$$

The moment $\langle x^0 \rangle$ is simply the integral of $p(x)$, which is equal to unity, by definition. However, higher-order moments may or may not exist. An example of a PDF that has no even-order moments is the Lorentz or Cauchy distribution,

$$p_L(x) \equiv \frac{1}{\pi} \frac{\gamma}{\gamma^2 + x^2}, \quad -\infty < x < \infty. \quad (1.9)$$

Its first moment, and other odd-order moments, can be assigned a finite value if they are defined as the “principal value” of the integrals, *e.g.*,

$$\langle x \rangle_L = \lim_{a \rightarrow \infty} \int_{-a}^{+a} x \frac{1}{\pi} \frac{\gamma}{\gamma^2 + x^2} dx = 0. \quad (1.10)$$

However, the second and higher even-order moments are infinite, irrespective of the way they are defined.

The first moment, when it exists, is called the mean or expected value of the random variable x ,

$$\langle x \rangle = \int x p(x) dx. \quad (1.11)$$

The expected value of a function $f(x)$ is defined in a similar way,

$$\langle f(x) \rangle \equiv \int f(x) p(x) dx. \quad (1.12)$$

Since $f(x)$ is a random variable, it has its own PDF, $\pi(f)$, which is such that the probability of having f in a certain interval of length df is equal to the probability of having x in the corresponding interval or intervals¹. Thus, if $f(x)$ is a monotonically increasing function of x (so that there is a one-to-one correspondence between the values of x and f), $p(x) dx = \pi(f) df$ and

$$\pi(f) = p(x) (df/dx)^{-1}. \quad (1.13)$$

It can be shown that the definitions (1.11) and (1.12) are equivalent. If $f(x)$ increases monotonically with x , the proof is trivial: we can start from the definition (1.11) and write

$$\langle f \rangle = \int f \pi(f) df = \int f(x) p(x) dx,$$

which agrees with (1.12). Notice that the expectation value is linear, *i.e.*,

$$\langle a_1 f_1(x) + a_2 f_2(x) \rangle = a_1 \langle f_1(x) \rangle + a_2 \langle f_2(x) \rangle, \quad (1.14)$$

¹When $f(x)$ does not increase or decrease monotonically with x , there may be multiple values of x corresponding to a given value of f .

where a_1 and a_2 are arbitrary real constants.

If the first and second moments of the PDF $p(x)$ exist, we define the variance of x [or of $p(x)$] by

$$\text{var}(x) \equiv \langle (x - \langle x \rangle)^2 \rangle = \int (x - \langle x \rangle)^2 p(x) dx = \langle x^2 \rangle - \langle x \rangle^2. \quad (1.15)$$

The square root of the variance, $\sigma \equiv [\text{var}(x)]^{1/2}$, is called the “standard deviation” (and sometimes the “standard uncertainty”); it gives a measure of the dispersion of the random variable (*i.e.*, of the width of the PDF). The Dirac delta is the only PDF that has zero variance. Similarly, the variance of a function $f(x)$ is defined as

$$\text{var}\{f(x)\} = \langle f^2(x) \rangle - \langle f(x) \rangle^2. \quad (1.16)$$

Thus, for a constant $f(x) = a$, $\langle f \rangle = a$ and $\text{var}\{f\} = 0$.

1.1.1 Two-dimensional random variables

Let us now consider the case of a two-dimensional random variable, (x, y) . The corresponding (joint) PDF $p(x, y)$ satisfies the conditions

$$p(x, y) \geq 0 \quad \text{and} \quad \int dx \int dy p(x, y) = 1. \quad (1.17)$$

The *marginal* PDFs of x and y are defined as

$$q(x) \equiv \int p(x, y) dy \quad \text{and} \quad q(y) \equiv \int p(x, y) dx, \quad (1.18)$$

i.e., $q(x)$ is the probability of obtaining the value x and *any* value of y . The joint PDF can be expressed as

$$p(x, y) = q(x) p(y|x) = q(y) p(x|y), \quad (1.19)$$

where

$$p(x|y) = \frac{p(x, y)}{q(y)} \quad \text{and} \quad p(y|x) = \frac{p(x, y)}{q(x)} \quad (1.20)$$

are the *conditional* PDFs of x and y , respectively. Notice that $p(x|y)$ is the normalised PDF of x for a fixed value of y .

The expectation value of a function $f(x, y)$ is

$$\langle f(x, y) \rangle = \int dx \int dy f(x, y) p(x, y). \quad (1.21)$$

The moments of the PDF are defined by

$$\langle x^n y^m \rangle = \int dx \int dy x^n y^m p(x, y). \quad (1.22)$$

In particular,

$$\langle x^n \rangle = \int dx \int dy x^n p(x, y) = \int x^n q(x) dx. \quad (1.23)$$

Again, the only moment that is necessarily defined is $\langle x^0 y^0 \rangle = 1$. When the corresponding moments exist, the variances of x and y are given by

$$\text{var}(x) = \langle x^2 \rangle - \langle x \rangle^2 \quad \text{and} \quad \text{var}(y) = \langle y^2 \rangle - \langle y \rangle^2. \quad (1.24)$$

The variance of $x + y$ is

$$\text{var}(x + y) = \langle (x + y)^2 \rangle - \langle x + y \rangle^2 = \text{var}(x) + \text{var}(y) + 2 \text{cov}(x, y), \quad (1.25)$$

where

$$\text{cov}(x, y) = \langle xy \rangle - \langle x \rangle \langle y \rangle \quad (1.26)$$

is the *covariance* of x and y , which can be positive or negative. A related quantity is the *correlation coefficient*,

$$\rho(x, y) = \frac{\text{cov}(x, y)}{\sqrt{\text{var}(x) \text{var}(y)}}, \quad (1.27)$$

which takes values from -1 to 1 . Notice that $\text{cov}(x, x) = \text{var}(x)$. When the variables x and y are independent, *i.e.*, when $p(x, y) = p_x(x) p_y(y)$, we have

$$\text{cov}(x, y) = 0 \quad \text{and} \quad \text{var}(x + y) = \text{var}(x) + \text{var}(y). \quad (1.28)$$

Moreover, for independent variables,

$$\text{var}\{a_1 x + a_2 y\} = a_1^2 \text{var}(x) + a_2^2 \text{var}(y). \quad (1.29)$$

1.2 Random-sampling methods

The first component of a Monte Carlo calculation is the numerical sampling of random variables with specified PDFs. In this Section we describe different techniques to generate random values of a variable x distributed in the interval $(x_{\min}, x_{\max})$ according to a given PDF $p(x)$. We concentrate on the simple case of single-variable distributions, because random sampling from multivariate distributions can always be reduced to single-variable sampling (see below). A more detailed description of sampling methods can be found in the textbooks of Rubinstein (1981) and Kalos and Whitlock (1986).

1.2.1 Random-number generator

In general, random-sampling algorithms are based on the use of random numbers ξ uniformly distributed in the interval $(0, 1)$. These random numbers can be easily generated on the computer (see, *e.g.*, Kalos and Whitlock, 1986; James, 1990). Among the “good”

Table 1.1: Fortran 77 random-number generator.

```

C *****
C                               FUNCTION RAND
C *****
C                               FUNCTION RAND(DUMMY)
C
C This is an adapted version of subroutine RANECU written by F. James
C (Comput. Phys. Commun. 60 (1990) 329-344), which has been modified to
C give a single random number at each call.
C
C The 'seeds' ISEED1 and ISEED2 must be initialised in the main program
C and transferred through the named common block /RSEED/.
C
C      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER*4 (I-N)
C      PARAMETER (USCALE=1.0D0/2.147483563D9)
C      COMMON/RSEED/ISEED1,ISEED2
C
C      I1=ISEED1/53668
C      ISEED1=40014*(ISEED1-I1*53668)-I1*12211
C      IF(ISEED1.LT.0) ISEED1=ISEED1+2147483563
C
C      I2=ISEED2/52774
C      ISEED2=40692*(ISEED2-I2*52774)-I2*3791
C      IF(ISEED2.LT.0) ISEED2=ISEED2+2147483399
C
C      IZ=ISEED1-ISEED2
C      IF(IZ.LT.1) IZ=IZ+2147483562
C      RAND=IZ*USCALE
C
C      RETURN
C      END

```

random-number generators currently available, the simplest ones are the so-called multiplicative congruential generators (Press and Teukolsky, 1992). A popular example of this kind of generator is the following,

$$R_n = 7^5 R_{n-1} \pmod{2^{31} - 1}, \quad \xi_n = R_n / (2^{31} - 1), \quad (1.30)$$

which produces a sequence of random numbers ξ_n uniformly distributed in (0,1) from a given “seed” R_0 ($< 2^{31} - 1$). Actually, the generated sequence is not truly random, because it is obtained from a deterministic algorithm (the term “pseudo-random” would be more appropriate), but it is very unlikely that the subtle correlations between the values in the sequence have an appreciable effect on the simulation results. The generator (1.30) is known to have good random properties (Press and Teukolsky, 1992). However, the sequence is periodic, with a period of the order of 10^9 . With present-day computational facilities, this value is not large enough to prevent re-initiation in a single simulation run. An excellent critical review of random-number generators has been published by James (1990), where he recommends using algorithms that are more sophisticated than simple congruential ones. The generator implemented in the Fortran 77 function `RAND` (Table 1.1) is due to L’Ecuyer (1988); it produces 32-bit floating-point

numbers uniformly distributed in the *open* interval between zero and one. Its period is of the order of 10^{18} , which is virtually inexhaustible in practical simulations.

1.2.2 Inverse-transform method

The cumulative distribution function of $p(x)$, Eq. (1.6), is a non-decreasing function of x and, therefore, it has an inverse function $\mathcal{P}^{-1}(\xi)$. The transformation $\xi = \mathcal{P}(x)$ defines a new random variable that takes values in the interval $(0,1)$, see Fig. 1.1. Owing to the correspondence between x and ξ values, the PDF of ξ , $p_\xi(\xi)$, and that of x , $p(x)$, are related by $p_\xi(\xi) d\xi = p(x) dx$. Hence,

$$p_\xi(\xi) = p(x) \left(\frac{d\xi}{dx} \right)^{-1} = p(x) \left(\frac{d\mathcal{P}(x)}{dx} \right)^{-1} = 1, \quad (1.31)$$

that is, ξ is distributed uniformly in the interval $(0,1)$.

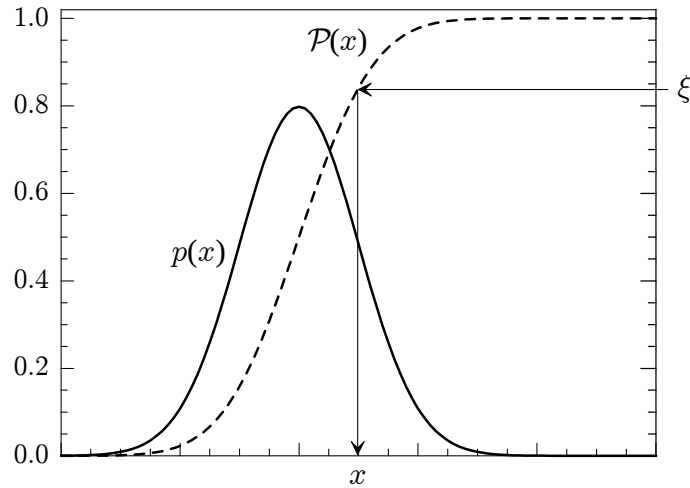


Figure 1.1: Random sampling from a distribution $p(x)$ using the inverse-transform method.

Now it is clear that if ξ is a random number, the variable x defined by $x = \mathcal{P}^{-1}(\xi)$ is randomly distributed in the interval $(x_{\min}, x_{\max})$ with PDF $p(x)$ (see Fig. 1.1). This provides a practical method for generating random values of x using a generator of random numbers uniformly distributed in $(0,1)$. The randomness of x is guaranteed by that of ξ . Notice that x is the (unique) root of the equation

$$\xi = \int_{x_{\min}}^x p(x') dx', \quad (1.32)$$

which will be referred to as the *sampling equation* of the variable x . This procedure for random sampling is known as the *inverse-transform method*; it is particularly adequate for PDFs $p(x)$ given by simple analytical expressions such that the sampling equation (1.32) can be solved analytically.

The inverse-transform method can also be efficiently used for random sampling from continuous distributions $p(x)$ that are given in numerical form, or that are too complicated to be sampled analytically. To apply this method, the cumulative distribution function $\mathcal{P}(x)$ has to be evaluated at the points x_i of a certain grid. The sampling equation $\mathcal{P}(x) = \xi$ can then be solved by inverse interpolation, *i.e.*, by interpolating in the table (ξ_i, x_i) , where $\xi_i \equiv \mathcal{P}(x_i)$ (ξ is regarded as the independent variable). Care must be exercised to make sure that the numerical integration and interpolation do not introduce significant errors. An adaptive algorithm for random sampling from arbitrary continuous distributions is described in Section 1.2.4.

1.2.2.1 Examples

- **Uniform distribution.** The uniform distribution in the interval (a, b) is given by

$$p(x) = U_{a,b}(x) = \frac{1}{b-a}.$$

The sampling equation (1.32) for this PDF reads

$$\xi = \frac{x-a}{b-a}, \quad (1.33)$$

which leads to the well-known sampling formula

$$x = a + \xi(b-a). \quad (1.34)$$

- **Exponential distribution.** The exponential distribution,

$$p(s) = \frac{1}{\lambda} \exp(-s/\lambda), \quad s \geq 0, \quad (1.35)$$

is the PDF of the free path s of a particle between interaction events (see Section 1.4.2). The parameter λ represents the mean free path. In this case, the sampling equation (1.32) is easily solved to give the sampling formula

$$s = -\lambda \ln(1 - \xi) \stackrel{=}{=} -\lambda \ln \xi. \quad (1.36)$$

The last equality ($\stackrel{=}{=}$) indicates that the two sampling formulas are equivalent, in the sense that they generate random values from the exponential distribution. Their equivalence follows from the fact that $1 - \xi$ is, like ξ , a random number uniformly distributed in $(0,1)$. The last formula avoids one subtraction and is, therefore, somewhat faster.

- **Wentzel distribution.** The Wentzel distribution is defined by

$$p(x) = \frac{A(A+1)}{(A+x)^2}, \quad 0 \leq x \leq 1, \quad A > 0. \quad (1.37)$$

This distribution describes the scattering of charged particles by an exponentially-screened Coulomb (or Yukawa) potential within the first Born approximation (Wentzel, 1927). The sampling equation (1.32) for this PDF reads

$$\xi = A(A+1) \left[\frac{1}{A} - \frac{1}{A+x} \right], \quad (1.38)$$

and yields the following sampling formula,

$$x = \frac{A\xi}{A+1-\xi}. \quad (1.39)$$

1.2.3 Discrete distributions

The inverse-transform method can also be applied to discrete distributions. Consider that the random variable x can take the discrete values $x = 1, \dots, N$ with point probabilities $p_1, \dots, p_N$, respectively. The corresponding PDF can be expressed as

$$p(x) = \sum_{i=1}^N p_i \delta(x-i), \quad (1.40)$$

where $\delta(x)$ is the Dirac distribution. Here $p(x)$ is assumed to be defined for x in an interval $(x_{\min}, x_{\max})$ with $x_{\min} < 1$ and $x_{\max} > N$. The corresponding cumulative distribution function is

$$\mathcal{P}(x) = \begin{cases} 0 & \text{if } x < 1, \\ \sum_{i=1}^{[x]} p_i & \text{if } 1 \leq x \leq N, \\ 1 & \text{if } x > N, \end{cases} \quad (1.41)$$

where $[x]$ stands for the integer part of x . Then, Eq. (1.32) leads to the sampling formula

$$\begin{aligned} x &= 1 && \text{if } \xi \leq p_1 \\ &= 2 && \text{if } p_1 < \xi \leq p_1 + p_2 \\ &\vdots && \\ &= j && \text{if } \sum_{i=1}^{j-1} p_i < \xi \leq \sum_{i=1}^j p_i \\ &\vdots && \end{aligned} \quad (1.42)$$

We can define the quantities

$$\mathcal{P}_1 = 0, \quad \mathcal{P}_2 = p_1, \quad \mathcal{P}_3 = p_1 + p_2, \quad \dots, \quad \mathcal{P}_{N+1} = \sum_{i=1}^N p_i = 1. \quad (1.43)$$

To sample x we generate a random number ξ and set x equal to the index i such that

$$\mathcal{P}_i < \xi \leq \mathcal{P}_{i+1}. \quad (1.44)$$

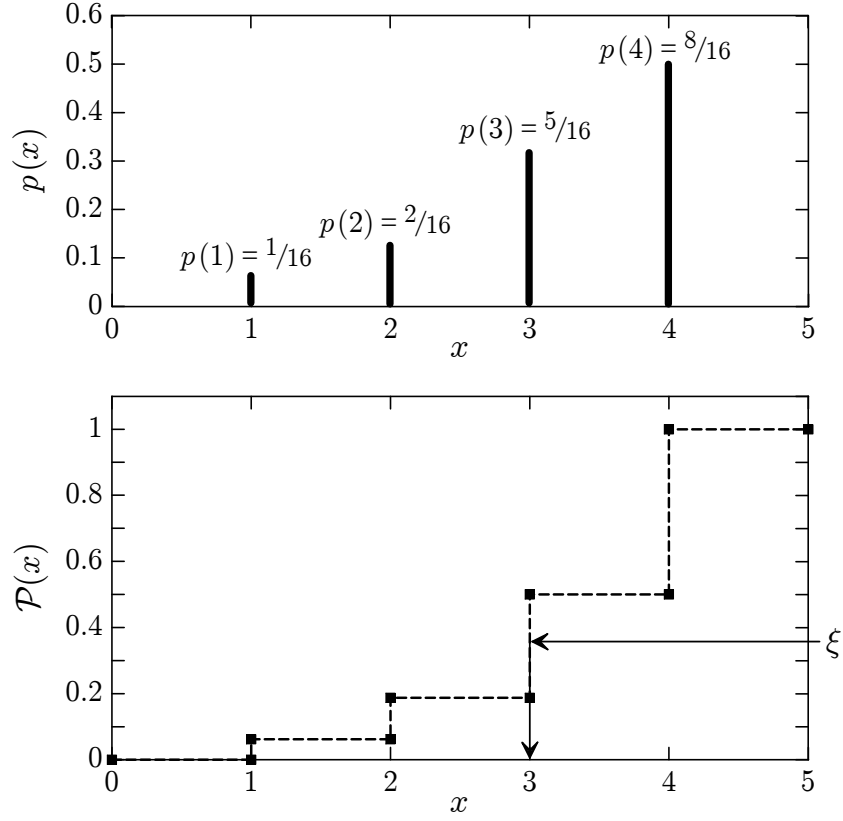


Figure 1.2: Random sampling from a discrete PDF using the inverse-transform method. The random variable can take the values $i = 1, 2, 3$ and 4 with relative probabilities $1, 2, 5$ and 8 , respectively.

The method is illustrated in Fig. 1.2 for a discrete distribution with $N = 4$ values. Notice the similarity with Fig. 1.1.

If the number N of x -values is large and the index i is searched sequentially, the sampling algorithm given by Eq. (1.44) may be quite slow because of the large number of comparisons needed to determine the sampled value. The easiest method to reduce the number of comparisons is to use binary search instead of sequential search. The algorithm for binary search, for a given value of ξ , proceeds as follows:

- (i) Set $i = 1$ and $j = N + 1$.
- (ii) Set $k = [(i + j)/2]$.
- (iii) If $\mathcal{P}_k < \xi$, set $i = k$; otherwise set $j = k$.
- (iv) If $j - i > 1$, go to step (ii).
- (v) Deliver i .

When $2^n < N \leq 2^{n+1}$, i is obtained after $n+1$ comparisons. This number of comparisons is evidently much less than the number required when using purely sequential search. Although the algorithm uses multiple divisions of integer numbers by 2 , this operation is relatively fast (much faster than the division of real numbers).

1.2.3.1 Walker's aliasing method

Walker (1977) described an optimal sampling method for discrete distributions, which yields the sampled value with only one comparison. The idea underlying Walker's method can be easily understood by resorting to graphical arguments (Salvat, 1987). To this end, let us represent the PDF (1.40) as a histogram constructed with N bars of width $1/N$ and heights Np_i (see Fig. 1.3). Now, the histogram bars can be cut off at convenient heights and the resulting pieces can be arranged to fill up the square of unit side in such a way that each vertical line crosses, at most, two different pieces. This arrangement can be performed systematically by selecting the lowest and the highest bars in the histogram, say the ℓ -th and the j -th, respectively, and by cutting the highest bar off to complete the lowest one, which is subsequently kept unaltered. In order to keep track of the performed transformation, we label the moved piece with the "alias" value $K_\ell = j$, giving its original position in the histogram, and we introduce the "cutoff" value F_ℓ defined as the height of the lower piece in the ℓ -th bar of the resulting square. This lower piece keeps the label ℓ . Evidently, iteration of this process eventually leads to the complete square (after, at most, $N - 1$ steps). Notice that the point probabilities p_i can be reconstructed from the alias and cutoff values. We have

$$Np_i = F_i + \sum_{j \neq i} (1 - F_j) \delta(i, K_j), \quad (1.45)$$

where $\delta(i, j)$ denotes the Kronecker delta ($= 1$ if $i = j$, and $= 0$ otherwise). Walker's method for random sampling of x proceeds as follows: We sample two independent random numbers, say ξ_1 and ξ_2 , and define the random point (ξ_1, ξ_2) , which is uniformly distributed in the square. If (ξ_1, ξ_2) lies over a piece labelled with the index i , we take $x = i$ as the selected value. Obviously, the probability of obtaining i as a result of the sampling equals the fractional area of the pieces labelled with i , which coincides with p_i .

As formulated above, Walker's algorithm requires the generation of two random numbers for each sampled value of x . With the aid of the following trick, the x -value can be generated from a single random number. Continuing with our graphical picture, assume that the N bars in the square are aligned consecutively to form a segment of length N (bottom of Fig. 1.3). To sample x , we can generate a single random value ξN , which is uniformly distributed in $(0, N)$ and determines one of the segment pieces. The result of the sampling is the label of the selected piece. Explicitly, the sampling algorithm proceeds as follows:

- (i) Generate a random number ξ and set $R = \xi N + 1$.
- (ii) Set $i = [R]$ and $r = R - i$.
- (iii) If $r > F_i$, deliver $x = K_i$.
- (iv) Deliver $x = i$.

We see that the sampling of x involves only the generation of a random number and one comparison (irrespective of the number N of possible outcomes). The price we

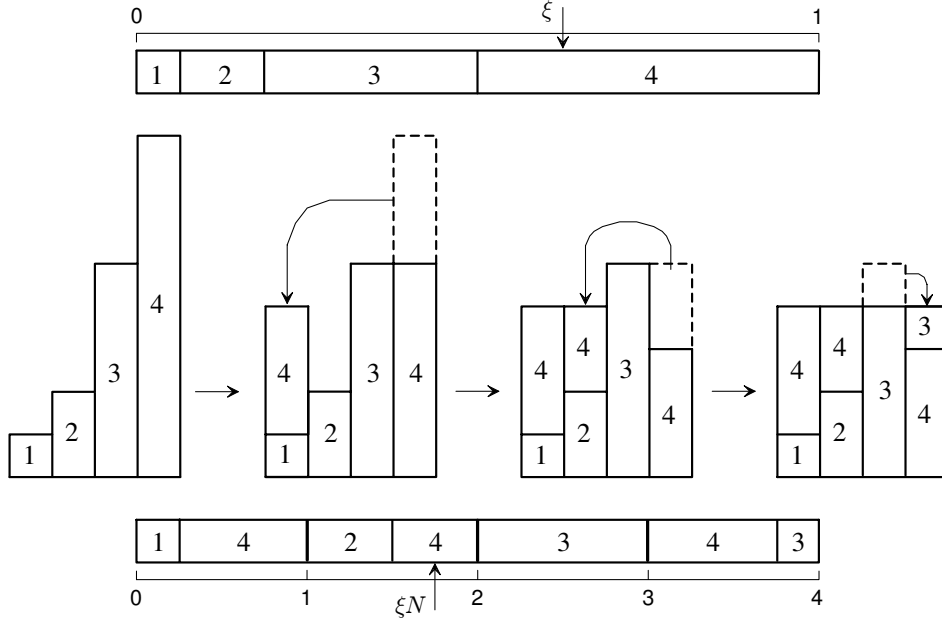


Figure 1.3: Graphical representation of the inverse-transform method (top) and Walker's aliasing method (bottom) for random sampling from the discrete distribution shown in Fig. 1.2.

pay for this simplification reduces to doubling the number of memory locations that are needed: the two arrays K_i and F_i are used instead of the single array p_i (or $\mathcal{P}_i$). Unfortunately, the calculation of alias and cutoff values is fairly involved and this limits the applicability of Walker's algorithm to distributions that remain constant during the course of the simulation.

1.2.4 Numerical inverse transform for continuous PDFs

We can now formulate a general numerical algorithm for random sampling from continuous distributions using the inverse-transform method. Let us consider a random variable x that can take values within a (finite) interval $[x_{\min}, x_{\max}]$ with a given PDF $p(x)$. We assume that the function $p(x)$ is continuous and that it can be calculated accurately for any value of x in the interval $[x_{\min}, x_{\max}]$. In practice, numerical distributions are defined by a table of values, from which $p(x)$ has to be obtained by interpolation. We consider that the tabulated values are exact and spaced closely enough to ensure that interpolation errors are negligible. In PENELOPE we frequently use cubic spline log-log interpolation (see Section B.1), which has the advantage of yielding an interpolated PDF that is continuous and has continuous first and second derivatives.

Let us assume that the cumulative distribution function $\mathcal{P}(x)$ has been evaluated

numerically for a certain grid of x -values that spans the interval $[x_{\min}, x_{\max}]$,

$$x_1 = x_{\min} < x_2 < \dots < x_{N-1} < x_N = x_{\max}. \quad (1.46)$$

Setting $\xi_i = \mathcal{P}(x_i)$, we get a table of the inverse cumulative distribution function $\mathcal{P}^{-1}(\xi_i) = x_i$ for a grid of ξ -values that spans the interval $[0, 1]$,

$$\xi_1 = 0 < \xi_2 < \dots < \xi_{N-1} < \xi_N = 1. \quad (1.47)$$

In principle, the solution of the sampling equation, $x = \mathcal{P}^{-1}(\xi)$, can be obtained by interpolation in this table. The adopted interpolation scheme must be able to accurately reproduce the first derivative of the function $\mathcal{P}^{-1}(\xi)$,

$$\frac{d\mathcal{P}^{-1}(\xi)}{d\xi} = \left(\frac{d\mathcal{P}(x)}{dx} \right)^{-1} = \frac{1}{p(x)}. \quad (1.48)$$

Notice that this function is very steep in regions where the PDF is small. Linear interpolation of $\mathcal{P}^{-1}(\xi)$ is in general too crude, because it is equivalent to approximating $p(x)$ by a stepwise distribution. It is more expedient to use a *rational interpolation* scheme of the type²

$$\tilde{\mathcal{P}}^{-1}(\xi) = x_i + \frac{(1 + a_i + b_i)\eta}{1 + a_i\eta + b_i\eta^2} (x_{i+1} - x_i) \quad \text{if } \xi_i \leq \xi < \xi_{i+1}, \quad (1.49)$$

where

$$\eta \equiv (\xi - \xi_i)/(\xi_{i+1} - \xi_i) \quad (1.50)$$

and a_i and b_i are parameters. Notice that $\tilde{\mathcal{P}}^{-1}(\xi_i) = x_i$ and $\tilde{\mathcal{P}}^{-1}(\xi_{i+1}) = x_{i+1}$, irrespective of the values of a_i and b_i . Moreover,

$$\frac{d\tilde{\mathcal{P}}^{-1}(\xi)}{d\xi} = \frac{(1 + a_i + b_i)(1 - b_i\eta^2)}{(1 + a_i\eta + b_i\eta^2)^2} \frac{x_{i+1} - x_i}{\xi_{i+1} - \xi_i}. \quad (1.51)$$

The parameters a_i and b_i are determined by requiring that

$$\left[\frac{d\tilde{\mathcal{P}}^{-1}(\xi)}{d\xi} \right]_{\xi=\xi_i} = \frac{1}{p(x_i)} \quad \text{and} \quad \left[\frac{d\tilde{\mathcal{P}}^{-1}(\xi)}{d\xi} \right]_{\xi=\xi_{i+1}} = \frac{1}{p(x_{i+1})}. \quad (1.52)$$

This implies

$$1 + a_i + b_i = \frac{\xi_{i+1} - \xi_i}{x_{i+1} - x_i} \frac{1}{p(x_i)} \quad \text{and} \quad \frac{1 - b_i}{1 + a_i + b_i} = \frac{\xi_{i+1} - \xi_i}{x_{i+1} - x_i} \frac{1}{p(x_{i+1})},$$

and it follows that

$$b_i = 1 - \left(\frac{\xi_{i+1} - \xi_i}{x_{i+1} - x_i} \right)^2 \frac{1}{p(x_{i+1}) p(x_i)}, \quad (1.53a)$$

$$a_i = \frac{\xi_{i+1} - \xi_i}{x_{i+1} - x_i} \frac{1}{p(x_i)} - b_i - 1. \quad (1.53b)$$

²We denote by $\tilde{\mathcal{P}}^{-1}(\xi)$ the function obtained by interpolation of the tabulated values $\mathcal{P}^{-1}(\xi_i)$.

Thus, a_i and b_i are determined by the values of the PDF $p(x)$ at the grid points x_i . Once these parameters have been calculated, the sampling formula

$$x = x_i + \frac{(1 + a_i + b_i)\eta}{1 + a_i\eta + b_i\eta^2} (x_{i+1} - x_i) \quad \text{if } \xi_i \leq \xi < \xi_{i+1} \quad (1.54)$$

gives random values of x that are *exactly* distributed according to the PDF

$$\tilde{p}(x) = \left(\frac{d\tilde{\mathcal{P}}^{-1}(\xi)}{d\xi} \right)^{-1} = \frac{(1 + a_i\eta + b_i\eta^2)^2}{(1 + a_i + b_i)(1 - b_i\eta^2)} \frac{\xi_{i+1} - \xi_i}{x_{i+1} - x_i} \quad \text{if } x_i \leq x < x_{i+1}. \quad (1.55)$$

From Eq. (1.53a) we see that b_i is always less than unity and, therefore, the denominator in expression (1.55) is positive, *i.e.*, $\tilde{p}(x)$ is positive, as required for a proper PDF. To calculate $\tilde{p}(x)$ for a given x , we have to determine the value of η by solving Eq. (1.54). The root that satisfies the conditions $\eta = 0$ for $x = x_i$ and $\eta = 1$ for $x = x_{i+1}$ is

$$\eta = \frac{1 + a_i + b_i - a_i\tau}{2b_i\tau} \left[1 - \sqrt{1 - \frac{4b_i\tau^2}{(1 + a_i + b_i - a_i\tau)^2}} \right], \quad \tau \equiv \frac{x - x_i}{x_{i+1} - x_i}. \quad (1.56)$$

It is worth noting that the distributions $p(x)$ and $\tilde{p}(x)$ not only coincide at the grid points x_i , but also their integrals over the interval (x_i, x_{i+1}) are equal [*i.e.*, $\tilde{p}(x)$ gives the correct probability of finding x in each interval]. The PDF (1.55) turns out to be fairly flexible and can approximate smooth PDFs over relatively wide intervals to good accuracy. Moreover, because formula (1.54) involves only a few arithmetic operations, random sampling will be faster than with alternative interpolation schemes that lead to sampling formulas involving transcendental functions.

1.2.4.1 Determining the interpolation grid

The key to ensure accuracy of the sampling is to set a suitable grid of x -values, x_i ($i = 1, \dots, N$), such that errors introduced by the rational interpolation (1.55) are negligible (say, of the order of 0.01% or less). A simple, and effective strategy for defining the x -grid is the following. We start with a uniform grid of ~ 10 equally spaced x -values. The cumulative distribution function at these grid points, $\mathcal{P}(x_i) = \xi_i$, is evaluated numerically (see below). After calculating the parameters a_i and b_i of the interpolating PDF, Eq. (1.55), the interpolation “error” in the i -th interval (x_i, x_{i+1}) is defined as

$$\epsilon_i = \int_{x_i}^{x_{i+1}} |p(x) - \tilde{p}(x)| dx, \quad (1.57)$$

where the integral is evaluated numerically. To reduce the interpolation error efficiently, new points x_i are added where the error is larger. The position of each new point is selected at the midpoint of the interval j with the largest ϵ value. After inserting each new point, the interpolation parameters a_i and b_i , for the two new intervals (the two

halves of the initial j -th interval) are evaluated, as well as the corresponding interpolation errors ϵ_i , Eq. (1.57). The process is iterated until the last, N -th, grid point has been set. Obviously, to reduce the interpolation error we only need to increase the number N of grid points.

Using this strategy to set new grid points, the algorithm is self-adaptive: grid points automatically accumulate in intervals where the PDF and its approximation (1.55) differ most. If the number N of grid points is large enough, the functions $p(x)$ and $\tilde{p}(x)$ vary smoothly within each grid interval (x_i, x_{i+1}) . Then, integrals over grid intervals can be calculated accurately by using simple quadrature formulas. In our implementation of the sampling algorithm, we use the extended Simpson rule with 51 equally-spaced points,

$$\int_{x_i}^{x_{i+1}} f(x) dx = \frac{h}{3} \left[f_0 + 4 \left(f_1 + f_3 + \cdots + f_{49} \right) + 2 \left(f_2 + f_4 + \cdots + f_{48} \right) + f_{50} \right] - \frac{25}{90} h^5 f^{(iv)}(x^*), \quad (1.58)$$

where $h = (x_{i+1} - x_i)/50$, $f_k = f(x_i + kh)$, and $f^{(iv)}(x^*)$ is the fourth derivative of the function $f(x)$ at an unknown point x^* in the interval (x_i, x_{i+1}) .

Figure 1.4 displays the rational interpolation, Eq. (1.55), of the analytical PDF defined in the inset and limited to the interval $[0,5]$. The crosses indicate the points of the grid for $N = 32$. Agreement between the interpolating PDF (dashed curve, not visible) and the original distribution is striking. The rational interpolation is seen to very closely reproduce the curvature of the original distribution, even when the grid points are quite spaced. The lower plot in Fig. 1.4 represents the local interpolation error ϵ_i in each interval of the grid (as a stepwise function for visual aid); the maximum error in this case is 3.2×10^{-4} . For a denser grid with $N = 128$ values, the maximum error decreases to 8.3×10^{-7} .

1.2.4.2 Sampling algorithm

After determining the interpolation grid and the parameters of the rational interpolation,

$$x_i, \xi_i = \mathcal{P}(x_i), a_i, b_i \quad (i = 1, \dots, N), \quad (1.59)$$

the sampling from the distribution (1.55) can be performed exactly by using the following algorithm:

- (i) Generate a random number ξ .
- (ii) Find the interval i that contains ξ ,

$$\xi_i \leq \xi < \xi_{i+1}, \quad (1.60)$$

using the binary-search method.

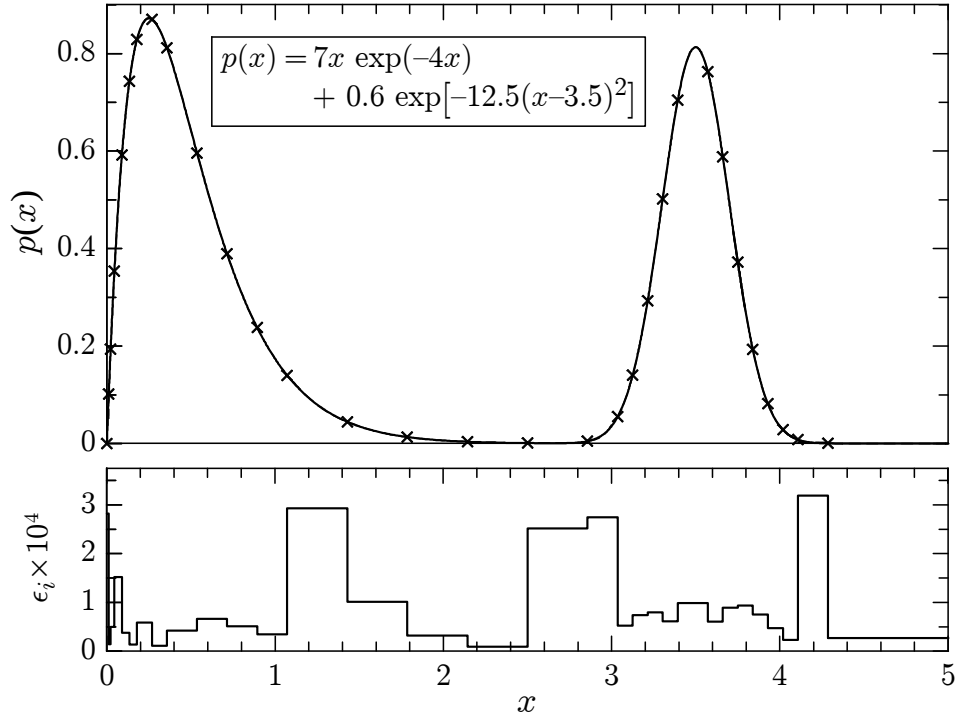


Figure 1.4: Rational interpolation of the continuous PDF defined by the analytical expression indicated in the inset and restricted to the interval $[0,5]$. The crosses are grid points determined as described in the text with $N = 32$. The rational interpolating function given by Eq. (1.55) is represented by a dashed curve, which is not visible on this scale. The lower plot displays the interpolation error ϵ_i .

(iii) Set $\nu \equiv \xi - \xi_i$, $\Delta_i \equiv \xi_{i+1} - \xi_i$.

(iv) Deliver

$$x = x_i + \frac{(1 + a_i + b_i)\Delta_i\nu}{\Delta_i^2 + a_i\Delta_i\nu + b_i\nu^2} (x_{i+1} - x_i). \quad (1.61)$$

The sampling speed decreases (slowly) when the number N of grid points increases, due to the increasing number of comparisons needed in step (ii). This loss of speed can be readily avoided by using Walker's aliasing (Section 1.2.3.1) to sample the “active” interval. Walker's method requires only a single comparison and, hence, the algorithm becomes optimal, at the expense of some additional memory storage. A drawback of Walker's method is that the sampled value x is not a continuous function of the random number ξ . This feature impedes the use of the method for sampling the variable in a restricted domain, as needed, *e.g.*, in mixed simulations of electron transport. A less sophisticated procedure to reduce the number of comparisons, which is free from this drawback (the generated x values increase monotonically with ξ), consists of providing pre-calculated limits (*e.g.*, tabulated as functions of the integer variable $k = \lfloor \xi N \rfloor$) for the range of interval indices i that needs to be explored. In practical calculations, this

procedure is only slightly slower than Walker's aliasing. The present sampling algorithm, either with Walker's aliasing or with pre-calculated index intervals, will be referred to as the RITA (Rational Inverse Transform with Aliasing) algorithm. In PENELOPE, RITA is used to simulate elastic collisions of electrons and positrons (Section 3.1), and coherent (Rayleigh) scattering of photons (Section 2.1).

1.2.5 Rejection methods

The inverse-transform method for random sampling is based on a one-to-one correspondence between x and ξ values, which is expressed in terms of a single-valued function. There is another kind of sampling method, due to von Neumann, that consists of sampling a random variable from a certain distribution [different to $p(x)$] and subjecting it to a random test to determine whether it will be accepted for use or rejected. These rejection methods lead to very general techniques for sampling from any PDF.

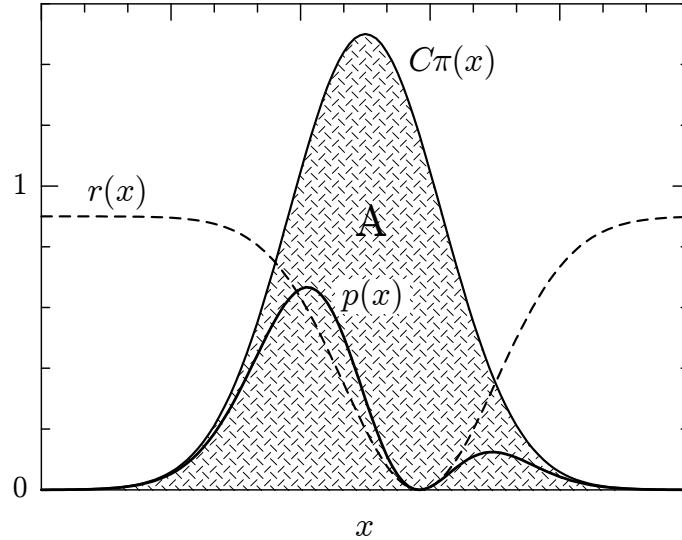


Figure 1.5: Random sampling from a distribution $p(x)$ using a rejection method.

The rejection algorithms can be understood in terms of simple graphical arguments (Fig. 1.5). Consider that, by means of the inverse-transform method or any other available sampling method, random values of x are generated from a PDF $\pi(x)$. For each sampled value of x we sample a random value y uniformly distributed in the interval $(0, C\pi(x))$, where C is a positive constant. Evidently, the points (x, y) , generated in this way, are uniformly distributed in the region A of the plane limited by the x -axis ($y = 0$) and the curve $y = C\pi(x)$. Conversely, if (by some means) we generate random points (x, y) uniformly distributed in A, their x -coordinate is a random variable distributed according to $\pi(x)$ (irrespective of the value of C). Now, consider that the distribution $\pi(x)$ is such that $C\pi(x) \geq p(x)$ for some $C > 0$ and that we generate random points (x, y) uniformly distributed in the region A as described above. If we reject the points

with $y > p(x)$, the accepted ones (with $y \leq p(x)$) are uniformly distributed in the region between the x -axis and the curve $y = p(x)$ and hence, their x -coordinate is distributed according to $p(x)$.

A rejection method is thus completely specified by representing the PDF $p(x)$ as

$$p(x) = C\pi(x)r(x), \quad (1.62)$$

where $\pi(x)$ is a PDF that can be easily sampled, *e.g.*, by the inverse-transform method, C is a positive constant and the function $r(x)$ satisfies the conditions $0 \leq r(x) \leq 1$. The rejection algorithm for sampling from $p(x)$ proceeds as follows:

- (i) Generate a random value x from $\pi(x)$.
- (ii) Generate a random number ξ .
- (iii) If $\xi > r(x)$, go to step (i).
- (iv) Deliver x .

From the geometrical arguments given above, it is clear that the algorithm does yield x values distributed according to $p(x)$. The following is a more formal proof: Step (i) produces x -values in the interval $(x, x + dx)$ with probability $\pi(x) dx$, these values are accepted with probability $r(x) = p(x)/[C\pi(x)]$ and, therefore, (apart from a normalisation constant) the probability of delivering a value in $(x, x + dx)$ is equal to $p(x) dx$ as required. It is important to realise that, as regards Monte Carlo, the normalisation of the simulated PDF is guaranteed by the mere fact that the algorithm delivers some value of x .

The efficiency of the algorithm, *i.e.*, the probability of accepting a generated x -value, is

$$\epsilon = \int_a^b r(x)\pi(x) dx = \frac{1}{C}. \quad (1.63)$$

Graphically, the efficiency equals the ratio of the areas under the curves $y = p(x)$ and $y = C\pi(x)$, which are 1 and C , respectively. For a given $\pi(x)$, since $r(x) \leq 1$, the constant C must satisfy the condition $C\pi(x) \geq p(x)$ for all x . The minimum value of C , with the requirement that $C\pi(x) = p(x)$ for some x , gives the optimum efficiency.

The PDF $\pi(x)$ in Eq. (1.62) should be selected in such a way that the resulting sampling algorithm is as fast as possible. In particular, random sampling from $\pi(x)$ must be performed rapidly, by the inverse-transform method or by the composition method (see below). High efficiency is also desirable, but not decisive. One hundred percent efficiency is obtained only with $\pi(x) = p(x)$ (however, random sampling from this PDF is just the problem we want to solve); any other PDF gives a lower efficiency. The usefulness of the rejection method lies in the fact that a certain loss of efficiency can be largely compensated with the ease of sampling x from $\pi(x)$ instead of $p(x)$. A disadvantage of this method is that it requires the generation of several random numbers ξ to sample each x -value.

1.2.6 Two-dimensional variables. Composition methods

Let us consider a two-dimensional random variable (x, y) with joint probability distribution function $p(x, y)$. Introducing the marginal PDF $q(y)$ and the conditional PDF $p(x|y)$ [see Eqs. (1.18) and (1.20)],

$$q(y) \equiv \int p(x, y) dx, \quad p(x|y) = \frac{p(x, y)}{q(y)},$$

the bivariate distribution can be expressed as

$$p(x, y) = q(y) p(x|y). \quad (1.64)$$

It is now evident that to generate random points (x, y) from $p(x, y)$ we can first sample y from $q(y)$ and then x from $p(x|y)$. Hence, two-dimensional random variables can be generated by using single-variable sampling methods. This is also true for multivariate distributions, because an n -dimensional PDF can always be expressed as the product of a single-variable marginal distribution and an $(n - 1)$ -dimensional conditional PDF.

From the definition of the marginal PDF of x ,

$$q(x) \equiv \int p(x, y) dy = \int q(y) p(x|y) dy, \quad (1.65)$$

it is clear that if we sample y from $q(y)$ and, then, x from $p(x|y)$, the generated values of x are distributed according to $q(x)$. This idea is the basis of *composition* methods, which are applicable when $p(x)$, the distribution to be simulated, is a probability mixture of several PDFs. More specifically, we consider that $p(x)$ can be expressed as

$$p(x) = \int w(y) p_y(x) dy, \quad (1.66)$$

where $w(y)$ is a continuous distribution and $p_y(x)$ is a family of one-parameter PDFs, where y is the parameter identifying a unique distribution. Notice that if the parameter y only took integer values $y = i$ with point probabilities w_i , we would write

$$p(x) = \sum_i w_i p_i(x). \quad (1.67)$$

The composition method for random sampling from the PDF $p(x)$ is as follows. First, a value of y (or i) is drawn from the PDF $w(y)$ and then x is sampled from the PDF $p_y(x)$ for that chosen y .

This technique may be applied to generate random values from complex distributions obtained by combining simpler distributions that are themselves easily generated, *e.g.*, by the inverse-transform method or by rejection methods.

Devising fast, exact methods for random sampling from a given PDF is an interesting technical challenge. The ultimate criterion for the quality of a sampling algorithm is its

speed in actual simulations: the best algorithm is the fastest. However, programming simplicity and elegance may justify the use of slower algorithms. For simple analytical distributions that have an analytical inverse cumulative distribution function, the inverse-transform method is usually satisfactory. This is the case for a few elementary distributions (*e.g.*, the uniform and exponential distributions considered above). The inverse-transform method is also adequate for discrete distributions, particularly when combined with Walker's aliasing. The adaptive sampling algorithm RITA, described in Section 1.2.4, provides a practical method for sampling from continuous single-variate PDFs, defined either analytically or in numerical form; this algorithm is fast and quite accurate, but it is not exact. By combining the inverse-transform, rejection and composition methods we can devise exact sampling algorithms for virtually any (single- or multivariate) PDF.

1.2.6.1 Examples

• **Sampling from the normal distribution.** Frequently we need to generate random values from the normal (or Gaussian) distribution

$$p_G(x) = \frac{1}{\sqrt{2\pi}} \exp(-x^2/2). \quad (1.68)$$

Since the cumulative distribution function cannot be inverted analytically, the inverse-transform method is not appropriate. The easiest (but not the fastest) method to sample from the normal distribution consists of generating two independent random variables at a time, as follows. Let x_1 and x_2 be two independent normal variables. They determine a random point in the plane with PDF

$$p_{2G}(x_1, x_2) = p_G(x_1) p_G(x_2) = \frac{1}{2\pi} \exp[-(x_1^2 + x_2^2)/2].$$

Introducing the polar coordinates r and ϕ ,

$$x_1 = r \cos \phi, \quad x_2 = r \sin \phi,$$

the PDF can be expressed as

$$p_{2G}(x_1, x_2) dx_1 dx_2 = \frac{1}{2\pi} \exp(-r^2/2) r dr d\phi = [\exp(-r^2/2) r dr] \left[\frac{1}{2\pi} d\phi \right].$$

We see that r and ϕ are independent random variables. The angle ϕ is distributed uniformly on $(0, 2\pi)$ and can be sampled as $\phi = 2\pi\xi$. The PDF of r is $\exp(-r^2/2) r$ and the corresponding cumulative distribution function is $\mathcal{P}(r) = 1 - \exp(-r^2/2)$. Therefore, r can be generated by the inverse-transform method as

$$r = \sqrt{-2 \ln(1 - \xi)} \doteq \sqrt{-2 \ln \xi}.$$

The two independent normal random variables are given by

$$\begin{aligned}x_1 &= \sqrt{-2 \ln \xi_1} \cos(2\pi \xi_2), \\x_2 &= \sqrt{-2 \ln \xi_1} \sin(2\pi \xi_2),\end{aligned}\tag{1.69}$$

where ξ_1 and ξ_2 are two independent random numbers. This procedure is known as the Box-Müller method. It has the advantages of being exact and easy to program (it can be coded as a single Fortran statement).

The mean and variance of the normal variable are $\langle x \rangle = 0$ and $\text{var}(x) = 1$. The linear transformation

$$X = m + \sigma x \quad (\sigma > 0)\tag{1.70}$$

defines a new random variable. From the properties (1.14) and (1.29), we have

$$\langle X \rangle = m \quad \text{and} \quad \text{var}(X) = \sigma^2.\tag{1.71}$$

The PDF of X is

$$p(X) = p_G(x) \frac{dx}{dX} = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(X-m)^2}{2\sigma^2}\right],\tag{1.72}$$

i.e., X is normally distributed with mean m and variance σ^2 . Hence, to generate X we only have to sample x using the Box-Müller method and apply the transformation (1.70).

• **Uniform distribution on the unit sphere.** In radiation-transport theory, the direction of motion of a particle is described by a unit vector $\hat{\mathbf{d}}$. Given a certain frame of reference, the direction $\hat{\mathbf{d}}$ can be specified by giving either its direction cosines (u, v, w) (*i.e.*, the projections of $\hat{\mathbf{d}}$ on the directions of the coordinate axes) or the polar angle θ and the azimuthal angle ϕ , defined as in Fig. 1.6,

$$\hat{\mathbf{d}} = (u, v, w) = (\sin \theta \cos \phi, \sin \theta \sin \phi, \cos \theta).\tag{1.73}$$

Notice that $\theta \in (0, \pi)$ and $\phi \in (0, 2\pi)$.

A direction vector can be regarded as a point on the surface of the unit sphere. Consider an isotropic source of particles, *i.e.*, such that the initial direction (θ, ϕ) of emitted particles is a random point uniformly distributed on the surface of the sphere. The PDF is

$$p(\theta, \phi) d\theta d\phi = \frac{1}{4\pi} \sin \theta d\theta d\phi = \left[\frac{\sin \theta}{2} d\theta \right] \left[\frac{1}{2\pi} d\phi \right].\tag{1.74}$$

That is, θ and ϕ are independent random variables with PDFs $p_\theta(\theta) = \sin \theta/2$ and $p_\phi(\phi) = 1/(2\pi)$, respectively. Therefore, the initial direction of a particle from an isotropic source can be generated by applying the inverse-transform method to these PDFs,

$$\theta = \arccos(1 - 2\xi_1), \quad \phi = 2\pi\xi_2.\tag{1.75}$$

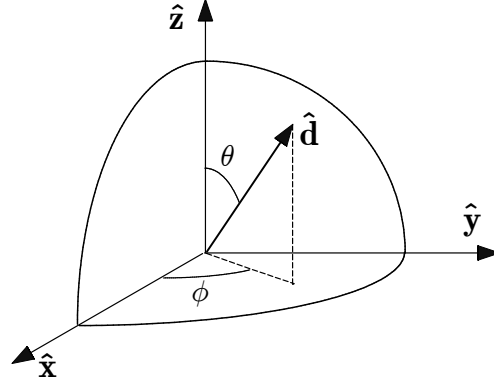


Figure 1.6: Polar and azimuthal angles of a direction vector.

In some cases, it is convenient to replace the polar angle θ by the variable

$$\mu = (1 - \cos \theta)/2, \quad (1.76)$$

which varies from 0 ($\theta = 0$) to 1 ($\theta = \pi$). In the case of an isotropic distribution, the PDF of μ is

$$p_\mu(\mu) = p_\theta(\theta) \left(\frac{d\mu}{d\theta} \right)^{-1} = 1. \quad (1.77)$$

That is, a set of random points (μ, ϕ) uniformly distributed on the rectangle $(0, 1) \times (0, 2\pi)$ corresponds to a set of random directions (θ, ϕ) uniformly distributed on the unit sphere.

1.3 Monte Carlo integration

As pointed out by James (1980), at least in a formal sense, all Monte Carlo calculations are equivalent to integrations. This equivalence permits a formal theoretical foundation for Monte Carlo techniques. An important aspect of simulation is the evaluation of the statistical uncertainties of the calculated quantities. We shall derive the basic formulas by considering the simplest Monte Carlo calculation, namely, the evaluation of a one-dimensional integral. Evidently, the results are also valid for multidimensional integrals.

Consider the integral

$$I = \int_a^b F(x) dx, \quad (1.78)$$

which we recast in the form of an expectation value,

$$I = \int f(x) p(x) dx \equiv \langle f \rangle, \quad (1.79)$$

by introducing an arbitrary PDF $p(x)$ and setting $f(x) = F(x)/p(x)$ [it is assumed that $p(x) > 0$ in (a, b) and $p(x) = 0$ outside this interval]. The Monte Carlo evaluation of the

integral I is very simple: generate a large number N of random points x_i from the PDF $p(x)$ and accumulate the sum of values $f(x_i)$ in a counter. At the end of the calculation the expected value of f is estimated as

$$\bar{f} \equiv \frac{1}{N} \sum_{i=1}^N f(x_i). \quad (1.80)$$

The law of large numbers says that, as N becomes very large,

$$\bar{f} \rightarrow I \quad (\text{in probability}). \quad (1.81)$$

In statistical terminology, this means that $\bar{f}$, the Monte Carlo result, is a *consistent estimator* of the integral (1.78). This is valid for any function $f(x)$ that is finite and piecewise continuous, *i.e.*, with a finite number of discontinuities.

The law of large numbers (1.81) can be restated as

$$\langle f \rangle = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N f(x_i). \quad (1.82)$$

By applying this law to the integral that defines the variance of $f(x)$ [cf. Eq. (1.16)]

$$\text{var}\{f(x)\} = \int f^2(x) p(x) dx - \langle f \rangle^2, \quad (1.83)$$

we obtain

$$\text{var}\{f(x)\} = \lim_{N \rightarrow \infty} \left\{ \frac{1}{N} \sum_{i=1}^N [f(x_i)]^2 - \left[\frac{1}{N} \sum_{i=1}^N f(x_i) \right]^2 \right\}. \quad (1.84)$$

The expression in curly brackets is a consistent estimator of the variance of $f(x)$. In practical simulations, it is advisable (see below) to accumulate the squared function values $[f(x_i)]^2$ in a counter and, at the end of the simulation, estimate $\text{var}\{f(x)\}$ according to Eq. (1.84).

It is clear that different Monte Carlo runs [with different, independent sequences of N random numbers x_i from $p(x)$] will yield different estimates $\bar{f}$. This implies that the outcome of our Monte Carlo code is affected by statistical uncertainties, similar to those found in laboratory experiments, which need to be properly evaluated to determine the “accuracy” of the Monte Carlo result. To this end, we may consider $\bar{f}$ as a random variable, the PDF of which is, in principle, unknown. Its mean and variance are given by

$$\langle \bar{f} \rangle = \left\langle \frac{1}{N} \sum_{i=1}^N f(x_i) \right\rangle = \frac{1}{N} \sum_{i=1}^N \langle f \rangle = \langle f \rangle \quad (1.85)$$

and

$$\text{var}(\bar{f}) = \text{var} \left[\frac{1}{N} \sum_{i=1}^N f(x_i) \right] = \frac{1}{N^2} \sum_{i=1}^N \text{var}\{f(x)\} = \frac{1}{N} \text{var}\{f(x)\}, \quad (1.86)$$

where use has been made of properties of the expectation and variance operators. The standard deviation (or standard error) of $\bar{f}$,

$$\sigma_f \equiv \sqrt{\text{var}(\bar{f})} = \sqrt{\frac{\text{var}\{f(x)\}}{N}}, \quad (1.87)$$

gives a measure of the statistical uncertainty of the Monte Carlo estimate $\bar{f}$. The result (1.87) has an important practical implication: in order to reduce the statistical uncertainty by a factor of 10, we have to increase the sample size N by a factor of 100. Evidently, this sets a limit to the accuracy that can be attained with the available computer power.

We can now invoke the central-limit theorem (see, *e.g.*, James, 1980), which establishes that, in the limit $N \rightarrow \infty$, the PDF of $\bar{f}$ is a normal (Gaussian) distribution with mean $\langle f \rangle$ and standard deviation σ_f ,

$$p(\bar{f}) = \frac{1}{\sigma_f \sqrt{2\pi}} \exp\left(-\frac{(\bar{f} - \langle f \rangle)^2}{2\sigma_f^2}\right). \quad (1.88)$$

It follows that, for sufficiently large values of N , for which the theorem is applicable, the interval $\bar{f} \pm n\sigma_f$ contains the exact value $\langle f \rangle$ with a probability of 68.3% if $n = 1$, 95.4% if $n = 2$ and 99.7% if $n = 3$ (3σ rule).

The central-limit theorem is a very powerful tool, because it predicts that the generated values of $\bar{f}$ follow a specific distribution, but it applies only asymptotically. The minimum number N of sampled values needed to apply the theorem with confidence depends on the problem under consideration. If, in the case of our problem, the third central moment of f ,

$$\mu_3 \equiv \int [f(x) - \langle f \rangle]^3 p(x) dx, \quad (1.89)$$

exists, the theorem is essentially satisfied when

$$|\mu_3| \ll \sigma_f^3 \sqrt{N}. \quad (1.90)$$

In general, it is advisable to study the distribution of the estimator to ascertain the applicability of the central-limit theorem. In most Monte Carlo calculations, however, statistical errors are estimated by simply assuming that the theorem is satisfied, irrespective of the sample size. We shall adopt this practice and report Monte Carlo results in the form $\bar{f} \pm 3\sigma_f$. In simulations of radiation transport, this is empirically validated by the fact that simulated continuous distributions do “look” continuous (*i.e.*, the “error bars” define a smooth band).

Each possible $p(x)$ defines a Monte Carlo algorithm to calculate the integral I , Eq. (1.78). The simplest algorithm (crude Monte Carlo) is obtained by using the uniform distribution $p(x) = 1/(b-a)$. Evidently, $p(x)$ determines not only the density of sampled points x_i , but also the magnitude of the variance $\text{var}\{f(x)\}$, Eq. (1.83),

$$\text{var}\{f(x)\} = \int_a^b p(x) \left[\frac{F(x)}{p(x)} \right]^2 dx - I^2 = \int_a^b F(x) \left[\frac{F(x)}{p(x)} - I \right] dx. \quad (1.91)$$

As a measure of the effectiveness of a Monte Carlo algorithm, it is common to use the efficiency ϵ , which is defined by

$$\epsilon = \left(\frac{\bar{f}}{\sigma_f} \right)^2 \frac{1}{T}, \quad (1.92)$$

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_f^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. When reporting Monte Carlo efficiencies, it is important to make sure that the value of ϵ has stabilised (this usually requires controlling the evolution of ϵ as N increases).

The so-called variance-reduction methods are techniques that aim to optimise the *efficiency* of the simulation through an adequate choice of the PDF $p(x)$. Improving the efficiency of the algorithms is an important, and delicate, part of the art of Monte Carlo simulation. The interested reader is addressed to the specialised bibliography (*e.g.*, Rubinstein, 1981). Although in common use, the term “variance reduction” is somewhat misleading, since a reduction in variance does not necessarily lead to improved efficiency. In certain cases, the variance (1.91) can be reduced to zero. For instance, when $F(x)$ is non-negative, we can consider the distribution $p(x) = F(x)/I$, which evidently gives $\text{var}\{f(x)\} = 0$. This implies that $f(x) = I$ for all points x in (a, b) , *i.e.*, we would obtain the exact value of the integral with just one sampled value! In principle, we can devise a Monte Carlo algorithm, based on an appropriate PDF $p(x)$, which has a variance that is less than that of crude Monte Carlo (*i.e.*, with the uniform distribution). However, if the generation of x -values from $p(x)$ takes a longer time than for the uniform distribution, the “variance-reduced” algorithm may be less efficient than crude Monte Carlo. Hence, one should avoid using PDFs that are too difficult to sample.

1.3.1 Monte Carlo *vs.* numerical quadrature

It is interesting to compare the efficiency of the Monte method with that of conventional numerical quadrature. Let us thus consider the calculation of an integral over the D -dimensional unit cube,

$$I = \int_0^1 du_1 \int_0^1 du_2 \dots \int_0^1 du_D F(u_1, u_2, \dots, u_D), \quad (1.93)$$

where the integrand $F(u_1, u_2, \dots, u_D)$ is assumed to be defined (by an analytic expression or by a numerical procedure) in such a way that it can be calculated *exactly* at any point in the unit cube. This problem is not as specific as it may seem at first sight because, with appropriate changes of variables, we may transform the integral into a much more general form.

To evaluate the integral (1.93) numerically, we can split the interval $[0,1]$ into $N^{1/D}$ subintervals of length $h = 1/N^{1/D}$; the centre of the i -th subinterval ($i = 1, \dots, N^{1/D}$)

is at $x_i = (i - 1/2)h$. This sets a partition of the unit cube into N cubic cells of volume h^D centred at the positions $(x_{i_1}, x_{i_2}, \dots, x_{i_D})$. The integral (1.93) can then be expressed as the sum of contributions from each cell,

$$I = \sum_{i_1=1}^{N^{1/D}} \int_{x_{i_1}-h/2}^{x_{i_1}+h/2} du_1 \dots \sum_{i_D=1}^{N^{1/D}} \int_{x_{i_D}-h/2}^{x_{i_D}+h/2} du_D F(u_1, \dots, u_D). \quad (1.94)$$

Within each cell, we can approximate the integrand by its Taylor expansion about the centre of the cell,

$$\begin{aligned} F(u_1, \dots, u_D) = & F(x_{i_1}, \dots, x_{i_D}) + \sum_{j=1}^D (u_j - x_{i_j}) \left[\frac{\partial F(u_1, \dots, u_D)}{\partial u_j} \right]_{\{u_\ell = x_{i_\ell}\}} \\ & + \frac{1}{2} \sum_{j,k=1}^D (u_j - x_{i_j})(u_k - x_{i_k}) \left[\frac{\partial^2 F(u_1, \dots, u_D)}{\partial u_j \partial u_k} \right]_{\{u_\ell = x_{i_\ell}\}} + \dots, \end{aligned} \quad (1.95)$$

where $\{u_\ell = x_{i_\ell}\}$ denotes the set of coordinates of the centre of the cell. If we retain only the constant zero-order terms in this expansion, we obtain the following first-order estimate of the integral:

$$I_{\text{num}} = h^D \sum_{i_1=1}^{N^{1/D}} \dots \sum_{i_D=1}^{N^{1/D}} F(x_{i_1}, \dots, x_{i_D}). \quad (1.96)$$

This formula involves only the N values of the function $F(u_1, \dots, u_D)$ at the centres of the N cells; it is essentially equivalent to the trapezoidal rule (see, *e.g.*, Abramowitz and Stegun, 1974). The error, $(\Delta I)_{\text{num}} = I - I_{\text{num}}$, can be estimated from the higher-order terms in the Taylor expansion (1.95). All the linear terms with $(u_j - x_{i_j})$, and all the quadratic terms with $(u_j - x_{i_j})(u_k - x_{i_k})$ and $j \neq k$ give vanishing contributions [because the factors $(u_j - x_{i_j})$ are antisymmetric with respect to the centre of the cell], and we have

$$\begin{aligned} (\Delta I)_{\text{num}} &= \sum_{i_1=1}^{N^{1/D}} \int_{x_{i_1}-h/2}^{x_{i_1}+h/2} du_1 \dots \sum_{i_D=1}^{N^{1/D}} \int_{x_{i_D}-h/2}^{x_{i_D}+h/2} du_D \frac{1}{2} \sum_{j=1}^D (u_j - x_{i_j})^2 \left[\frac{\partial^2 F}{\partial u_j^2} \right]_{\{u_\ell = x_{i_\ell}\}} \\ &= h^{D+2} \frac{1}{24} \sum_{i_1=1}^{N^{1/D}} \dots \sum_{i_D=1}^{N^{1/D}} \sum_{j=1}^D \left[\frac{\partial^2 F}{\partial u_j^2} \right]_{\{u_\ell = x_{i_\ell}\}}. \end{aligned} \quad (1.97)$$

Recalling that $h = N^{-1/D}$, the relative error of the numerical integral can be expressed as

$$(\Delta I)_{\text{num}} / I_{\text{num}} = N^{-2/D} e_{\text{num}}, \quad (1.98)$$

with

$$e_{\text{num}} = \frac{\sum_{i_1=1}^{N^{1/D}} \dots \sum_{i_D=1}^{N^{1/D}} \sum_{j=1}^D \left[\frac{\partial^2 F}{\partial u_j^2} \right]_{\{u_\ell = x_{i_\ell}\}}}{24 \sum_{i_1=1}^{N^{1/D}} \dots \sum_{i_D=1}^{N^{1/D}} F(x_{i_1}, \dots, x_{i_D})}. \quad (1.99)$$

Let us now consider the crude Monte Carlo calculation of the integral (1.93) with random points sampled uniformly in the unit cube, *i.e.*, with $p(u_1, \dots, u_D) = 1$. The relative (1σ) uncertainty of a Monte Carlo estimate, with the same number N of samples as cells in the numerical calculation, is [see Eq. (1.87)]

$$(\Delta I)_{\text{MC}}/I_{\text{MC}} = N^{-1/2} e_{\text{MC}}, \quad \text{with} \quad e_{\text{MC}} = \frac{\sqrt{\text{var}\{F(u_1, \dots, u_D)\}}}{I_{\text{MC}}}. \quad (1.100)$$

The important fact here is that the error of numerical quadrature is proportional to $N^{-2/D}$ while the statistical uncertainty of the Monte Carlo estimate is proportional to $N^{-1/2}$. Evidently, the calculation of the N values of the integrand takes the same time for both Monte Carlo and numerical quadrature. Moreover, the amount of numerical work needed for performing the rest of the calculation (sums and averages) is similar for both techniques. Therefore, assuming that the proportionality factors e_{num} and e_{MC} in Eqs. (1.99) and (1.100) are both finite, and that the number N of cells and samples is sufficiently large, we conclude that the Monte Carlo method will yield more accurate results than first-order numerical quadrature for dimensions $D > 4$.

It can be argued that with a more elaborate algorithm, numerical quadrature could beat Monte Carlo for dimensions higher than 4, and this may well be the case for specific problems with smooth functions. For instance, if we adopt the Simpson rule (see, *e.g.*, Abramowitz and Stegun, 1974) [which is analogous to retaining terms up to second order in the expansion (1.95)] the relative error of the numerical result would be proportional to $N^{-4/D}$,

$$(\Delta I)_{\text{num}}/I_{\text{num}} = N^{-4/D} e'_{\text{num}}, \quad (1.101)$$

with e'_{num} given by an expression similar to (1.99), but with fourth-order derivatives (cubic terms in (1.95) yield vanishing contributions, by symmetry). In this case, Monte Carlo is only expected to be more efficient than numerical quadrature for dimensions $D > 8$.

In turn, it may be claimed that with the aid of suitable variance-reduction techniques, the efficiency of Monte Carlo methods can be increased and make Monte Carlo more advantageous than numerical quadrature. In addition, the foregoing analysis is based on the assumption that the integrand in Eq. (1.93) can be evaluated *exactly*. In high-dimensionality problems, straight calculation of the function $F(u_1, \dots, u_D)$ may be very difficult and we may need to have recourse to interpolations or approximations. If the calculation of the integrand introduces errors with strong short-range correlations, the accuracy of numerical quadrature may be seriously impaired. The sensitivity of numerical quadrature to local errors of the integrand is determined by the order of the derivatives in the error factor e_{num} ; higher-order formulas are more sensitive to local errors. In this respect, low-order quadrature formulas (such as the trapezoidal rule studied above) are favoured, and this gives a distinct advantage to Monte Carlo methods.

In practice, the efficiencies of Monte Carlo integration and numerical quadrature depend to a great extent on the details of a given problem. The utility of Monte

Carlo stems not only from the convenient properties of statistical uncertainties for high-dimensionality problems, but also from the ease with which Monte Carlo methods can handle complicated geometries.

1.4 Simulation of radiation transport

In this Section, we describe the essentials of Monte Carlo simulation of radiation transport. For the sake of simplicity, we limit our considerations to the detailed simulation method, where all the interaction events experienced by a particle are simulated in chronological succession, and we disregard the production of secondary particles, so that only one kind of particle is transported.

The trajectory picture underlying conventional Monte Carlo simulations of radiation transport applies to homogeneous “random scattering” media, such as gases, liquids and amorphous solids, where the “molecules” are distributed at random with uniform density. The composition of the medium is specified by its stoichiometric formula, *i.e.*, atomic number Z_i and number of atoms per molecule n_i of all the elements present. The stoichiometric indices n_i need not have integer values. In the case of alloys, for instance, they may be set equal to the percentage in number of each element and then a “molecule” is a group of 100 atoms with the appropriate proportion of each element. The molar mass (mass of a mol of substance) is $A_M = \sum n_i A_i$, where A_i is the molar mass of the i -th element. The number of molecules per unit volume is given by

$$\mathcal{N} = N_A \frac{\rho}{A_M}, \quad (1.102)$$

where N_A is Avogadro’s number and ρ is the mass density of the material.

1.4.1 Interaction cross sections

Particles interact with the atoms or molecules of the medium through various competing mechanisms. Each interaction mechanism is characterised by the associated differential cross section (DCS), which is a function of the particle state variables that are modified in the course of the interaction. For simplicity, let us consider an interaction mechanism in which the projectile particle is deflected and loses energy. Compton scattering of photons and inelastic collisions of electrons are examples of this type of mechanism.

To define the DCS for our interaction mechanism, we consider the scattering experiment described in Fig. 1.7. A parallel monoenergetic beam of particles, with energy E and direction of movement $\hat{\mathbf{d}}$, impinges on a target atom or molecule, T, which is located at the origin of the reference frame. We assume that the beam is laterally homogeneous and that its lateral extension is much larger than the dimensions of the target. The beam is then characterised by its current density $\mathbf{J}_{\text{inc}}$ ³. We assume that particles interact only

³The current density vector $\mathbf{J}_{\text{inc}}$ is parallel to the direction of incidence $\hat{\mathbf{d}}$ and its magnitude is equal

through the considered mechanism; in an interaction, a particle loses a certain energy W and is deflected. A detector, placed at a macroscopic distance from the origin in the direction (θ, ϕ) and covering a small solid angle $d\Omega$, detects and counts all particles that enter its sensitive volume with energy in the interval $(E - W - dW, E - W)$ (*i.e.*, particles that have lost an energy between W and $W + dW$). We consider that the detector is effectively screened from the incident beam so that only particles that have undergone an interaction are counted. Let $\dot{N}_{\text{count}}$ denote the number of counts per unit time. The double-differential DCS (per unit solid angle and unit energy loss) is defined as

$$\frac{d^2\sigma}{d\Omega dW} \equiv \frac{\dot{N}_{\text{count}}}{|\mathbf{J}_{\text{inc}}| d\Omega dW}. \quad (1.103)$$

The DCS has the dimensions of area/(solid angle $\times$ energy); the product $[d^2\sigma/(d\Omega dW)] \times d\Omega dW$ represents the area of a plane surface that, placed perpendicularly to the incident beam, is hit by as many projectiles as are scattered into directions $\hat{\mathbf{d}}'$ within $d\Omega$ with energy loss between W and $W + dW$.

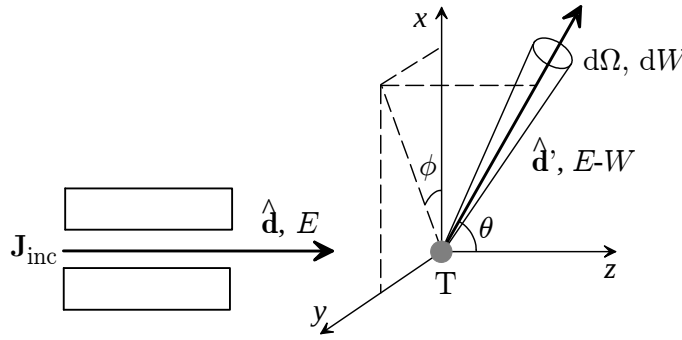


Figure 1.7: Schematic diagram of an experiment to measure the DCS.

The energy-loss DCS, differential in only the energy loss, is obtained by integrating the double-differential DCS over directions,

$$\frac{d\sigma}{dW} \equiv \int \frac{d^2\sigma}{d\Omega dW} d\Omega. \quad (1.104)$$

The total cross section σ is defined as the integral of the energy-loss DCS over the energy loss,

$$\sigma \equiv \int_0^E \frac{d\sigma}{dW} dW = \int_0^E \left(\int \frac{d^2\sigma}{d\Omega dW} d\Omega \right) dW. \quad (1.105)$$

Geometrically, the total cross section gives the area of a plane surface that, when placed perpendicularly to the incident beam, is crossed by the same number of projectiles that undergo interactions with any angular deflection and energy loss.

to the number of incident particles that cross a small probe surface, perpendicular to the beam and at rest with respect to the target, per unit time and unit probe surface.

1.4.2 Mean free path

Let us consider that our particles move within a random-scattering medium with $\mathcal{N}$ molecules per unit volume. We wish to determine the PDF $p(s)$ of the path length s of a particle from its current position to the site of the next interaction. To get a more visual picture of the interaction process, we can replace each molecule by a sphere of radius r_s such that the cross-sectional area πr_s^2 equals the total cross section σ_T . An interaction takes place when the particle strikes one of these spheres.

We can start by studying a simpler situation, in which a homogeneous beam of particles impinges normally on a very thin material foil of thickness ds (see Fig. 1.8). What the incident particles see directly ahead of their path is a uniform distribution of $\mathcal{N}ds$ spheres per unit surface. Let J be the current density of the incident beam. The current density of particles transmitted through the foil without interacting is $J - dJ$, where $dJ = J\mathcal{N}\sigma ds$ is the number of particles that undergo interactions per unit time and unit surface of the foil (note that $\mathcal{N}\sigma ds$ is the fractional area covered by the spheres). Therefore, the interaction probability per unit path length is

$$\frac{dJ}{J} \frac{1}{ds} = \mathcal{N}\sigma. \quad (1.106)$$

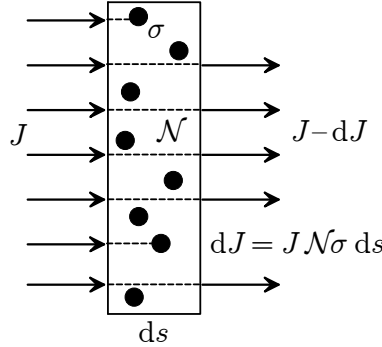


Figure 1.8: Attenuation of a beam through a thin material foil.

Let us now return to our original problem, where particles move within an unbounded medium. The probability that a particle travels a path length s without interacting is

$$\mathcal{F}(s) = \int_s^\infty p(s') ds'. \quad (1.107)$$

The probability $p(s) ds$ of having the next interaction when the travelled length is in the interval $(s, s + ds)$ equals the product of $\mathcal{F}(s)$ (the probability of arrival at s without interacting) and $\mathcal{N}\sigma ds$ (the probability of interacting within ds). It then follows that

$$p(s) = \mathcal{N}\sigma \int_s^\infty p(s') ds'. \quad (1.108)$$

The solution of this integral equation, with the boundary condition $p(\infty) = 0$, is the familiar exponential distribution

$$p(s) = \mathcal{N}\sigma \exp[-s(\mathcal{N}\sigma)]. \quad (1.109)$$

The mean free path λ is defined as the average path length between collisions:

$$\lambda \equiv \langle s \rangle = \int_0^\infty s p(s) ds = \frac{1}{\mathcal{N}\sigma}. \quad (1.110)$$

Its inverse,

$$\lambda^{-1} = \mathcal{N}\sigma, \quad (1.111)$$

is the interaction probability per unit path length.

1.4.3 Scattering model and probability distributions

Consider a particle with energy E (kinetic energy, in the case of electrons and positrons) moving in a given medium. In each interaction, the particle may lose a certain energy W and change its direction of movement. The angular deflection is determined by the polar scattering angle θ , *i.e.*, the angle between the directions of the particle before and after the interaction, and the azimuthal angle ϕ (see Fig. 1.7). These quantities are random variables, whose PDFs are determined by the molecular DCSs.

Let us assume that particles can interact with the medium through two independent mechanisms, denoted as “A” and “B” (for instance, elastic and inelastic scattering, in the case of low-energy electrons). The scattering model is completely specified by the molecular DCSs

$$\frac{d^2\sigma_A(E; \theta, W)}{d\Omega dW} \quad \text{and} \quad \frac{d^2\sigma_B(E; \theta, W)}{d\Omega dW}, \quad (1.112)$$

where $d\Omega$ is a solid angle element in the direction (θ, ϕ) . We have made the parametric dependence of the DCSs on the particle energy E explicit. Considering that the molecules in the medium are oriented at random, the DCS is independent of the azimuthal scattering angle, *i.e.*, the angular distribution of scattered particles is axially symmetrical around the direction of incidence. The total cross sections (per molecule) for mechanisms A and B are

$$\sigma_{A,B}(E) = \int_0^E dW \int_0^\pi 2\pi \sin \theta d\theta \frac{d^2\sigma_{A,B}(E; \theta, W)}{d\Omega dW}. \quad (1.113)$$

The total interaction cross section is

$$\sigma_T(E) = \sigma_A(E) + \sigma_B(E). \quad (1.114)$$

The interaction probability per unit path length is

$$\lambda_T^{-1} = \mathcal{N}\sigma_T. \quad (1.115)$$

Notice that the total inverse mean free path is the sum of the inverse mean free paths of the different active interaction mechanisms,

$$\lambda_T^{-1} = \lambda_A^{-1} + \lambda_B^{-1}. \quad (1.116)$$

The PDF of the path length s of a particle from its current position to the site of the next collision is

$$p(s) = \lambda_T^{-1} \exp(-s/\lambda_T). \quad (1.117)$$

The average path length between interactions equals the (total) mean free path,

$$\langle s \rangle = \lambda_T = (\mathcal{N}\sigma_T)^{-1}. \quad (1.118)$$

When the particle interacts with the medium, the kind of interaction that occurs is a discrete random variable, that takes the values “A” and “B” with probabilities

$$p_A = \sigma_A/\sigma_T \quad \text{and} \quad p_B = \sigma_B/\sigma_T. \quad (1.119)$$

The PDFs of the polar scattering angle and the energy loss in individual scattering events are

$$p_{A,B}(E; \theta, W) = \frac{2\pi \sin \theta}{\sigma_{A,B}(E)} \frac{d^2 \sigma_{A,B}(E; \theta, W)}{d\Omega dW}. \quad (1.120)$$

Notice that $p_A(E; \theta, W) d\theta dW$ gives the (normalised) probability that, in a scattering event of type A, the particle loses energy in the interval $(W, W + dW)$ and is deflected into directions with polar angle (relative to the initial direction) in the interval $(\theta, \theta + d\theta)$. The azimuthal scattering angle in each collision is uniformly distributed in the interval $(0, 2\pi)$, *i.e.*,

$$p(\phi) = \frac{1}{2\pi}. \quad (1.121)$$

Our analysis of the interactions in a random-scattering medium involves the DCSs, which describe the interactions with individual molecules, and the density of molecules $\mathcal{N}$, which characterises the macroscopic state of the medium. In cases where the DCSs are affected by the aggregation state of the medium, it could be more appropriate to describe each interaction mechanism by means of the so-called *differential inverse mean free path*. The differential inverse mean free path for the interaction process A is defined as

$$\frac{d^2 \lambda_A^{-1}(E; \theta, W)}{d\Omega dW} = \mathcal{N} \frac{d^2 \sigma_A(E; \theta, W)}{d\Omega dW}. \quad (1.122)$$

The differential inverse mean free path has the dimensions of $(\text{length} \times \text{solid angle} \times \text{energy})^{-1}$; the product $[d^2 \lambda_A^{-1}/(d\Omega dW)] d\Omega dW$ is the probability per unit path length that a particle undergoes a collision of type A with angular deflection within $d\Omega$ and energy loss between W and $W + dW$. Evidently, the integral of the differential inverse mean free path gives the inverse mean free path for the process,

$$\lambda_A^{-1} = \int dW \int 2\pi \sin \theta d\theta \frac{d^2 \lambda_A^{-1}(E; \theta, W)}{d\Omega dW} = \mathcal{N} \sigma_A. \quad (1.123)$$

In the literature, the product $\mathcal{N} \sigma_A$ is frequently called the *macroscopic cross section*, although this name is not appropriate for a quantity that has the dimensions of inverse length.

1.4.4 Generation of random track

Each particle track starts off at a given position, with initial direction and energy in accordance with the characteristics of the source. The “state” of a particle immediately after an interaction (or after entering the sample or starting its trajectory) is defined by its position coordinates $\mathbf{r} = (x, y, z)$, energy E and direction cosines of the direction of flight, *i.e.*, the components of the unit vector $\hat{\mathbf{d}} = (u, v, w)$, as seen from the laboratory reference frame. Each simulated track is thus characterised by a series of states $\mathbf{r}_n, E_n, \hat{\mathbf{d}}_n$, where $\mathbf{r}_n$ is the position of the n -th scattering event and E_n and $\hat{\mathbf{d}}_n$ are the energy and direction cosines of the direction of movement just *after* that event.

The generation of random tracks proceeds as follows. Let us assume that a track has already been simulated up to a state $\mathbf{r}_n, E_n, \hat{\mathbf{d}}_n$. The length s of the free path to the next collision, the involved scattering mechanism, the change of direction and the energy loss in this collision are random variables that are sampled from the corresponding PDFs, using the methods described in Section 1.2. Hereafter, ξ stands for a random number uniformly distributed in the interval $(0,1)$.

The length of the free flight is distributed according to the PDF given by Eq. (1.117). Random values of s are generated by using the sampling formula [see Eq. (1.36)]

$$s = -\lambda_T \ln \xi. \quad (1.124)$$

The following interaction occurs at the position

$$\mathbf{r}_{n+1} = \mathbf{r}_n + s\hat{\mathbf{d}}_n. \quad (1.125)$$

The interaction type (“A” or “B”) is selected from the point probabilities given by Eq. (1.119) using the inverse-transform method (Section 1.2.3). The polar scattering angle θ and the energy loss W are sampled from the distribution $p_{A,B}(E; \theta, W)$, Eq. (1.120), by using a suitable sampling technique. The azimuthal scattering angle is generated, according to the uniform distribution in $(0, 2\pi)$, as $\phi = 2\pi\xi$.

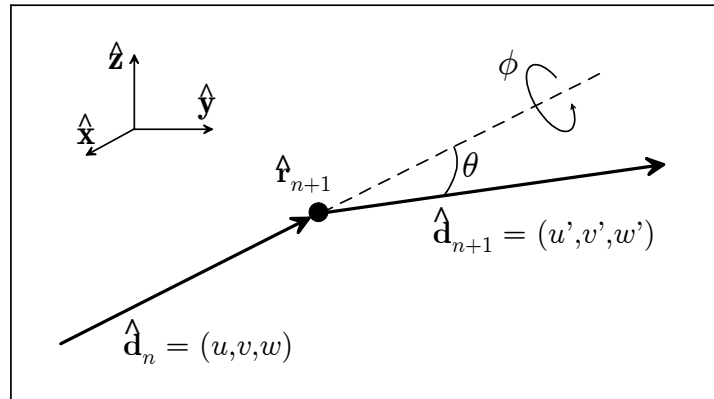


Figure 1.9: Angular deflections in single-scattering events.

After sampling the values of W , θ and ϕ , the energy of the particle is reduced, $E_{n+1} = E_n - W$, and the direction of movement after the interaction, $\hat{\mathbf{d}}_{n+1} = (u', v', w')$, is obtained by performing a rotation of $\hat{\mathbf{d}}_n = (u, v, w)$ (see Fig. 1.9). The rotation matrix $\mathcal{R}(\theta, \phi)$ is determined by the polar and azimuthal scattering angles. To explicitly obtain the direction vector $\hat{\mathbf{d}}_{n+1} = \mathcal{R}(\theta, \phi)\hat{\mathbf{d}}_n$ after the interaction, we first note that, if the initial direction is along the z -axis, $\hat{\mathbf{d}}_n = \hat{\mathbf{z}} = (0, 0, 1)$, the direction after the collision is⁴

$$\begin{pmatrix} \sin \theta \cos \phi \\ \sin \theta \sin \phi \\ \cos \theta \end{pmatrix} = \mathcal{R}_z(\phi)\mathcal{R}_y(\theta) \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \quad (1.126)$$

where

$$\mathcal{R}_y(\theta) = \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix} \quad \text{and} \quad \mathcal{R}_z(\phi) = \begin{pmatrix} \cos \phi & -\sin \phi & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (1.127)$$

are rotation matrices corresponding to active rotations of angles θ and ϕ about the y - and z -axes, respectively. On the other hand, if ϑ and φ are the polar and azimuthal angles of the initial direction

$$\hat{\mathbf{d}}_n = (\sin \vartheta \cos \varphi, \sin \vartheta \sin \varphi, \cos \vartheta), \quad (1.128)$$

the rotation $\mathcal{R}_y(-\vartheta)\mathcal{R}_z(-\varphi)$ transforms the vector $\hat{\mathbf{d}}_n$ into $\hat{\mathbf{z}}$. It is then clear that the final direction vector $\hat{\mathbf{d}}_{n+1}$ can be obtained by performing the following sequence of rotations of the initial direction vector: 1) $\mathcal{R}_y(-\vartheta)\mathcal{R}_z(-\varphi)$, which transforms $\hat{\mathbf{d}}_n$ into $\hat{\mathbf{z}}$; 2) $\mathcal{R}_z(\phi)\mathcal{R}_y(\theta)$, which rotates $\hat{\mathbf{z}}$ according to the sampled polar and azimuthal scattering angles; and 3) $\mathcal{R}_z(\varphi)\mathcal{R}_y(\vartheta)$, which inverts the rotation of the first step. Hence

$$\mathcal{R}(\theta, \phi) = \mathcal{R}_z(\varphi)\mathcal{R}_y(\vartheta)\mathcal{R}_z(\phi)\mathcal{R}_y(\theta)\mathcal{R}_y(-\vartheta)\mathcal{R}_z(-\varphi). \quad (1.129)$$

The final direction vector is

$$\hat{\mathbf{d}}_{n+1} = \mathcal{R}(\theta, \phi)\hat{\mathbf{d}}_n = \mathcal{R}_z(\varphi)\mathcal{R}_y(\vartheta) \begin{pmatrix} \sin \theta \cos \phi \\ \sin \theta \sin \phi \\ \cos \theta \end{pmatrix} \quad (1.130)$$

and its direction cosines are

$$\begin{aligned} u' &= u \cos \theta + \frac{\sin \theta}{\sqrt{1-w^2}} [uw \cos \phi - v \sin \phi], \\ v' &= v \cos \theta + \frac{\sin \theta}{\sqrt{1-w^2}} [vw \cos \phi + u \sin \phi], \\ w' &= w \cos \theta - \sqrt{1-w^2} \sin \theta \cos \phi. \end{aligned} \quad (1.131)$$

⁴Vectors are defined by specifying their Cartesian components. When a vector is defined within a paragraph, or in a displayed equation, it is customary to represent it as a one-row matrix. However, in the product of a matrix by a vector, the vector must be represented as a one-column matrix.

These equations are indeterminate when $w \simeq \pm 1$, *i.e.*, when the initial direction is nearly parallel or antiparallel to the z -axis; in this case we can simply set

$$u = \pm \sin \theta \cos \phi, \quad v = \pm \sin \theta \sin \phi, \quad w = \pm \cos \theta. \quad (1.132)$$

Moreover, Eqs. (1.131) are not very stable numerically and the normalisation of $\hat{\mathbf{d}}_{n+1}$ tends to drift from 1 after repeated usage. This must be remedied by periodically renormalizing $\hat{\mathbf{d}}_{n+1}$. The change of direction expressed by Eqs. (1.131) and (1.132) is performed by the subroutine DIRECT (see the PENELOPE source listing).

The simulation of the track then proceeds by repeating these steps (see Fig. 1.10). A track is finished either when it leaves the material system or when the energy becomes smaller than a given energy E_{abs} , which is the energy where particles are assumed to be effectively stopped and absorbed in the medium.

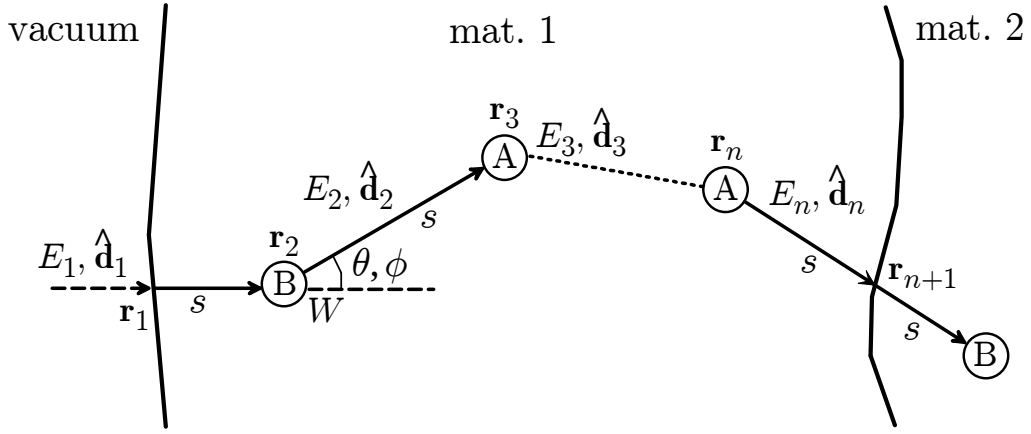


Figure 1.10: Generation of random trajectories using detailed simulation. A particle enters material 1 from the vacuum and, after multiple interactions crosses the interface between materials 1 and 2.

It is worth recalling that the kind of analogue trajectory model that we have described is only valid when diffraction effects resulting from coherent scattering from several centres (*e.g.*, Bragg diffraction, channelling of charged particles) are negligible. This means that the simulation is applicable only to amorphous media and, with some care, to polycrystalline solids.

1.4.5 Particle transport as a Markov process

The foregoing concepts, definitions and simulation scheme rest on the assumption that particle transport can be modelled as a Markov process⁵, *i.e.*, “future values of a random variable (interaction event) are statistically determined by present events and depend

⁵The quoted definition is from the Webster’s Encyclopedic Unabridged Dictionary of the English Language (Portland House, New York, 1989).

only on the event immediately preceding”. Owing to the Markovian character of the transport, we can stop the generation of a particle history at an arbitrary state (any point of the track) and resume the simulation from this state without introducing any bias in the results.

Up to this point we have considered transport in a single homogeneous medium. In practical cases, however, the material structure where radiation is transported may consist of various regions with different compositions. We assume that the interfaces between contiguous media are sharp (*i.e.*, there is no diffusion of chemical species across them) and passive (which amounts to neglecting, *e.g.*, surface plasmon excitation and transition radiation). In the simulation code, when a particle arrives at an interface, it is stopped there and the simulation is resumed with the interaction properties of the new medium. Obviously, this procedure is consistent with the Markovian property of the transport process. Thus, interface crossings are consistently described by means of simple geometrical considerations. The main advantage of Monte Carlo simulation lies in the ease of handling complicated geometries; this is at variance with conventional numerical methods, which find great difficulties in accommodating non-trivial boundary conditions.

Consider two homogeneous media, 1 and 2 (with corresponding mean free paths $\lambda_{T,1}$ and $\lambda_{T,2}$), separated by an interface, which is crossed by particles that move from the first medium to the second (as in Fig. 1.10). The average path length between the last real interaction in medium 1 and the first real interaction in medium 2 is $\lambda_{T,1} + \lambda_{T,2}$, as can be easily verified by simulation. This result seemed paradoxical to some authors and induced confusion in the past. In fact, there is nothing odd here as it can be easily verified (again by simulation) as follows. Assume particles being transported within a single homogeneous medium with an imaginary plane that acts as a “virtual” interface, splitting the medium into two halves. In the simulation, the particles do not see this interface, *i.e.*, they do not stop when crossing it. Every time a particle crosses the plane, we score the length s_{plane} of the track segment between the two real interactions immediately before and after the crossing. It is found that the average value of s_{plane} is $2\lambda_T$, in spite of the fact that the free path length between consecutive collisions was sampled from an exponential PDF with the mean free path λ_T [yes, the scored values s_{plane} were generated from this PDF!]. The explanation of this result is that, as a consequence of the Markovian character, the average path length from the plane (an arbitrary *fixed* point in the track) back to the last collision (or up to the next collision) is λ_T .

In mixed simulations of electron/positron transport, it is necessary to limit the length s of each “free jump” so that it does not exceed a given value s_{max} . To accomplish this, we still sample the free path length s to the next interaction from the exponential PDF (1.117), but when $s > s_{\text{max}}$ we only let the particle advance a distance s_{max} along the direction of motion. At the end of the truncated free jump we do nothing (*i.e.*, the particle keeps its energy and direction of motion unaltered); however, for programming convenience, we shall say that the particle suffers a *delta interaction* (actually, a “non-interaction”). When the sampled value of s is less than s_{max} , a real interaction is

simulated. After the interaction (either real or delta), we sample a new free path s , move the particle a distance $s' = \min(s, s_{\max})$, etc. From the Markovian character of the transport, it is clear that the insertion of delta interactions keeps the simulation unbiased. If this is not immediately clear, here is a direct proof. First we note that the probability that a free jump ends with a delta interaction is

$$p_\delta = \int_{s_{\max}}^{\infty} p(s) ds = \exp(-s_{\max}/\lambda_T). \quad (1.133)$$

To obtain the probability $p(s) ds$ of having the first real interaction at a distance in the interval $(s, s + ds)$, we write $s = ns_{\max} + s'$ with $n = [s/s_{\max}]$ and, hence, $s' < s_{\max}$. The sought probability is then equal to the probability of having n successive delta interactions followed by a real interaction at a distance in $(s', s' + ds)$ from the last, n -th, delta interaction,

$$p(s) ds = p_\delta^n \lambda_T^{-1} \exp(-s'/\lambda_T) ds = \lambda_T^{-1} \exp(-s/\lambda_T) ds, \quad (1.134)$$

which is the correct value [cf. Eq. (1.117)].

1.5 Statistical averages and uncertainties

For the sake of being more specific, let us consider the simulation of a high-energy electron beam impinging on the surface of a semi-infinite water phantom. Each primary electron originates a shower of electrons and photons, which are individually tracked down to the corresponding absorption energy. Any quantity of interest Q is evaluated as the average score of a large number N of simulated random showers. Formally, Q can be expressed as an integral of the form (1.79),

$$Q = \int q p(q) dq, \quad (1.135)$$

where the PDF $p(q)$ is usually unknown. The simulation of individual showers provides a practical method to sample q from the “natural” PDF $p(q)$: from each generated shower we get a random value q_i distributed according to $p(q)$. The only difference to the case of Monte Carlo integration considered above is that now the PDF $p(q)$ describes a cascade of random interaction events, each with its characteristic PDF. The Monte Carlo estimate of Q is

$$\bar{Q} = \frac{1}{N} \sum_{i=1}^N q_i. \quad (1.136)$$

Thus, for instance, the average energy E_{dep} deposited within the water phantom per incident electron is obtained as

$$E_{\text{dep}} = \frac{1}{N} \sum_{i=1}^N e_i, \quad (1.137)$$

where e_i is the energy deposited by *all* the particles of the i -th shower. The statistical uncertainty (standard deviation) of the Monte Carlo estimate [Eq. (1.87)] is

$$\sigma_Q = \sqrt{\frac{\text{var}(q)}{N}} = \sqrt{\frac{1}{N} \left[\frac{1}{N} \sum_{i=1}^N q_i^2 - \bar{Q}^2 \right]}. \quad (1.138)$$

As mentioned above, we shall usually express the simulation result in the form $\bar{Q} \pm 3\sigma_Q$, so that the interval $(\bar{Q} - 3\sigma_Q, \bar{Q} + 3\sigma_Q)$ contains the true value Q with 99.7% probability. Notice that to evaluate the standard deviation (1.138) we must score the squared contributions q_i^2 . In certain cases, the contributions q_i can only take the values 0 and 1, and the standard error can be determined without scoring the squares,

$$\sigma_Q = \sqrt{\frac{1}{N} \bar{Q}(1 - \bar{Q})}. \quad (1.139)$$

Simulation/scoring can also be used to compute continuous distributions. The simplest method is to “discretise” the distributions, by treating them as histograms, and to determine the “heights” of the different bars. To make the arguments clear, let us consider the depth-dose distribution $D(z)$, defined as the average energy deposited per unit depth and per incident electron within the water phantom. $D(z)dz$ is the average energy deposited at depths between z and $z+dz$ per incident electron, and the integral of $D(z)$ from 0 to ∞ is the average deposited energy E_{dep} (again, per incident electron). Since part of the energy is reflected back from the water phantom (through backscattered radiation), E_{dep} is less than the kinetic energy E_{inc} of the incident electrons. We are interested in determining $D(z)$ in a limited depth interval, say from $z = z_{\min}$ to $z = z_{\max}$. The calculation proceeds as follows. First of all, we have to select a partition of the interval $(z_{\min}, z_{\max})$ into M different depth bins (z_{k-1}, z_k) , with $z_{\min} = z_0 < z_1 < \dots < z_M = z_{\max}$. Let $e_{ij,k}$ denote the amount of energy deposited into the k -th bin by the j -th particle of the i -th shower (each incident electron may produce multiple secondary particles). The average energy deposited into the k -th bin (per incident electron) is obtained as

$$E_k = \frac{1}{N} \sum_{i=1}^N e_{i,k} \quad \text{with} \quad e_{i,k} \equiv \sum_j e_{ij,k}, \quad (1.140)$$

and is affected by a statistical uncertainty

$$\sigma_{E_k} = \sqrt{\frac{1}{N} \left[\frac{1}{N} \sum_{i=1}^N e_{i,k}^2 - E_k^2 \right]}. \quad (1.141)$$

The Monte Carlo depth-dose distribution $D_{\text{MC}}(z)$ is a stepwise constant function (see Fig. 1.11),

$$D_{\text{MC}}(z) = D_k \pm 3\sigma_{Dk} \quad \text{for } z_{k-1} < z < z_k \quad (1.142)$$

with

$$D_k \equiv \frac{1}{z_k - z_{k-1}} E_k, \quad \sigma_{Dk} \equiv \frac{1}{z_k - z_{k-1}} \sigma_{Ek}. \quad (1.143)$$

Notice that the bin average and standard deviation have to be divided by the bin width to obtain the final Monte Carlo distribution. Defined in this way, $D_{MC}(z)$ is an unbiased estimator of the *average* energy deposited per unit depth in each bin. The limitation here is that we are approximating the continuous distribution $D(z)$ as a histogram with finite bar widths. In principle, we could obtain a closer approximation by using narrower bins. However, care has to be taken in selecting the bin widths because statistical uncertainties may completely hide the information in narrow bins.

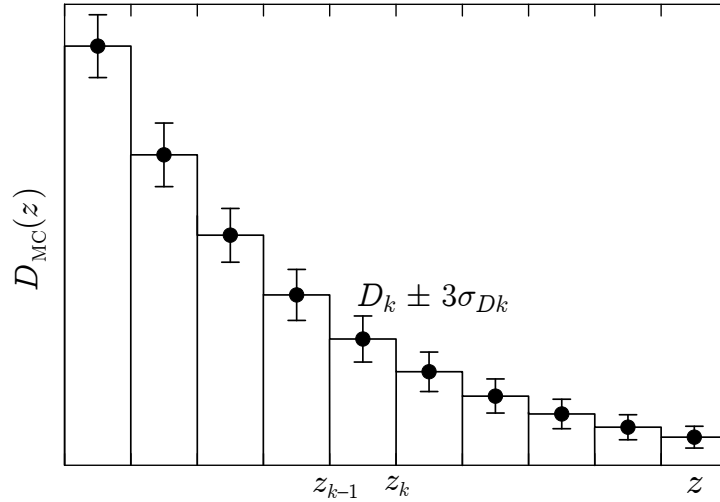


Figure 1.11: Typical continuous distribution generated by a Monte Carlo simulation, represented as a histogram. With the definition given by Eq. (1.143), the area of the histogram is independent of the bin size, except for statistical uncertainties.

A few words regarding programming details are in order. To evaluate the average deposited energy and its standard deviation for each bin, Eqs. (1.140) and (1.141), we must score the shower contributions $e_{i,k}$ and their squares $e_{i,k}^2$. There are cases in which the literal application of this recipe may take a large fraction of the simulation time. Consider, for instance, the simulation of the 3D dose distribution in the phantom, which may involve several thousand volume bins. For each bin, the energies $e_{ij,k}$ deposited by the individual particles of a shower must be accumulated in a partial counter to obtain the shower contribution $e_{i,k}$ and, after completion of the whole shower, the value $e_{i,k}$ and its square must be added to the accumulated counters. As only a small fraction of the bins receive energy from a single shower, it is not practical to treat all bin counters on an equal footing. The fastest method is to transfer partial scores to the accumulated counters only when the partial counter will receive a contribution from a new shower. This can be easily implemented in a computer program as follows. For each quantity of interest, say Q , we define three real counters, Q , $Q2$ and QP , and an integer label LQ ; all these quantities are initially set to zero. The partial scores q_{ij} of the particles of a shower

are accumulated in the partial counter QP , whereas the global shower contribution q_i and its square are accumulated in Q and $Q2$, respectively. Each shower is assigned a label, for instance its order number i , which is stored in LQ the first time that the shower contributes to QP . In the course of the simulation, the value of QP is transferred to the global counters Q and $Q2$ only when it is necessary to store a contribution q_{ij} from a new shower. Explicitly, the Fortran code for scoring Q is

```

      IF (i.NE.LQ) THEN
        Q=Q+QP
        Q2=Q2+QP**2
        QP=qij
        LQ=i
      ELSE
        QP=QP+qij
      ENDIF

```

At the end of the simulation, the residual contents of QP must be transferred to the global counters.

For some quantities (*e.g.*, the mean number of scattering events per track, the depth-dose distribution, ...) almost all the simulated tracks contribute to the score and the inherent statistical uncertainties of the simulation results are comparatively small. Other quantities (*e.g.*, angle and energy distributions of the particles transmitted through a thick foil) have considerable statistical uncertainties (*i.e.*, large variances) because only a small fraction of the simulated tracks contribute to the partial scores.

1.6 Variance reduction

In principle, the statistical uncertainty of a quantity may be somewhat reduced (without increasing the computer simulation time) by using variance-reduction techniques. Unfortunately, these optimisation techniques are extremely problem-dependent, and general recipes to minimise the variance cannot be given. On the other hand, the importance of variance reduction should not be overvalued. In many cases, analogue⁶ simulation does the work in a reasonable time. Spending man-hours by complicating the program to get a modest reduction in computing time may not be a good investment. It is also important to realise that an efficient variance-reduction method usually lowers the statistical error of a given quantity Q at the expense of increasing the uncertainties of other quantities. Thus, variance-reduction techniques are not recommended when a global description of the transport process is sought. Here we give a brief description of techniques which, with a modest programming effort, can be useful in improving the solution of some ill-conditioned problems. For the sake of generality, we consider that secondary particles can be generated in the interactions with the medium. A nice, and

⁶We use the term “analogue” to refer to detailed, condensed or mixed simulations that do not incorporate variance-reduction procedures.

practically oriented, review of variance-reduction methods in radiation transport has been given by Bielajew and Rogers (1988).

1.6.1 Interaction forcing

Sometimes, a high variance results from an extremely low interaction probability. Consider, for instance, the simulation of the energy spectrum of bremsstrahlung photons emitted by medium energy (~ 100 keV) electrons in a thin foil of a certain material. As radiative events are much less probable than elastic and inelastic scattering, the uncertainty of the simulated photon spectrum will be relatively large. In such cases, an efficient variance-reduction method is to artificially increase the interaction probability of the process A of interest, *i.e.*, to force interactions of type A to occur more frequently than for the real process. Our practical implementation of interaction forcing consists of replacing the mean free path λ_A of the real process by a shorter one, $\lambda_{A,f}$. We consider that the PDFs for the energy loss and the angular deflections (and the directions of emitted secondary particles, if any) in the forced interactions are the same as for the real interactions. To sample the length of the free jump to the next interaction, we use the exponential distribution with the reduced mean free path $\lambda_{A,f}$. This is equivalent to increasing the interaction probability per unit path length of the process A by a factor

$$\mathcal{F} = \frac{\lambda_A}{\lambda_{A,f}} > 1. \quad (1.144)$$

To keep the simulation unbiased, we must correct for the introduced distortion as follows:

- (i) A weight $w = 1$ is associated with each primary particle. Secondary particles produced in forced interactions of a particle with weight w are given a weight $w_s = w/\mathcal{F}$. Secondary particles generated in non-forced interactions (*i.e.*, of types other than A) are given a weight equal to that of their parent particle.
- (ii) Forced interactions are simulated to determine the energy loss and possible emission of secondary radiation, but the state variables of the interacting particle are altered only with probability $1/\mathcal{F}$. That is, the energy E and direction of movement $\hat{\mathbf{d}}$ of the projectile are varied only when the value ξ of a random number falls below $1/\mathcal{F}$, otherwise E and $\hat{\mathbf{d}}$ are kept unchanged.
- (iii) A weight $w_E = w/\mathcal{F}$ is given to the deposited energy⁷ (and to any other alteration of the medium such as, *e.g.*, charge deposition) that results from forced interactions of a particle with weight w . For non-forced interactions $w_E = w$.

Of course, interaction forcing should be applied only to interactions that are dynamically allowed, *i.e.*, for particles with energy above the corresponding “reaction” threshold.

⁷This option may effectively reduce the statistical uncertainties of simulated dose distributions in very thin media. However, it violates energy conservation (because the sum of energies deposited along a track differs from the energy lost by the projectile) and, therefore, yields energy deposition spectra that are biased.

Let w_{i1} and f_{i1} denote the weight and the contribution to the score of the i -th primary, and let w_{ij} and f_{ij} ($j > 1$) represent the weights and contributions of the j -th secondary particles generated by the i -th primary. The Monte Carlo estimate of F obtained from the N simulated histories is

$$\bar{F} = \frac{1}{N} \sum_{i,j} w_{ij} f_{ij}. \quad (1.145)$$

Evidently, the estimates $\bar{F}$ obtained with interaction forcing and from an analogue simulation are equal (in the statistical sense, *i.e.*, in the limit $N \rightarrow \infty$, their difference tends to zero). The standard deviation is given by

$$\sigma_F = \sqrt{\frac{1}{N} \left[\frac{1}{N} \sum_i \left(\sum_j w_{ij} f_{ij} \right)^2 - \bar{F}^2 \right]}. \quad (1.146)$$

Quantities directly related to forced interactions will have a reduced statistical error, due to the increase in number of these interactions. However, for a given simulation time, other quantities may exhibit standard deviations larger than those of the analogue simulation, because of the time spent in simulating the forced interactions.

1.6.2 Splitting and Russian roulette

These two techniques, which are normally used in conjunction, are effective in problems where interest is focused on a localised spatial region. Typical examples are the calculation of dose functions in deep regions of irradiated objects and, in the case of collimated radiation beams, the evaluation of radial doses far from the beam axis. The basic idea of splitting and Russian roulette methods is to favour the flux of radiation towards the region of interest and inhibit the radiation that leaves that region. These techniques are also useful in other problems where only a partial description of the transport process is required. The “region of interest” may then be a limited volume in the space of state variables $(\mathbf{r}, E, \hat{\mathbf{d}})$. Thus, in studies of radiation backscattering, the region of interest may be selected as the spatial region of the sample close to the irradiated surface *and* the set of particle directions that point towards this surface.

As in the case of interaction forcing, variance reduction is accomplished by modifying the weights of the particles. It is assumed that primary particles start moving with unit weight and each secondary particle produced by a primary one is assigned an initial weight equal to that of the primary. Splitting consists of transforming a particle, with weight w_0 and in a certain state, into a number $\mathcal{S} > 1$ of identical particles with weights $w = w_0/\mathcal{S}$ in the same state. Splitting should be applied when the particle “approaches” the region of interest. The Russian roulette technique is, in a way, the reverse process: when a particle tends to move away from the region of interest it is “killed” with a certain probability, $\mathcal{K} < 1$, and, if it survives, its weight is increased by a factor $1/(1 - \mathcal{K})$. Here, killing means that the particle is just discarded (and does not contribute to the scores

anymore). Evidently, splitting and killing leave the simulation unbiased. The mean and standard deviation of the calculated quantities are given by Eqs. (1.145) and (1.146). The effectiveness of these methods relies on the adopted values of the parameters $\mathcal{S}$ and $\mathcal{K}$, and on the strategy used to decide when splitting and killing are to be applied. These details can only be dictated by the user's experience.

1.6.3 Other methods

Very frequently, an effective “reduction of variance” may be obtained by simply avoiding unnecessary calculations. This is usually true for simulation codes that incorporate general-purpose geometry packages. In the case of simple (*e.g.*, planar, spherical, cylindrical) geometries the program may be substantially simplified and this may speed up the simulation appreciably. In general, the clever use of possible symmetries of the problem under consideration may lead to spectacular variance reductions. As a last example, we can quote the so-called “range rejection” method, which simply consists of absorbing a particle when it (and its possible secondaries) cannot leave (or reach) regions of interest. Range rejection is useful, *e.g.*, when computing the total energy deposition of electrons or positrons in a given spatial region. When the residual range of a particle (and its possible secondaries) is less than the distance to the nearest limiting surface of the region of interest, the particle will deposit all its energy either inside or outside the considered region (depending on its current position) and the simulation of the track can be stopped. Range rejection is not adequate for photon transport simulation, since the concept of photon range is not well defined (or, to be more precise, photon path length fluctuations are very large).

Chapter 2

Photon interactions

In this Chapter, we consider the interactions of unpolarised photons of energy E with atoms of atomic number Z . We limit our considerations to the energy range from 50 eV up to 1 GeV, where the dominant interaction processes are coherent (Rayleigh) scattering, incoherent (Compton) scattering, the photoelectric effect and electron-positron pair production. Other interactions, such as photonuclear absorption, occur with much smaller probability and can be disregarded for most practical purposes (see, *e.g.*, Hubbell *et al.*, 1980).

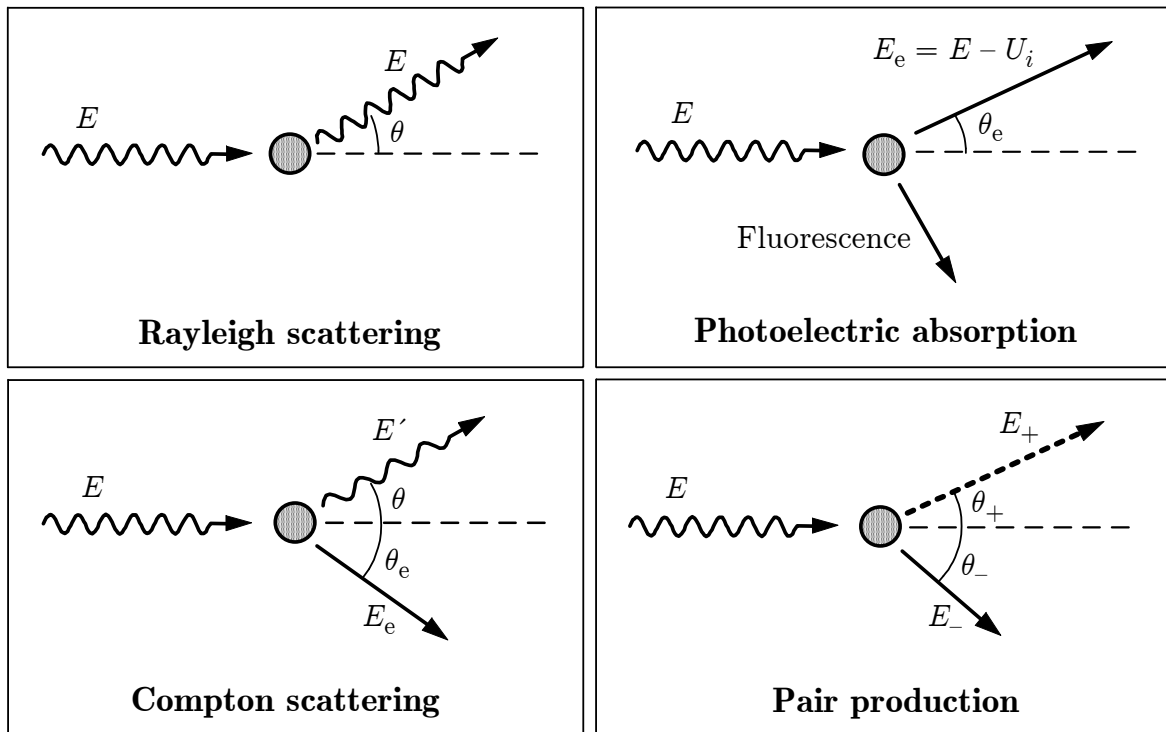


Figure 2.1: Basic interactions of photons with matter.

As long as the response of an atom is not appreciably distorted by molecular binding, the single-atom theory can be extended to molecules by using the additivity approximation, *i.e.*, the molecular cross section for a process is approximated by the sum of the atomic cross sections of all the atoms in the molecule. The additivity approximation can also be applied to dense media whenever interference effects between waves scattered by different centres (which, for instance, give rise to Bragg diffraction in crystals) are small. We assume that these conditions are always satisfied.

The ability of Monte Carlo simulation methods to describe photon transport in complex geometries has been established from research during the last five decades (Hayward and Hubbell, 1954; Zerby, 1963; Berger and Seltzer, 1972; Chan and Doi, 1983; Ljungberg and Strand, 1989). The most accurate DCSs available are given in numerical form and, therefore, advanced Monte Carlo codes make use of extensive databases. To reduce the amount of required numerical information, in PENELOPE we use a combination of analytical DCSs and numerical tables of total cross sections. The adopted DCSs are defined by simple, but physically sound analytical forms. The corresponding total cross sections are obtained by a single numerical quadrature that is performed very quickly using the **SUMGA** external function described in Appendix B. Moreover, the random sampling from these DCSs can be done analytically and, hence, exactly. Only coherent scattering requires a simple preparatory numerical step.

It may be argued that using analytical *approximate* DCSs, instead of more accurate tabulated DCSs implies a certain loss of accuracy. To minimise this loss, PENELOPE renormalises the analytical DCSs so as to reproduce total cross sections that are read from the input material data file. As a consequence, the free path between events and the kind of interaction are sampled using total cross sections that are nominally exact; approximations are introduced only in the description of individual interaction events.

In the following, κ stands for the photon energy in units of the electron rest energy, *i.e.*,

$$\kappa \equiv \frac{E}{m_e c^2}. \quad (2.1)$$

2.1 Coherent (Rayleigh) scattering

Coherent or Rayleigh scattering is the process by which photons are scattered by bound atomic electrons without excitation of the target atom, *i.e.*, the energies of the incident and scattered photons are the same. The scattering is qualified as “coherent” because it arises from the interference between secondary electromagnetic waves coming from different parts of the atomic charge distribution.

The atomic DCS per unit solid angle for coherent scattering is given approximately by (see, *e.g.*, Born, 1969)

$$\frac{d\sigma_{\text{Ra}}}{d\Omega} = \frac{d\sigma_{\text{T}}}{d\Omega} [F(q, Z)]^2, \quad (2.2)$$

where

$$\frac{d\sigma_T(\theta)}{d\Omega} = r_e^2 \frac{1 + \cos^2 \theta}{2} \quad (2.3)$$

is the classical Thomson DCS for scattering by a free electron at rest, θ is the polar scattering angle (see Fig. 2.1) and $F(q, Z)$ is the atomic form factor. The quantity r_e is the classical electron radius and q is the magnitude of the momentum transfer. Since the incident and scattered photons have linear momentum $p = E/c$, we have

$$q = |\mathbf{p} - \mathbf{p}'| = 2(E/c) \sin(\theta/2) = (E/c) [2(1 - \cos \theta)]^{1/2}. \quad (2.4)$$

In the literature on x-ray crystallography, the dimensionless variable

$$x \equiv \frac{q \cdot 10^{-8} \text{cm}}{4\pi\hbar} = 20.6074 \frac{q}{m_e c} \quad (2.5)$$

is normally used instead of q .

The atomic form factor can be expressed as the Fourier transform of the atomic electron density $\rho(\mathbf{r})$ which, for a spherically symmetrical atom, simplifies to

$$F(q, Z) = 4\pi \int_0^\infty \rho(r) \frac{\sin(qr/\hbar)}{qr/\hbar} r^2 dr. \quad (2.6)$$

$F(q, Z)$ is a monotonically decreasing function of q that varies from $F(0, Z) = Z$ to $F(\infty, Z) = 0$. The most accurate form factors are those obtained from Hartree-Fock or configuration-interaction atomic-structure calculations; here we adopt the non-relativistic atomic form factors tabulated by Hubbell *et al.* (1975). Although relativistic form factors are available (Doyle and Turner, 1968), Hubbell has pointed out that the non-relativistic form factors yield results in closer agreement with experiment (Cullen *et al.*, 1997).

In the calculations, we use the following analytical approximation (Baró *et al.*, 1994a)

$$F(q, Z) = \begin{cases} f(x, Z) \equiv Z \frac{1 + a_1 x^2 + a_2 x^3 + a_3 x^4}{(1 + a_4 x^2 + a_5 x^4)^2}, \\ \max \{f(x, Z), F_K(q, Z)\} & \text{if } Z > 10 \text{ and } f(x, Z) < 2, \end{cases} \quad (2.7)$$

where

$$F_K(q, Z) \equiv \frac{\sin(2b \arctan Q)}{bQ (1 + Q^2)^b}, \quad (2.8)$$

with

$$Q = \frac{q}{2m_e c a}, \quad b = \sqrt{1 - a^2}, \quad a \equiv \alpha(Z - 5/16), \quad (2.9)$$

where α is the fine-structure constant. The function $F_K(q, Z)$ is the contribution to the atomic form factor due to the two K-shell electrons. The parameters of expression $f(x, Z)$ for $Z = 1$ to 99, which have been determined by numerically fitting the atomic

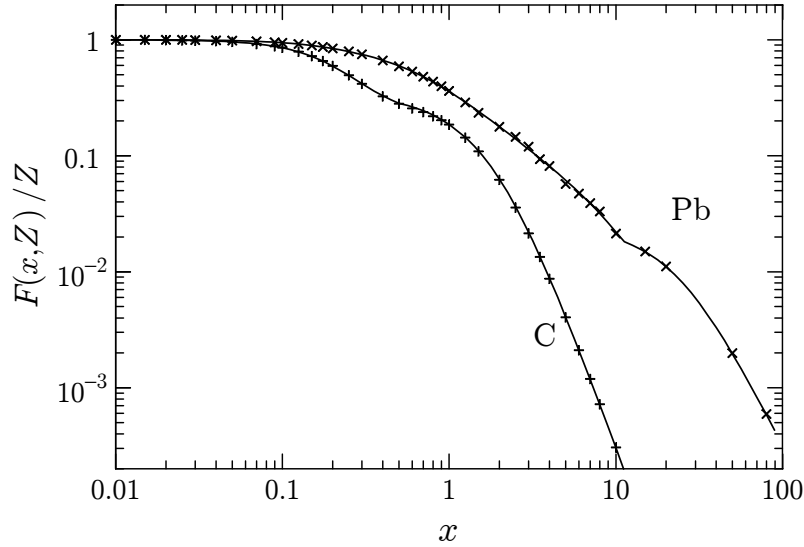


Figure 2.2: Atomic form factors for carbon and lead. Crosses are values from the tables of Hubbell *et al.* (1975), continuous curves represent the analytical approximation given by Eq. (2.7).

form factors tabulated by Hubbell *et al.* (1975), are included in the block data subprogram PENDAT. The average relative difference between the analytical and tabulated form factors is less than 0.5% (see Fig. 2.2).

The total coherent scattering cross section per atom is

$$\sigma_{\text{Ra}} = \int \frac{d\sigma_{\text{Ra}}}{d\Omega} d\Omega = \pi r_e^2 \int_{-1}^1 (1 + \cos^2 \theta) [F(q, Z)]^2 d(\cos \theta). \quad (2.10)$$

Introducing q , Eq. (2.4), as a new integration variable, the asymptotic behaviour of the total cross section for small and large photon energies is made clear. For low photon energies, the form factor in the integrand does not depart appreciably from the value $F(0, Z) = Z$, *i.e.*, coherent scattering reduces to pure Thomson scattering. Consequently, we have

$$\sigma_{\text{Ra}} \simeq \frac{8}{3} \pi r_e^2 Z^2. \quad (2.11)$$

In the high-energy limit, we get

$$\sigma_{\text{Ra}} \propto E^{-2}. \quad (2.12)$$

In practice, this limiting behaviour is attained for energies of the order of $Z/2$ MeV.

Strictly speaking, expression (2.2) is adequate only for photons with energy well above the K absorption edge. The low-energy behaviour given by Eq. (2.11) is substantially altered when anomalous scattering factors are introduced (see, *e.g.*, Cullen *et al.*, 1989; Kane *et al.*, 1986). These factors lead to a general decrease of the coherent scattering cross section near the absorption edges and at low energies. Nevertheless,

at the energies where anomalous scattering effects become significant, coherent scattering is much less probable than photoelectric absorption (see Fig. 2.10 below), and the approximation given by Eq. (2.2) is usually sufficient for simulation purposes.

2.1.1 Simulation of coherent scattering events

The PDF of the angular deflection, $\cos \theta$, can be written as [see Eqs. (2.2) and (2.3); normalisation is irrelevant here]

$$p_{\text{Ra}}(\cos \theta) = \frac{1 + \cos^2 \theta}{2} [F(x, Z)]^2, \quad (2.13)$$

where x , which is defined by Eqs. (2.4) and (2.5), can take values in the interval from 0 to

$$x_{\text{max}} = 20.6074 \times 2\kappa. \quad (2.14)$$

This PDF can be factorised in the form

$$p_{\text{Ra}}(\cos \theta) = g(\cos \theta) \pi(x^2) \quad (2.15)$$

with

$$g(\cos \theta) \equiv \frac{1 + \cos^2 \theta}{2} \quad \text{and} \quad \pi(x^2) \equiv [F(x, Z)]^2. \quad (2.16)$$

Notice that, for a compound, $[F(x, Z)]^2$ has to be replaced by the sum of squared form factors of the atoms in the molecule.

The function $\pi(x^2)$ can be considered as the (unnormalised) PDF of the variable x^2 , which takes values in the interval $(0, x_{\text{max}}^2)$. In PENELOPE, random values of x^2 distributed according to this PDF are generated by using the RITA algorithm (see Section 1.2.4) with 128 grid points.

The angular deflection $\cos \theta$ can now be sampled by the rejection method (Section 1.2.5), because the function $g(\cos \theta)$ is a valid rejection function (*i.e.*, it is positive and less than or equal to unity). The algorithm for sampling $\cos \theta$ proceeds as follows:

- (i) Using the RITA algorithm, sample a random value of x^2 from the distribution $\pi(x^2)$, restricted to the interval $(0, x_{\text{max}}^2)$.

- (ii) Set

$$\cos \theta = 1 - \frac{1}{2} \frac{x^2}{(20.6074\kappa)^2}. \quad (2.17)$$

- (iii) Generate a new random number ξ .

- (iv) If $\xi > g(\cos \theta)$, go to step (ii).

- (v) Deliver $\cos \theta$.

It is worth noting that this sampling method is essentially independent of the adopted form factor, and is directly applicable to molecules. The efficiency of the algorithm (*i.e.*, the fraction of generated values of $\cos\theta$ that is accepted) increases with photon energy. At low energies, it equals 2/3 (exactly) for all elements. For $E = 100$ keV, the efficiencies for hydrogen and uranium are 100% and 86%, respectively.

2.2 Photoelectric effect

In the photoelectric effect, a photon of energy E is absorbed by the target atom, which makes a transition to an excited state. The photon beams found in radiation transport studies have relatively low photon densities and, as a consequence, only single-photon absorption is observed¹. To represent the atomic states, we can adopt an independent-electron model, such as the Dirac-Hartree-Fock-Slater self-consistent model (see, *e.g.*, Pratt *et al.*, 1973), in which each electron occupies a single-particle orbital, with well-defined ionisation energy. The set of orbitals with the same principal and total angular momentum quantum numbers and the same parity constitute a shell. Each shell i can accommodate a finite number of electrons, with characteristic ionisation energy U_i . Notice that the shell ionisation energies are positive; the quantity $-U_i$ represents the “binding” energy of each individual electron. Figure 2.3 (left diagram) shows the various notations used to designate the innermost atomic electron shells (*i.e.*, those with the largest ionisation energies) as well as their ordering in energy and allowed occupancies. In our simulations, we use the experimental ionisation energies given by Lederer and Shirley (1978), which pertain to free, neutral atoms. Ionisation energies of K-, L- and M-shells are displayed in Fig. 2.4.

Considering the interaction with the photon field as a first-order perturbation (which is appropriate for fields with low photon densities) it follows that only one-electron transitions are allowed. That is, in the photoelectric effect, the photon is absorbed by an individual electron in the “active” shell i , which leaves the parent atom with kinetic energy $E_e = E - U_i$. Evidently, photoionisation of a given shell is only possible when the photon energy exceeds the corresponding ionisation energy; this gives rise to the characteristic absorption edges in the photoelectric cross section (see Fig. 2.5).

The photoelectric cross sections used in PENELOPE are obtained by interpolation in a numerical table that was extracted from the LLNL Evaluated Photon Data Library (EPDL; Cullen *et al.*, 1997). This library contains photoelectric cross sections for all shells of the elements $Z = 1 - 100$ and photon energies from 1 eV to 1000 GeV, derived from Scofield’s theoretical calculations of shell cross sections (Saloman *et al.*, 1988) and Hubbell’s total cross sections (Hubbell *et al.*, 1980; Berger and Hubbell, 1987). The PENELOPE database for photoelectric absorption (a subset of the EPDL) consists of tables of the total atomic cross section $\sigma_{\text{ph}}(E)$ and the cross sections for the K, L and M shells, $\sigma_{\text{ph},i}(E)$ ($i = \text{K, L1, L2, L3, and M1 to M5}$) for the elements $Z = 1 - 99$,

¹In intense low-energy photon beams, such as those from high-power lasers, simultaneous absorption of several photons is possible.

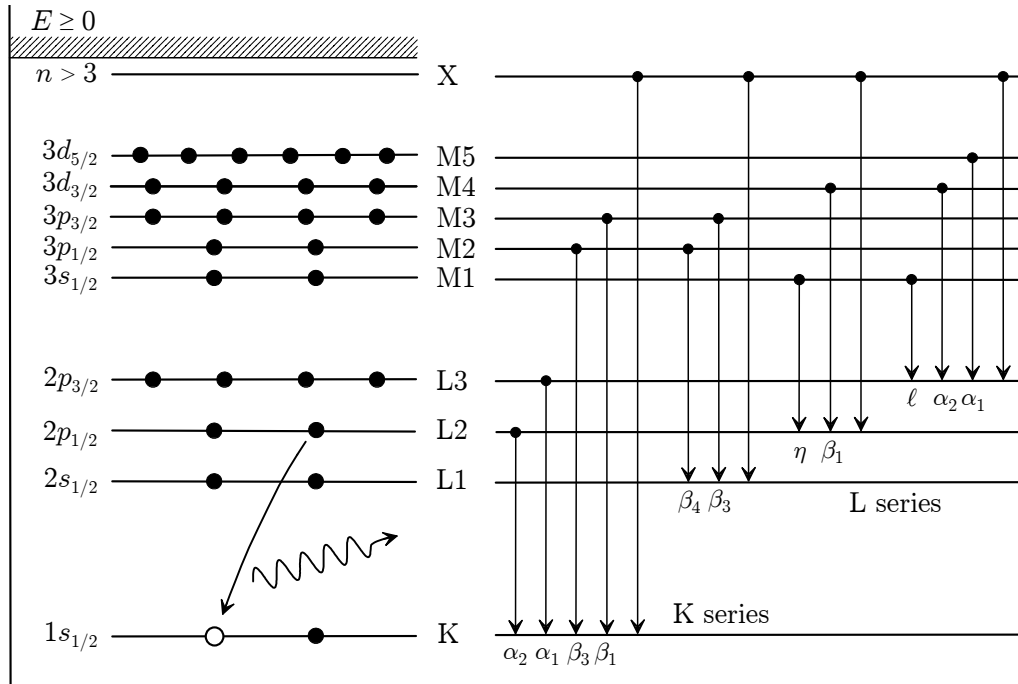


Figure 2.3: Various notations for inner atomic electron shells (left) and allowed radiative transitions (right) to these shells. Transitions that are different from the ones indicated in the diagram (*e.g.*, K-M4) are also possible, but their transition probabilities are extremely small.

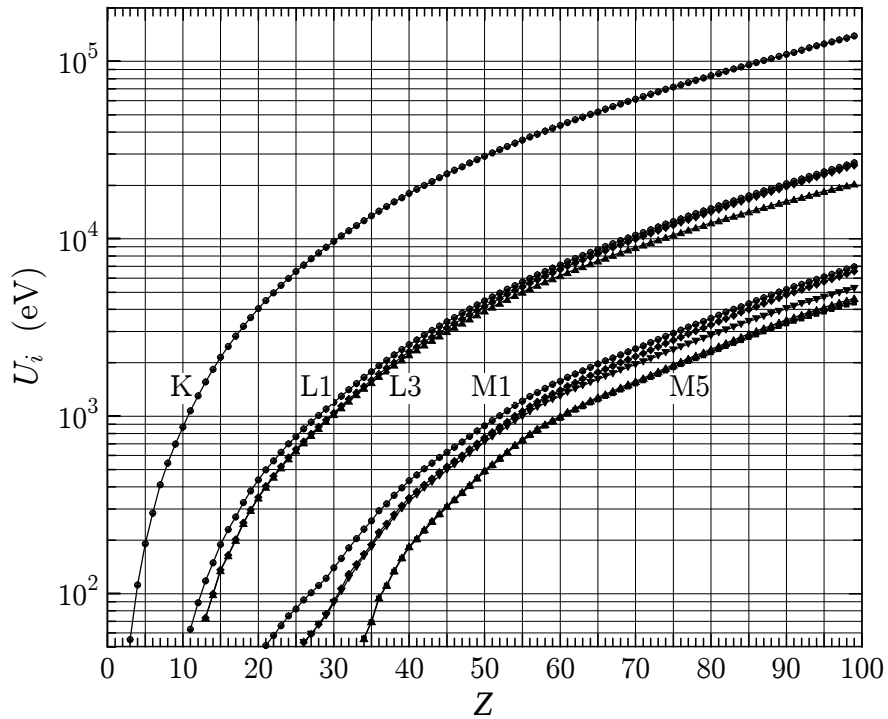


Figure 2.4: Ionisation energies of the innermost shells of free atoms, as given by Lederer and Shirley (1978).

which span the energy range from 50 eV to 1000 GeV. These tables are estimated to be accurate to within a few percent for photon energies above 1 keV (Cullen *et al.*, 1997). At lower energies, uncertainties in the data are much larger, of the order of 10–20% for $0.5 \text{ keV} < E < 1 \text{ keV}$, 100–200% for $0.1 \text{ keV} < E < 0.5 \text{ keV}$, and 1000% for $E < 100 \text{ eV}$. Notice that the cross sections in the EPDL are based on free-atom theoretical calculations and, therefore, near-edge absorption structures produced by molecular or crystalline ordering (*e.g.*, extended x-ray absorption fine-structure) are ignored.

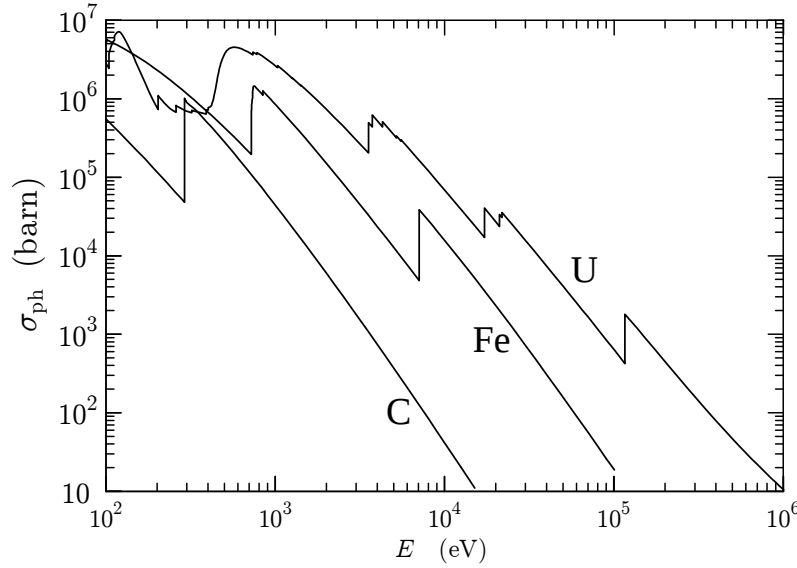


Figure 2.5: Atomic photoelectric cross sections for carbon, iron and uranium as functions of the photon energy E .

For compound materials (and also for mixtures) the molecular cross section $\sigma_{\text{ph}}(E)$ is evaluated by means of the additivity approximation, that is, as the sum of the atomic cross sections of the elements involved. In the energy range between successive absorption edges, the photoelectric cross section is a continuous function of the photon energy (see Fig. 2.5). In PENELOPE, the molecular cross section is defined by means of a table of numerical values $\sigma_{\text{ph}}(E_i)$ for a logarithmic grid of energies E_i , which is stored in memory. Photon mean free paths are determined by linear log-log interpolation in this table. Knowledge of the atomic cross sections is needed only when a photoabsorption event has effectively occurred in order to select the element that has been ionised (whose probability is proportional to the atomic cross section).

2.2.1 Simulation of photoelectron emission

Let us consider that a photon with energy E is absorbed by an atom of the element Z . The “active” shell i that is ionised is considered as a discrete random variable with PDF

$$p_i = \sigma_{\text{ph},i}(Z, E) / \sigma_{\text{ph}}(Z, E), \quad (2.18)$$

where $\sigma_{\text{ph},i}(Z, E)$ is the cross section for ionisation of shell i and $\sigma_{\text{ph}}(Z, E)$ is the total photoelectric cross section of the atom. PENELOPE incorporates a detailed description of photoabsorption in K-, L- and M-shells (including the subsequent atomic relaxation). The ionisation probabilities of these inner shells are determined from the corresponding partial cross sections. The probability of ionisation in outer shells is obtained as

$$p_{\text{outer}} = 1 - p_{\text{K}} - p_{\text{L1}} - \dots - p_{\text{M5}}. \quad (2.19)$$

When the ionisation occurs in an inner K-, L- or M-shell, the initial energy of the photoelectron is set equal to $E_e = E - U_i$; the residual atom, with a vacancy in the shell, subsequently relaxes to its ground state by emitting x rays and Auger electrons. If the ionisation occurs in an outer shell, we assume that the photoelectron leaves the target atom with kinetic energy equal to the energy deposited by the photon, $E_e = E$, and we disregard the emission of subsidiary fluorescent radiation (see Section 2.6).

2.2.1.1 Initial direction of photoelectrons

The direction of emission of the photoelectron, relative to that of the absorbed photon, is defined by the polar and azimuthal angles θ_e (Fig. 2.1) and ϕ_e . We consider that the incident photon is not polarised and, hence, the angular distribution of photoelectrons is independent of ϕ_e , which is uniformly distributed in the interval $(0, 2\pi)$. The polar angle θ_e is sampled from the K-shell cross section derived by Sauter (1931) using K-shell hydrogenic electron wave functions. The Sauter DCS (per electron) can be written as

$$\frac{d\sigma_{\text{ph}}}{d\Omega_e} = \alpha^4 r_e^2 \left(\frac{Z}{\kappa} \right)^5 \frac{\beta^3}{\gamma} \frac{\sin^2 \theta_e}{(1 - \beta \cos \theta_e)^4} \left[1 + \frac{1}{2} \gamma(\gamma - 1)(\gamma - 2)(1 - \beta \cos \theta_e) \right], \quad (2.20)$$

where α is the fine-structure constant, r_e is the classical electron radius, and

$$\gamma = 1 + E_e/(m_e c^2), \quad \beta = \frac{\sqrt{E_e(E_e + 2m_e c^2)}}{E_e + m_e c^2}. \quad (2.21)$$

Strictly speaking, the DCS (2.20) is adequate only for ionisation of the K-shell by high-energy photons. Nevertheless, in many practical simulations no appreciable errors are introduced when Sauter's distribution is used to describe any photoionisation event, irrespective of the atomic shell and the photon energy. The main reason is that the emitted photoelectron immediately starts to interact with the medium, and its direction of movement is strongly altered after travelling a path length much shorter than the photon mean free path. On the other hand, when the photon energy exceeds the K-edge, most of the ionisations occur in the K-shell and then the Sauter distribution represents a good approximation.

Introducing the variable $\nu = 1 - \cos \theta_e$, the angular distribution of photoelectrons can be expressed in the form

$$p(\nu) = (2 - \nu) \left[\frac{1}{A + \nu} + \frac{1}{2} \beta \gamma(\gamma - 1)(\gamma - 2) \right] \frac{\nu}{(A + \nu)^3}, \quad A = \frac{1}{\beta} - 1, \quad (2.22)$$

apart from a normalisation constant. Random sampling of ν from this distribution can be performed analytically. To this end, $p(\nu)$ can be factorised in the form

$$p(\nu) = g(\nu)\pi(\nu) \quad (2.23)$$

with

$$g(\nu) = (2 - \nu) \left[\frac{1}{A + \nu} + \frac{1}{2}\beta\gamma(\gamma - 1)(\gamma - 2) \right] \quad (2.24)$$

and

$$\pi(\nu) = \frac{A(A + 2)^2}{2} \frac{\nu}{(A + \nu)^3}. \quad (2.25)$$

The variable ν takes values in the interval (0,2), where the function $g(\nu)$ is definite positive and attains its maximum value at $\nu = 0$, while the function $\pi(\nu)$ is positive and normalised to unity. Random values from the probability distribution $\pi(\nu)$ are generated by means of the sampling formula (inverse-transform method, see Section 1.2.2)

$$\int_0^\nu \pi(\nu') d\nu' = \xi, \quad (2.26)$$

which can be solved analytically to give

$$\nu = \frac{2A}{(A + 2)^2 - 4\xi} [2\xi + (A + 2)\xi^{1/2}]. \quad (2.27)$$

Therefore, random sampling from Sauter's distribution can be performed by the rejection method (see Section 1.2.5) as follows:

- (i) Generate ν from $\pi(\nu)$ by using Eq. (2.27).
- (ii) Generate a random number ξ .
- (iii) If $\xi g(0) > g(\nu)$, go to step (i).
- (iv) Deliver $\cos \theta_e = 1 - \nu$.

The efficiency of this algorithm is ~ 0.33 at low energies and increases slowly with E_e ; for $E_e = 1$ MeV, the efficiency is 0.4. As photoelectric absorption occurs at most once in each photon history, this small sampling efficiency does not slow down the simulation significantly.

2.3 Incoherent (Compton) scattering

In Compton scattering, a photon of energy E interacts with an atomic electron, which absorbs it and re-emits a secondary (Compton) photon of energy E' in the direction $\Omega = (\theta, \phi)$ relative to the direction of the original photon. In PENELOPE, Compton scattering

events are described by means of the cross section obtained from the relativistic impulse approximation (Ribberfors, 1983). Contributions from different atomic electron shells are considered separately. After a Compton interaction with the i -th shell, the active target electron is ejected to a free state with kinetic energy $E_e = E - E' - U_i > 0$, where U_i is the ionisation energy of the considered shell, and the residual atom is left in an excited state with a vacancy in the i -th shell.

In the case of scattering by free electrons at rest, the conservation of energy and momentum implies the following relation between the energy E' of the scattered (Compton) photon and the scattering angle θ [cf. Eq. (A.19)]

$$E' \equiv \frac{E}{1 + \kappa(1 - \cos \theta)} \equiv E_C, \quad (2.28)$$

where $\kappa = E/m_e c^2$, as before. The DCS for Compton scattering by a free electron at rest is given by the familiar Klein-Nishina formula,

$$\frac{d\sigma_{\text{Co}}^{\text{KN}}}{d\Omega} = \frac{r_e^2}{2} \left(\frac{E_C}{E} \right)^2 \left(\frac{E_C}{E} + \frac{E}{E_C} - \sin^2 \theta \right). \quad (2.29)$$

Although this simple DCS was generally used in old Monte Carlo transport codes, it represents only a rough approximation for the Compton interactions of photons with atoms. In reality, atomic electrons are not at rest, but move with a certain momentum distribution, which gives rise to the so-called Doppler broadening of the Compton line. Moreover, transitions of bound electrons are allowed only if the energy transfer $E - E'$ is larger than the ionisation energy U_i of the active shell (binding effect).

The impulse approximation accounts for Doppler broadening and binding effects in a natural, and relatively simple, way. The DCS is obtained by considering that electrons in the i -th shell move with a momentum distribution $\rho_i(\mathbf{p})$. For an electron in an orbital $\psi_i(\mathbf{r})$, $\rho_i(\mathbf{p}) \equiv |\psi_i(\mathbf{p})|^2$, where $\psi_i(\mathbf{p})$ is the wave function in the momentum representation. The DCS for Compton scattering by an electron with momentum $\mathbf{p}$ is derived from the Klein-Nishina formula by applying a Lorentz transformation with velocity $\mathbf{v}$ equal to that of the moving target electron. The impulse approximation to the Compton DCS (per electron) of the considered shell is obtained by averaging over the momentum distribution $\rho_i(\mathbf{p})$.

After some manipulations, the Compton DCS of an electron in the i -th shell can be expressed as [Eq. (21) in Brusa *et al.*, 1996]

$$\frac{d^2\sigma_{\text{Co},i}}{dE'd\Omega} = \frac{r_e^2}{2} \left(\frac{E_C}{E} \right)^2 \left(\frac{E_C}{E} + \frac{E}{E_C} - \sin^2 \theta \right) F(p_z) J_i(p_z) \frac{dp_z}{dE'}, \quad (2.30)$$

where r_e is the classical electron radius. E_C is the energy of the Compton line, defined by Eq. (2.28), *i.e.*, the energy of photons scattered in the direction θ by *free electrons at rest*. The momentum transfer vector is given by $\mathbf{q} \equiv \hbar\mathbf{k} - \hbar\mathbf{k}'$, where $\hbar\mathbf{k}$ and $\hbar\mathbf{k}'$ are the momenta of the incident and scattered photons; its magnitude is

$$q = \frac{1}{c} \sqrt{E^2 + E'^2 - 2EE' \cos \theta}. \quad (2.31)$$

The quantity p_z is the projection of the initial momentum $\mathbf{p}$ of the target electron on the direction of the scattering vector $\hbar\mathbf{k}' - \hbar\mathbf{k} = -\mathbf{q}$; it is given by²

$$p_z \equiv -\frac{\mathbf{p} \cdot \mathbf{q}}{q} = \frac{EE'(1 - \cos \theta) - m_e c^2(E - E')}{c^2 q} \quad (2.32)$$

or, equivalently,

$$\frac{p_z}{m_e c} = \frac{E(E' - E_C)}{E_C c q}. \quad (2.33)$$

Notice that $p_z = 0$ for $E' = E_C$. Moreover,

$$\frac{dp_z}{dE'} = \frac{m_e c}{c q} \left(\frac{E}{E_C} + \frac{E \cos \theta - E'}{c q} \frac{p_z}{m_e c} \right). \quad (2.34)$$

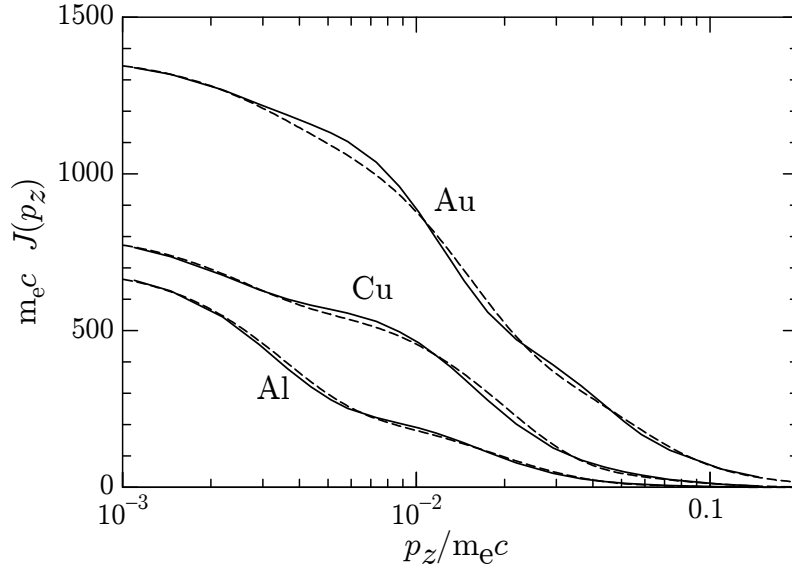


Figure 2.6: Atomic Compton profiles ($p_z > 0$) for aluminium, copper and gold. The continuous curves are numerical Hartree-Fock profiles tabulated by Biggs *et al.* (1975). The dashed curves represent the analytical profiles defined by Eq. (2.53). (Adapted from Brusa *et al.*, 1996.)

The function $J_i(p_z)$ in Eq. (2.30) is the one-electron Compton profile of the active shell, which is defined as

$$J_i(p_z) \equiv \iint \rho_i(\mathbf{p}) dp_x dp_y, \quad (2.35)$$

where $\rho_i(\mathbf{p})$ is the electron momentum distribution. That is, $J_i(p_z) dp_z$ gives the probability that the component of the electron momentum in the z -direction is in the interval

²The expression (2.32) contains an approximation; the exact relation is obtained by replacing the electron rest energy $m_e c^2$ in the numerator by the electron initial total energy, $\sqrt{(m_e c^2)^2 + (cp)^2}$.

$(p_z, p_z + dp_z)$. Notice that the normalisation

$$\int_{-\infty}^{\infty} J_i(p_z) dp_z = 1 \quad (2.36)$$

is assumed. In the Hartree-Fock approximation for closed-shell configurations, the momentum distribution of the electrons in an atomic shell, obtained by adding the contributions of the orbitals in that shell, is isotropic. For an isotropic distribution, expression (2.35) simplifies to

$$J_i(p_z) = 2\pi \int_{|p_z|}^{\infty} p \rho_i(p) dp. \quad (2.37)$$

The atomic Compton profile is given by

$$J(p_z) = \sum_i f_i J_i(p_z), \quad (2.38)$$

where f_i is the number of electrons in the i -th shell and $J_i(p_z)$ is the one-electron profile of this shell. The functions $J(p_z)$ and $J_i(p_z)$ are both bell-shaped and symmetrical about $p_z = 0$ (see Fig. 2.6). Extensive tables of Hartree-Fock Compton profiles for the elements have been published by Biggs *et al.* (1975). These numerical profiles are adequate for bound electron shells. In the case of conductors, the one-electron Compton profile for conduction electrons may be estimated by assuming that these form a free-electron gas with ρ_e electrons per unit volume. The one-electron profile for this system is (see, *e.g.*, Cooper, 1971)

$$J_i^{\text{feg}}(p_z) = \frac{3}{4p_F} \left(1 - \frac{p_z^2}{p_F^2} \right) \Theta(p_F - |p_z|), \quad J_i^{\text{feg}}(0) = \frac{3}{4p_F}, \quad (2.39)$$

where $p_F \equiv \hbar(3\pi^2\rho_e)^{1/3}$ is the Fermi momentum. For scattering in a compound material, the molecular Compton profile is obtained as the sum of atomic profiles of the atoms in a molecule (additivity rule).

The factor $F(p_z)$ in Eq. (2.30) is approximately given by

$$F(p_z) \simeq 1 + \frac{cq_C}{E} \left(1 + \frac{E_C(E_C - E \cos \theta)}{(cq_C)^2} \right) \frac{p_z}{m_e c}, \quad (2.40)$$

where q_C is the momentum transfer associated with the energy $E' = E_C$ of the Compton line,

$$q_C \equiv \frac{1}{c} \sqrt{E^2 + E_C^2 - 2EE_C \cos \theta}. \quad (2.41)$$

Expression (2.40) is accurate only for small $|p_z|$ -values. For large $|p_z|$, $J_i(p_z)$ tends to zero and the factor $F(p_z)$ has no effect on the DCS. We use the values given by expression (2.40) only for $|p_z| < 0.2m_e c$ and take $F(\pm|p_z|) = F(\pm 0.2m_e c)$ for $|p_z| > 0.2m_e c$. Owing to the approximations introduced, negative values of F may be obtained for large $|p_z|$; in this case, we must set $F = 0$.

We can now introduce the effect of electron binding: Compton excitations are allowed only if the target electron is promoted to a free state, *i.e.*, if the energy transfer $E - E'$ is larger than the ionisation energy U_i of the active shell. Therefore the atomic DCS, including Doppler broadening and binding effects, is given by

$$\begin{aligned} \frac{d^2\sigma_{\text{Co}}}{dE'd\Omega} &= \frac{r_e^2}{2} \left(\frac{E_C}{E} \right)^2 \left(\frac{E_C}{E} + \frac{E}{E_C} - \sin^2\theta \right) \\ &\times F(p_z) \left(\sum_i f_i J_i(p_z) \Theta(E - E' - U_i) \right) \frac{dp_z}{dE'}, \end{aligned} \quad (2.42)$$

where $\Theta(x)$ ($= 1$ if $x > 0$, $= 0$ otherwise) is the Heaviside step function. In the calculations we use the ionisation energies U_i given by Lederer and Shirley (1978), Fig. 2.4. The DCS for scattering of 10 keV photons by aluminium atoms is displayed in Fig. 2.7, for $\theta = 60$ and 180 deg, as a function of the fractional energy of the emerging photon. The DCS for a given scattering angle has a maximum at $E' = E_C$; its shape resembles that of the atomic Compton profile, except for the occurrence of edges at $E' = E - U_i$.

In the case of scattering by free electrons at rest we have $U_i = 0$ (no binding) and $J_i(p_z) = \delta(p_z)$ (no Doppler broadening). Moreover, from Eq. (2.33) $E' = E_C$, so that photons scattered through an angle θ have energy E_C . Integration of the DCS, Eq. (2.42), over E' then yields the familiar Klein-Nishina cross section,

$$\frac{d\sigma_{\text{Co}}^{\text{KN}}}{d\Omega} = Z \frac{r_e^2}{2} \left(\frac{E_C}{E} \right)^2 \left(\frac{E_C}{E} + \frac{E}{E_C} - \sin^2\theta \right), \quad (2.43)$$

for the Z atomic electrons [cf. Eq. (2.29)]. For energies of the order of a few MeV and larger, Doppler broadening and binding effects are relatively small and the free-electron theory yields results practically equivalent to those of the impulse approximation.

The angular distribution of scattered photons is given by the directional DCS,

$$\begin{aligned} \frac{d\sigma_{\text{Co}}}{d\Omega} &= \int \frac{d^2\sigma_{\text{Co}}}{dE'd\Omega} dE' = \frac{r_e^2}{2} \left(\frac{E_C}{E} \right)^2 \left(\frac{E_C}{E} + \frac{E}{E_C} - \sin^2\theta \right) \\ &\times \sum_i f_i \Theta(E - U_i) \int_{-\infty}^{p_{i,\text{max}}} F(p_z) J_i(p_z) dp_z, \end{aligned} \quad (2.44)$$

where $p_{i,\text{max}}$ is the highest p_z -value for which an electron in the i -th shell can be excited. It is obtained from Eq. (2.32) by setting $E' = E - U_i$,

$$p_{i,\text{max}}(E, \theta) = \frac{E(E - U_i)(1 - \cos\theta) - m_e c^2 U_i}{c \sqrt{2E(E - U_i)(1 - \cos\theta) + U_i^2}}. \quad (2.45)$$

Except for energies just above the shell ionisation threshold, the function $F(p_z)$ in the integral can be replaced by unity, since $p_z J_i(p_z)$ is an odd function and its integral is close to zero, *i.e.*,

$$\int_{-\infty}^{p_{i,\text{max}}} F(p_z) J_i(p_z) dp_z \simeq n_i(p_{i,\text{max}}), \quad (2.46)$$

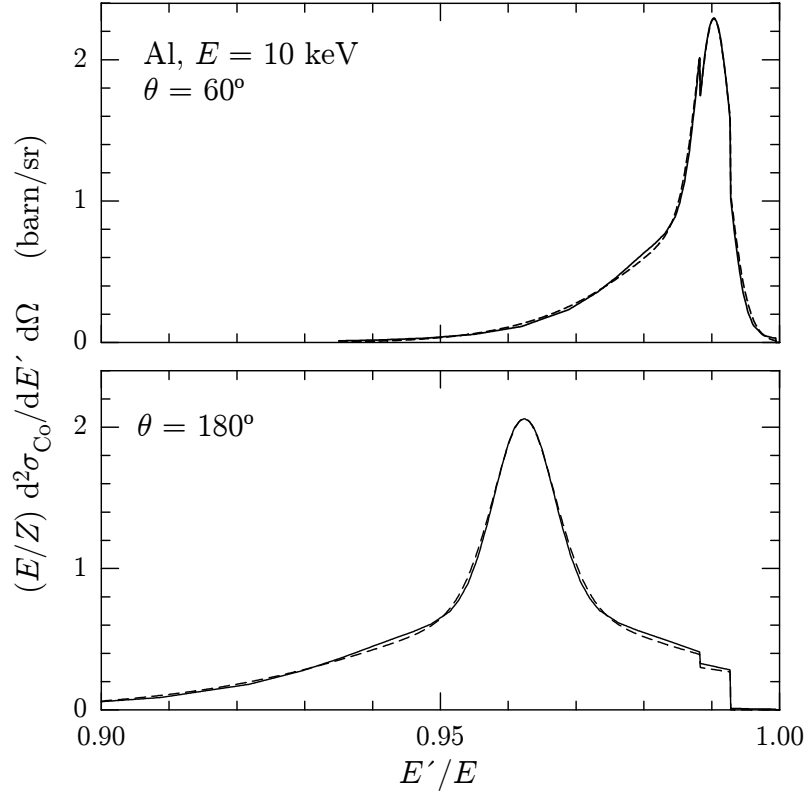


Figure 2.7: DCS for Compton scattering of 10 keV photons by aluminium atoms at the indicated scattering angles. The continuous curves represent the DCS (2.42) calculated using the Hartree-Fock Compton profile (Biggs *et al.*, 1975). The dashed curves are results from Eq. (2.42) with the analytical profiles given by Eq. (2.53). (Adapted from Brusa *et al.*, 1996.)

where

$$n_i(p_z) \equiv \int_{-\infty}^{p_z} J_i(p'_z) dp'_z. \quad (2.47)$$

Notice that $n_i(p_z)$ is a monotonously increasing function of p_z , which varies from 0 at $p_z = -\infty$ to unity at $p_z = \infty$; the quantity $n_i(p_{i,\max})$ represents the fraction of electrons in the i -th shell that can be effectively excited in a Compton interaction. We can then write

$$\frac{d\sigma_{\text{Co}}}{d\Omega} \simeq \frac{r_e^2}{2} \left(\frac{E_C}{E} \right)^2 \left(\frac{E_C}{E} + \frac{E}{E_C} - \sin^2 \theta \right) S(E, \theta). \quad (2.48)$$

The function

$$S(E, \theta) = \sum_i f_i \Theta(E - U_i) n_i(p_{i,\max}) \quad (2.49)$$

can be identified as the incoherent scattering function in the impulse approximation (see, *e.g.*, Ribberfors and Berggren, 1982). The total cross section can then be obtained as

$$\sigma_{\text{Co}} = 2\pi \int_{-1}^1 \frac{d\sigma_{\text{Co}}}{d\Omega} d(\cos \theta). \quad (2.50)$$

For comparison purposes, and also to calculate the energy deposition, it is useful to consider the cross section differential in only the energy of the scattered photon,

$$\frac{d\sigma_{\text{Co}}}{dE'} \equiv \int \frac{d^2\sigma_{\text{Co}}}{dE' d\Omega} d\Omega. \quad (2.51)$$

In the case of scattering by free electrons at rest, $E' = E_C$ and the Klein-Nishina formula (2.43) gives the following expression for the energy DCS,

$$\begin{aligned} \frac{d\sigma_{\text{Co}}^{\text{KN}}}{dE'} &= 2\pi \frac{d\sigma_{\text{Co}}^{\text{KN}}}{d\Omega} \frac{d(\cos\theta)}{dE_C} \\ &= \frac{\pi r_e^2}{E} \kappa^{-3} \left(\frac{E^2}{E'^2} + \frac{(\kappa^2 - 2\kappa - 2)E}{E'} + (2\kappa + 1) + \frac{\kappa^2 E'}{E} \right). \end{aligned} \quad (2.52)$$

Fig. 2.8 displays the energy DCSs obtained from this formula and from the impulse approximation for scattering of high-energy ($E > U_i$) photons by aluminium and gold atoms. These results clearly show the differences between the physics of the impulse approximation and the cruder free-electron approximation. The most conspicuous feature of the impulse approximation DCS is the absence of a threshold energy, which is a direct manifestation of the Doppler broadening. For relatively small energy transfers ($E' \sim E$) the Klein-Nishina DCS increases with the energy of the scattered photon, whereas the energy DCS obtained from the impulse approximation vanishes at $E' = E$ due to the effect of binding, which also causes the characteristic edge structure, similar to that of the photoelectric cross section (see Fig. 2.8).

2.3.1 Analytical Compton profiles

In order to simplify the random sampling and to minimise the required numerical information, we use approximate one-electron profiles of the form

$$J_i^A(p_z) = J_{i,0} \frac{nd_2}{2} \left(d_1 + d_2 J_{i,0} |p_z| \right)^{n-1} \exp \left[d_1^n - \left(d_1 + d_2 J_{i,0} |p_z| \right)^n \right] \quad (2.53)$$

with

$$n = 2, \quad d_1 = \left(\frac{n-1}{n} \right)^{1/n} = \sqrt{\frac{1}{2}}, \quad d_2 = \frac{2}{n} d_1^{1-n} = \sqrt{2}.$$

The quantity $J_{i,0} \equiv J_i(0)$ is the value of the profile at $p_z = 0$ obtained from the Hartree-Fock orbital (Biggs *et al.*, 1975). $J_i(0)$ is tabulated in the file `pdatconf.p06` for all shells of the elements $Z = 1$ to 99. Notice that $J_i^A(p_z)$ is normalised according to Eq. (2.36). With the profiles (2.53),

$$n_i^A(p_z) \equiv \int_{-\infty}^{p_z} J_i^A(p'_z) dp'_z = \begin{cases} \frac{1}{2} \exp \left[d_1^2 - \left(d_1 - d_2 J_{i,0} p_z \right)^2 \right] & \text{if } p_z < 0, \\ 1 - \frac{1}{2} \exp \left[d_1^2 - \left(d_1 + d_2 J_{i,0} p_z \right)^2 \right] & \text{if } p_z > 0. \end{cases} \quad (2.54)$$

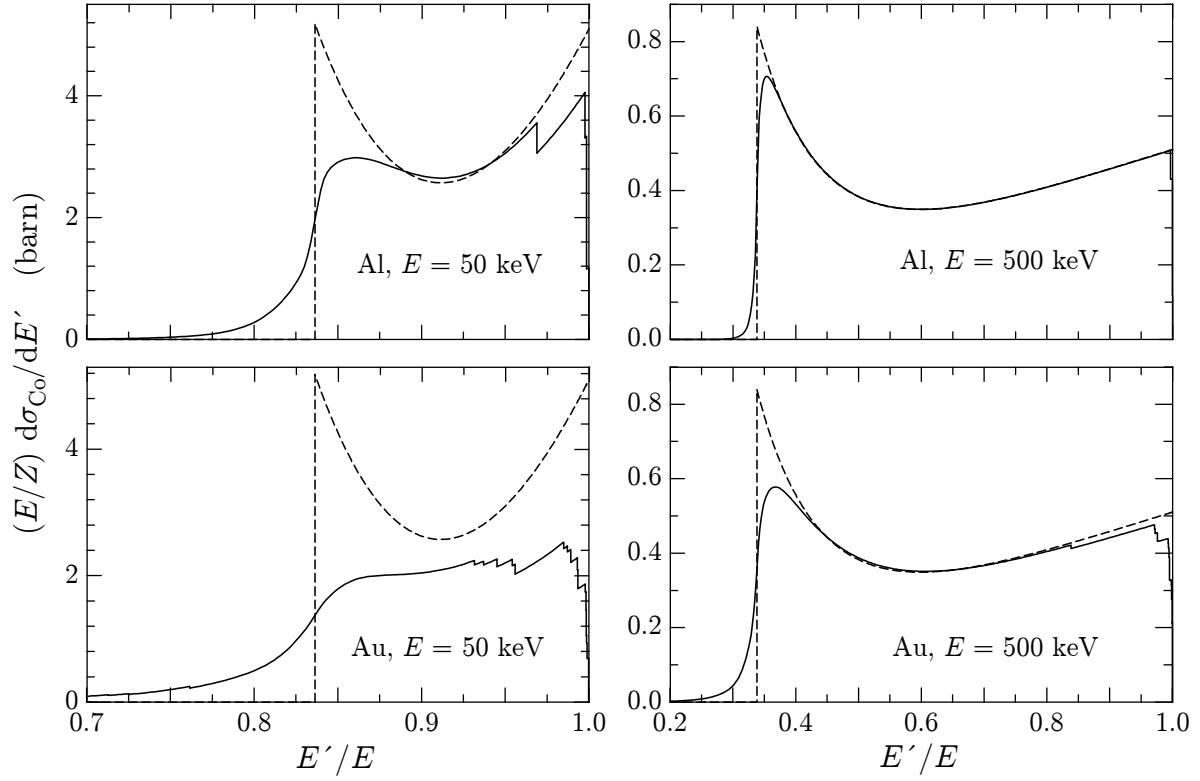


Figure 2.8: Energy DCSs for Compton scattering of 50 and 500 keV photons by aluminium and gold atoms. The continuous curves represent the DCS (2.51), computed using the analytical Compton profiles (2.53). The dashed curves are obtained from the Klein-Nishina formula (2.52), *i.e.*, assuming that the atomic electrons are free and at rest.

Thus, the incoherent scattering function (2.49) can be expressed analytically and the integral (2.50) evaluated very quickly with the aid of function **SUMGA** (Appendix B). On the other hand, the sampling equation $n_i^A(p_z) \equiv \xi n_i^A(p_{i,\max})$ (see Section 1.2.2) can be solved analytically,

$$p_z = \begin{cases} \frac{1}{d_2 J_{i,0}} \left[d_1 - (d_1^2 - \ln 2A)^{1/2} \right] & \text{if } A < \frac{1}{2}, \\ \frac{1}{d_2 J_{i,0}} \left[(d_1^2 - \ln 2(1-A))^{1/2} - d_1 \right] & \text{if } A > \frac{1}{2}, \end{cases} \quad (2.55)$$

where $A \equiv \xi n_i^A(p_{i,\max})$. Atomic Compton profiles obtained from the approximation given by Eq. (2.53) are accurate for small p_z and oscillate about the Hartree-Fock values for intermediate momenta (see Fig. 2.6). The relative differences are normally less than 5%, except for large momenta for which $J(p_z)$ is very small. Similar differences are found between the DCS computed from Hartree-Fock and analytical Compton profiles (see Fig. 2.7). For most applications (*e.g.*, studies of detector response, dosimetry, radiotherapy, etc.), the effect of these differences on the simulation results is not important. The

impulse approximation with the analytical one-electron profiles (2.53) then provides a conveniently simple method to introduce Doppler broadening and binding effects in the simulation of Compton scattering.

In PENELOPE, the maximum number of electron shells for each material is limited. For heavy elements, and also for compounds, the number of shells may be fairly large. In this case, outer shells with similar ionisation energies are grouped together and replaced by a single shell with a $J_{i,0}$ value and an effective ionisation energy equal to the corresponding averages of the grouped shells. This grouping does not alter the average effects of Doppler broadening and binding.

2.3.2 Simulation of incoherent scattering events

Compton events are simulated on the basis of the DCS given by Eq. (2.42) with the analytical Compton profiles (2.53). The sampling algorithm adopted here is due to Brusa *et al.* (1996). It is similar to the one described by Namito *et al.* (1994), but has a higher efficiency.

The PDF of the polar deflection $\cos \theta$ and the energy E' of the scattered photon is given by (apart from normalisation constants, which are irrelevant here)

$$P_{\text{Co}}(\cos \theta, E') = \left(\frac{E_C}{E} \right)^2 \left(\frac{E_C}{E} + \frac{E}{E_C} - \sin^2 \theta \right) \times F(p_z) \left(\sum_i f_i J_i(p_z) \Theta(E - E' - U_i) \right) \frac{dp_z}{dE'}. \quad (2.56)$$

Integration of expression (2.56) over E' , using the approximation (2.46), yields the PDF of the polar deflection

$$P_\theta(\cos \theta) = \left(\frac{E_C}{E} \right)^2 \left(\frac{E_C}{E} + \frac{E}{E_C} - \sin^2 \theta \right) S(E, \theta), \quad (2.57)$$

where $S(E, \theta)$ is the incoherent scattering function, Eq. (2.49).

Random values of $\cos \theta$ from the PDF (2.57) can be generated by using the following algorithm (Baró *et al.*, 1994a). Let us introduce the quantity

$$\tau \equiv \frac{E_C}{E} = \frac{1}{1 + \kappa(1 - \cos \theta)}. \quad (2.58)$$

The minimum and maximum values of τ are

$$\tau_{\min} = \frac{1}{1 + 2\kappa} \quad \text{and} \quad \tau_{\max} = 1, \quad (2.59)$$

which correspond to backward ($\theta = \pi$) and forward ($\theta = 0$) scattering, respectively. The PDF of this variable is (again ignoring normalisation constants)

$$P_\tau(\tau) = P_\theta(\cos \theta) \frac{d(\cos \theta)}{d\tau} = \left(\frac{1}{\tau^2} + \frac{\kappa^2 - 2\kappa - 2}{\tau} + (2\kappa + 1) + \kappa^2 \tau \right) S(E, \theta). \quad (2.60)$$

This distribution can be rewritten in the form (Nelson *et al.*, 1985)

$$P_\tau(\tau) = [a_1 P_1(\tau) + a_2 P_2(\tau)] T(\cos \theta), \quad (2.61)$$

where

$$a_1 = \ln(1 + 2\kappa), \quad a_2 = \frac{2\kappa(1 + \kappa)}{(1 + 2\kappa)^2}, \quad (2.62)$$

$$P_1(\tau) = \frac{1}{\ln(1 + 2\kappa)} \frac{1}{\tau}, \quad P_2(\tau) = \frac{(1 + 2\kappa)^2}{2\kappa(1 + \kappa)} \tau \quad (2.63)$$

and

$$T(\cos \theta) = \left\{ 1 - \frac{(1 - \tau)[(2\kappa + 1)\tau - 1]}{\kappa^2 \tau(1 + \tau^2)} \right\} \frac{S(E, \theta)}{S(E, \theta = \pi)}. \quad (2.64)$$

The function in braces is positive, it equals 1 at the end points of the interval $(\tau_{\min}, 1)$, and is less than unity inside this interval. Moreover, the ratio of incoherent scattering functions is also less than unity for any value of $\theta < \pi$. Hence, the function $T(\cos \theta)$ is a valid rejection function. The functions $P_i(\tau)$ ($i = 1, 2$) are normalised PDFs in the interval $(\tau_{\min}, 1)$, which can be easily sampled by using the inverse-transform method. The generation of random values of τ according to the PDF given by Eq. (2.60) can then be performed by combining the composition and rejection methods (Section 1.2). The algorithm to sample $\cos \theta$ proceeds as follows:

- (i) Sample a value of the integer i ($=1, 2$) according to the point probabilities

$$\pi(1) = \frac{a_1}{a_1 + a_2} \quad \text{and} \quad \pi(2) = \frac{a_2}{a_1 + a_2}. \quad (2.65)$$

- (ii) Sample τ from $P_i(\tau)$ using the sampling formulae

$$\tau = \begin{cases} \tau_{\min}^\xi & \text{if } i = 1, \\ [\tau_{\min}^2 + \xi(1 - \tau_{\min}^2)]^{1/2} & \text{if } i = 2, \end{cases} \quad (2.66)$$

which can be easily derived by the inverse-transform method (Section 1.2.2).

- (iii) Determine $\cos \theta$ using Eq. (2.58),

$$\cos \theta = 1 - \frac{1 - \tau}{\kappa \tau}, \quad (2.67)$$

and compute the quantities $p_{i,\max}(E, \theta)$, Eq. (2.45), and

$$S(E, \theta) = \sum_i f_i \Theta(E - U_i) n_i^A(p_{i,\max}). \quad (2.68)$$

- (iv) Generate a new random number ξ .

- (v) If $\xi > T(\cos \theta)$, go to step (i).

(vi) Deliver $\cos \theta$.

The efficiency of this algorithm, *i.e.*, the probability of accepting a generated $\cos \theta$ -value, increases monotonically with photon energy and is nearly independent of Z ; typical values are 35%, 80% and 95% for $E = 1$ keV, 1 MeV and 10 MeV, respectively.

Once the direction of the emerging photon has been set, the active electron shell i is selected with relative probability equal to $Z_i \Theta(E - U_i) n_i^A(p_{i,\max}(E, \theta))$. A random value of p_z is generated from the analytical Compton profile (2.53) using the sampling formula (2.55). If p_z is less than $-\mathbf{m}_e c$, it is rejected and a new shell and a p_z -value are sampled³. Finally, the factor $F(p_z)$ in the PDF (2.42) is accounted for by means of a rejection procedure. It should be noted that the approximation $F \simeq 1$ is valid only when the DCS is integrated over E' ; otherwise the complete expression (2.40) must be used. Let $F_{\max}$ denote the maximum value of $F(p_z)$, which occurs at $p_z = 0.2\mathbf{m}_e c$ or $-0.2\mathbf{m}_e c$; a random number ξ is generated and the value p_z is accepted if $\xi F_{\max} < F(p_z)$, otherwise the process of selecting a shell and a p_z -value is reinitiated. The energy E' of the emerging photon is then calculated from Eq. (2.32), which gives

$$E' = E \frac{\tau}{1 - t\tau^2} \left[(1 - t\tau \cos \theta) + \text{sign}(p_z) \sqrt{(1 - t\tau \cos \theta)^2 - (1 - t\tau^2)(1 - t)} \right], \quad (2.69)$$

where

$$t \equiv (p_z/\mathbf{m}_e c)^2 \quad \text{and} \quad \text{sign}(p_z) \equiv p_z/|p_z|. \quad (2.70)$$

For photons with energy larger than 5 MeV, for which Doppler broadening is negligible, we set $E' = E_C$ (which amounts to assuming that $p_z = 0$). In this case, the active electron shell i is sampled with relative probability f_i and binding effects are accounted for by simply rejecting E' -values such that $E - E' < U_i$.

The azimuthal scattering angle ϕ of the photon is sampled uniformly in the interval $(0, 2\pi)$. We assume that the Compton electron is emitted with energy $E_e = E - E' - U_i$ in the direction of the momentum transfer vector $\mathbf{q} = \hbar\mathbf{k} - \hbar\mathbf{k}'$, with polar angle θ_e and azimuthal angle $\phi_e = \phi + \pi$, relative to the direction of the incident photon. $\cos \theta_e$ is given by

$$\cos \theta_e = \frac{E - E' \cos \theta}{\sqrt{E^2 + E'^2 - 2EE' \cos \theta}}. \quad (2.71)$$

When $E' = E_C$, this expression simplifies to

$$\cos \theta_e = \frac{E + \mathbf{m}_e c^2}{E} \left(\frac{E - E_C}{2\mathbf{m}_e c^2 + E - E_C} \right)^{1/2}, \quad (2.72)$$

which coincides with the result (A.20). Since the active electron shell is known, characteristic x rays and Auger electrons emitted in the de-excitation of the ionised atom can also be followed. This is important, for instance, to account for escape peaks in scintillation or solid state detectors

³Notice that, due to the approximation introduced in Eq. (2.32), a value $p_z < -\mathbf{m}_e c$ would yield a negative energy for the scattered photon.

Table 2.1: Average number n_r of random numbers ξ needed to simulate a single incoherent scattering event for photons with energy E in aluminium, silver and gold.

E (eV)	Al	Ag	Au
10^3	16.6	11.9	13.4
10^4	11.0	11.4	11.5
10^5	9.5	9.8	10.0
10^6	8.2	8.2	8.3
10^7	7.5	7.5	7.5

As a measure of the efficiency of the sampling algorithm, we may consider the average number n_r of random numbers ξ required to simulate an incoherent scattering event. n_r is practically independent of the atomic number and decreases with photon energy (see Table 2.1). The increase of n_r at low energies stems from the loss of efficiency of the algorithm used to sample $\cos\theta$. Although the simulation of incoherent events becomes more laborious as the photon energy decreases, this has only a small influence on the speed of practical photon transport simulations because low-energy photons interact predominantly via photoelectric absorption (see Fig. 2.10 below).

2.4 Electron-positron pair production

Electron-positron pairs can be created by absorption of a photon in the vicinity of a massive particle, a nucleus or an electron, which absorbs energy and momentum so that these two quantities are conserved. The threshold energy for pair production in the field of a nucleus (assumed of infinite mass) is $2m_e c^2$. When pair production occurs in the field of an electron, the target electron recoils after the event with appreciable kinetic energy; the process is known as “triplet production” because it causes three visible tracks when observed, *e.g.*, in a cloud chamber. If the target electron is at rest, triplet production is only possible for photons with energy larger than $4m_e c^2$.

For the simulation of pair-production events in the field of an atom of atomic number Z , we shall use the following semiempirical model (Baró *et al.*, 1994a). Our starting point is the high-energy DCS for arbitrary screening, which was derived by Bethe and Heitler from the Born approximation (Moz *et al.*, 1969; Tsai, 1974). The Bethe-Heitler DCS for a photon of energy E to create an electron-positron pair, in which the electron has a kinetic energy $E_- = \epsilon E - m_e c^2$, can be expressed as (Tsai, 1974)

$$\frac{d\sigma_{pp}^{(BH)}}{d\epsilon} = r_e^2 \alpha Z [Z + \eta] \left\{ [\epsilon^2 + (1 - \epsilon)^2] (\Phi_1 - 4f_C) + \frac{2}{3} \epsilon (1 - \epsilon) (\Phi_2 - 4f_C) \right\}. \quad (2.73)$$

Notice that the “reduced energy” $\epsilon = (E_- + m_e c^2)/E$ is the fraction of the photon en-

ergy that is taken away by the electron. The screening functions Φ_1 and Φ_2 are given by integrals that involve the atomic form factor and, therefore, must be computed numerically when a realistic form factor is adopted (*e.g.*, the analytical form factor described in Section 2.1). To obtain approximate analytical expressions for these functions, we shall assume that the Coulomb field of the nucleus is exponentially screened by the atomic electrons (Schiff, 1968; Tsai, 1974), *i.e.*, the electrostatic potential of the atom is assumed to be (Wentzel model)

$$\varphi_W(r) = \frac{Ze}{r} \exp(-r/R), \quad (2.74)$$

with the screening radius R considered as an adjustable parameter (see below). The corresponding atomic electron density is obtained from Poisson's equation,

$$\rho_W(r) = \frac{1}{4\pi e} \nabla^2 \varphi(r) = \frac{1}{4\pi e} \frac{1}{r} \frac{d^2}{dr^2} [r\varphi(r)] = \frac{Z}{4\pi R^2 r} \exp(-r/R), \quad (2.75)$$

and the atomic form factor is

$$F_W(q, Z) = 4\pi \int_0^\infty \rho_W(r) \frac{\sin(qr/\hbar)}{qr/\hbar} r^2 dr = \frac{Z}{1 + (Rq/\hbar)^2}. \quad (2.76)$$

The screening functions for this particular form factor take the following analytical expressions (Tsai, 1974)

$$\begin{aligned} \Phi_1 &= 2 - 2 \ln(1 + b^2) - 4b \arctan(b^{-1}) + 4 \ln(Rm_e c/\hbar) \\ \Phi_2 &= \frac{4}{3} - 2 \ln(1 + b^2) + 2b^2 [4 - 4b \arctan(b^{-1}) - 3 \ln(1 + b^{-2})] \\ &\quad + 4 \ln(Rm_e c/\hbar), \end{aligned} \quad (2.77)$$

where

$$b = \frac{Rm_e c}{\hbar} \frac{1}{2\kappa} \frac{1}{\epsilon(1 - \epsilon)}. \quad (2.78)$$

The quantity η in Eq. (2.73) accounts for pair production in the field of the atomic electrons (triplet production), which is considered in detail by Hubbell *et al.* (1980) and Tsai (1974). In order to simplify the calculations, the dependence of the triplet cross section on the electron reduced energy, ϵ , is assumed to be the same as that of the pair cross section. The function f_C in (2.73) is the high-energy Coulomb correction of Davies, Bethe and Maximon (1954) given by

$$\begin{aligned} f_C(Z) &= a^2 [(1 + a^2)^{-1} + 0.202059 - 0.03693a^2 + 0.00835a^4 \\ &\quad - 0.00201a^6 + 0.00049a^8 - 0.00012a^{10} + 0.00003a^{12}], \end{aligned} \quad (2.79)$$

with $a = \alpha Z$. The total atomic cross section for pair (and triplet) production is obtained as

$$\sigma_{pp}^{(BH)} = \int_{\epsilon_{\min}}^{\epsilon_{\max}} \frac{d\sigma_{pp}^{(BH)}}{d\epsilon} d\epsilon, \quad (2.80)$$

where

$$\epsilon_{\min} = m_e c^2 / E = \kappa^{-1} \quad \text{and} \quad \epsilon_{\max} = 1 - m_e c^2 / E = 1 - \kappa^{-1}. \quad (2.81)$$

Extensive tables of pair-production total cross sections, evaluated by combining different theoretical approximations, have been published by Hubbell *et al.* (1980). These tables give the separate contributions of pair production in the field of the nucleus and in that of the atomic electrons for $Z = 1$ to 100 and for photon energies from threshold up to 10^5 MeV. Following Salvat and Fernández-Varea (1992), the screening radius R has been determined by requiring that Eq. (2.73) with $\eta = 0$ exactly reproduces the total cross sections given by Hubbell *et al.* (1980) for pair production in the nuclear field by 10^5 MeV photons (after exclusion of radiative corrections, which only amount to $\sim 1\%$ of the total cross section). The screening radii for $Z = 1$ –99 obtained in this way are given in Table 2.2.

Actually, the triplet contribution, η , varies with the photon energy. It increases monotonically from zero at $E \simeq 4m_e c^2$ and reaches a saturation value, η_∞ , at high energies. It can be obtained, for all elements and energies up to 10^5 MeV, as

$$\eta(E) = Z \sigma_{\text{triplet}}^{\text{HGO}}(E) / \sigma_{\text{pair}}^{\text{HGO}}(E), \quad (2.82)$$

where $\sigma_{\text{pair}}^{\text{HGO}}$ and $\sigma_{\text{triplet}}^{\text{HGO}}$ are the total pair and triplet production cross sections given by Hubbell *et al.* (1980). At 10^5 MeV, the high-energy limit is reached, *i.e.*,

$$\eta_\infty \simeq Z \sigma_{\text{triplet}}^{\text{HGO}}(10^5 \text{ MeV}) / \sigma_{\text{pair}}^{\text{HGO}}(10^5 \text{ MeV}). \quad (2.83)$$

The values of η_∞ for the elements $Z = 1$ –99 are given in Table 2.2. The average dependence of η on the photon energy is approximated by the following empirical expression

$$\eta = [1 - \exp(-v)] \eta_\infty, \quad (2.84)$$

where

$$\begin{aligned} v = & (0.2840 - 0.1909a) \ln(4/\kappa) + (0.1095 + 0.2206a) \ln^2(4/\kappa) \\ & + (0.02888 - 0.04269a) \ln^3(4/\kappa) + (0.002527 + 0.002623a) \ln^4(4/\kappa). \end{aligned} \quad (2.85)$$

Thus, the single quantity η_∞ now characterises triplet production for each element.

The approximation given by Eq. (2.73) with the fitted value of the screening radius, fails at low energies where it systematically underestimates the total cross section (it can even become negative). To compensate for this fact we introduce an empirical correcting term $F_0(\kappa, Z)$, which acts in a way similar to the Coulomb correction. To facilitate the random sampling, the Bethe-Heitler DCS, Eq. (2.73), including this low-energy correction and a high-energy radiative correction, is written in the form

$$\frac{d\sigma_{\text{pp}}}{d\epsilon} = r_e^2 \alpha Z [Z + \eta] C_r \frac{2}{3} \left[2 \left(\frac{1}{2} - \epsilon \right)^2 \phi_1(\epsilon) + \phi_2(\epsilon) \right], \quad (2.86)$$

Table 2.2: Reduced screening radius, $Rm_e c/\hbar$, and high-energy triplet contribution, η_∞ , for electron-positron pair production obtained from the tables of Hubbell *et al.* (1980) as described in the text. Notice that $\hbar/m_e c = 3.8616 \times 10^{-13}$ m is the Compton wavelength of the electron.

Z	$Rm_e c/\hbar$	η_∞	Z	$Rm_e c/\hbar$	η_∞	Z	$Rm_e c/\hbar$	η_∞
1	122.81	1.157	34	32.740	1.158	67	26.108	1.197
2	73.167	1.169	35	32.438	1.158	68	25.929	1.197
3	69.228	1.219	36	32.143	1.158	69	25.730	1.198
4	67.301	1.201	37	31.884	1.166	70	25.577	1.198
5	64.696	1.189	38	31.622	1.173	71	25.403	1.200
6	61.228	1.174	39	31.438	1.174	72	25.245	1.201
7	57.524	1.176	40	31.142	1.175	73	25.100	1.202
8	54.033	1.169	41	30.950	1.170	74	24.941	1.204
9	50.787	1.163	42	30.758	1.169	75	24.790	1.205
10	47.851	1.157	43	30.561	1.172	76	24.655	1.206
11	46.373	1.174	44	30.285	1.169	77	24.506	1.208
12	45.401	1.183	45	30.097	1.168	78	24.391	1.207
13	44.503	1.186	46	29.832	1.164	79	24.262	1.208
14	43.815	1.184	47	29.581	1.167	80	24.145	1.212
15	43.074	1.180	48	29.411	1.170	81	24.039	1.215
16	42.321	1.178	49	29.247	1.172	82	23.922	1.218
17	41.586	1.175	50	29.085	1.174	83	23.813	1.221
18	40.953	1.170	51	28.930	1.175	84	23.712	1.224
19	40.524	1.180	52	28.721	1.178	85	23.621	1.227
20	40.256	1.187	53	28.580	1.179	86	23.523	1.230
21	39.756	1.184	54	28.442	1.180	87	23.430	1.237
22	39.144	1.180	55	28.312	1.187	88	23.331	1.243
23	38.462	1.177	56	28.139	1.194	89	23.238	1.247
24	37.778	1.166	57	27.973	1.197	90	23.139	1.250
25	37.174	1.169	58	27.819	1.196	91	23.048	1.251
26	36.663	1.166	59	27.675	1.194	92	22.967	1.252
27	35.986	1.164	60	27.496	1.194	93	22.833	1.255
28	35.317	1.162	61	27.285	1.194	94	22.694	1.256
29	34.688	1.154	62	27.093	1.194	95	22.624	1.257
30	34.197	1.156	63	26.911	1.194	96	22.545	1.259
31	33.786	1.157	64	26.705	1.196	97	22.446	1.262
32	33.422	1.158	65	26.516	1.197	98	22.358	1.262
33	33.068	1.157	66	26.304	1.196	99	22.264	1.265

where

$$\begin{aligned}\phi_1(\epsilon) &= g_1(b) + g_0(\kappa) \\ \phi_2(\epsilon) &= g_2(b) + g_0(\kappa)\end{aligned}\tag{2.87}$$

with

$$\begin{aligned}g_1(b) &= \frac{1}{2}(3\Phi_1 - \Phi_2) - 4\ln(Rm_e c/\hbar) = \frac{7}{3} - 2\ln(1 + b^2) - 6b \arctan(b^{-1}) \\ &\quad - b^2 [4 - 4b \arctan(b^{-1}) - 3\ln(1 + b^{-2})], \\ g_2(b) &= \frac{1}{4}(3\Phi_1 + \Phi_2) - 4\ln(Rm_e c/\hbar) = \frac{11}{6} - 2\ln(1 + b^2) - 3b \arctan(b^{-1}) \\ &\quad + \frac{1}{2}b^2 [4 - 4b \arctan(b^{-1}) - 3\ln(1 + b^{-2})], \\ g_0(\kappa) &= 4\ln(Rm_e c/\hbar) - 4f_C(Z) + F_0(\kappa, Z).\end{aligned}\tag{2.88}$$

$C_r = 1.0093$ is the high-energy limit of Mork and Olsen's radiative correction (Hubbell *et al.*, 1980).

The correcting factor $F_0(\kappa, Z)$ has been determined by requiring that the total cross section for pair production obtained from the expression given in Eq. (2.86) (with $\eta = 0$) coincides with the total cross sections for pair production in the field of the nucleus tabulated by Hubbell *et al.* (1980). By inspection and numerical fitting, we have obtained the following analytical approximation

$$\begin{aligned}F_0(\kappa, Z) &= (-0.1774 - 12.10a + 11.18a^2)(2/\kappa)^{1/2} \\ &\quad + (8.523 + 73.26a - 44.41a^2)(2/\kappa) \\ &\quad - (13.52 + 121.1a - 96.41a^2)(2/\kappa)^{3/2} \\ &\quad + (8.946 + 62.05a - 63.41a^2)(2/\kappa)^2.\end{aligned}\tag{2.89}$$

The functions ϕ_1 and ϕ_2 are now positive except for ϵ -values very near the endpoints of the allowed interval, given by Eq. (2.81), for high-atomic-number elements. To avoid inconsistencies, these functions are set equal to zero when they take negative values.

The relative differences between the total atomic cross sections obtained from the DCS given by Eq. (2.86) and the total cross sections tabulated by Hubbell *et al.* (1980) are appreciable near the threshold [actually, (2.86) shifts the threshold for pair production to values slightly larger than $2m_e c^2$], but decrease rapidly with increasing photon energy. At $E = 3$ MeV, the differences reduce to 4% and do not exceed 2% for energies larger than 6 MeV, for almost all the elements. Although these differences are not important, they may be larger than the uncertainties in the cross sections given by Hubbell *et al.* (1980). To avoid systematic errors, the mean free paths for pair production used in PENELOPE are obtained by interpolation in a table generated with the XCOM program (Berger and Hubbell, 1987). The Bethe-Heitler DCS is only used to sample the kinetic energies of the produced pair.

It is also worth noting that the Bethe-Heitler theory predicts that the pair-production DCS, considered as a function of the electron reduced energy ϵ , is symmetrical about $\epsilon = 1/2$ (see Fig. 2.9). This dependence on ϵ is reasonably accurate only for photon energies larger than ~ 5 MeV. For lower photon energies, the effect of the electrostatic field of the atom (which slows down the electron and accelerates the positron) becomes increasingly important, with the result that the actual DCS becomes asymmetrical and the mean value of ϵ becomes less than $1/2$ (see, *e.g.*, Motz *et al.*, 1969). At these relatively low energies, however, pair production is not dominant and, moreover, the produced particles have ranges that are much less than the mean free path of the absorbed photon. Therefore, no appreciable simulation errors are incurred by using the Bethe-Heitler DCS, Eq. (2.86), for energies down to the threshold.

2.4.1 Simulation of pair-production events

The Bethe-Heitler DCS, Eq. (2.86), only depends on the kinetic energy $E_- = \epsilon E - m_e c^2$ of the produced electron, so that E_- can be directly sampled from Eq. (2.86); the kinetic energy of the positron is obtained as $E_+ = E - E_- - 2m_e c^2$. Notice that, although the Bethe-Heitler total atomic cross section accounts for pair and triplet production, all the events are simulated as if they were pairs. This approximation is justified by the fact that, in triplet production, the recoiling electron has a range that is much smaller than the mean free path of the incident photon.

The electron reduced energy ϵ is distributed in the interval $(\kappa^{-1}, 1 - \kappa^{-1})$, see Eq. (2.81), according to the PDF given by Eq. (2.86) (normalisation is again irrelevant)

$$p_{pp}(\epsilon) = 2 \left(\frac{1}{2} - \epsilon \right)^2 \phi_1(\epsilon) + \phi_2(\epsilon), \quad (2.90)$$

which is symmetrical about the point $\epsilon = 1/2$. Figure 2.9 shows this PDF for lead and various photon energies. The following algorithm for sampling ϵ is based on the fact that the functions $\phi_1(\epsilon)$ and $\phi_2(\epsilon)$ are non-negative and attain their maximum values at $\epsilon = 1/2$.

Except for a normalisation constant, the PDF (2.90) can be written in the form

$$p_{pp}(\epsilon) = u_1 U_1(\epsilon) \pi_1(\epsilon) + u_2 U_2(\epsilon) \pi_2(\epsilon) \quad (2.91)$$

with

$$u_1 = \frac{2}{3} \left(\frac{1}{2} - \frac{1}{\kappa} \right)^2 \phi_1(1/2), \quad u_2 = \phi_2(1/2), \quad (2.92)$$

$$\pi_1(\epsilon) = \frac{3}{2} \left(\frac{1}{2} - \frac{1}{\kappa} \right)^{-3} \left(\frac{1}{2} - \epsilon \right)^2, \quad \pi_2(\epsilon) = \frac{1}{2} \left(\frac{1}{2} - \frac{1}{\kappa} \right)^{-1} \quad (2.93)$$

and

$$U_1(\epsilon) = \phi_1(\epsilon)/\phi_1(1/2), \quad U_2(\epsilon) = \phi_2(\epsilon)/\phi_2(1/2). \quad (2.94)$$

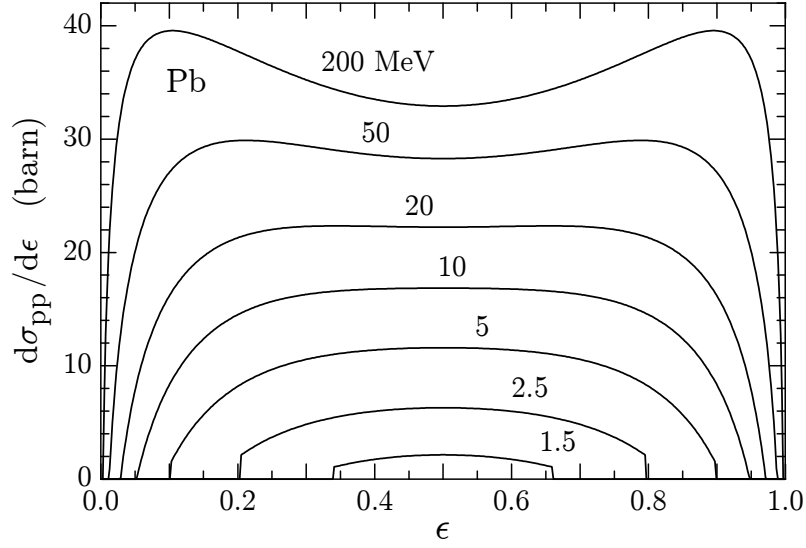


Figure 2.9: DCS for pair-production in lead as a function of the electron reduced energy, $\epsilon = (E_- + m_e c^2)/E$. (Adapted from Baró *et al.*, 1994a.)

The functions $\pi_i(\epsilon)$ are normalised PDFs in the interval $(\kappa^{-1}, 1 - \kappa^{-1})$, from which random values of ϵ can be easily sampled by using the inverse-transform method. In this interval, the functions $U_i(\epsilon)$ are positive and less than unity, *i.e.*, they are valid rejection functions. The generation of random values of ϵ from the distribution (2.91) can now be performed by combining the composition and rejection methods (see Section 1.2) according to the following algorithm:

- (i) Sample a value of the integer i ($=1, 2$) according to the point probabilities

$$p(1) = \frac{u_1}{u_1 + u_2} \quad \text{and} \quad p(2) = \frac{u_2}{u_1 + u_2}. \quad (2.95)$$

- (ii) Sample ϵ from $\pi_i(\epsilon)$ using the sampling formulas (inverse-transform method, see Section 1.2.2)

$$\epsilon = \begin{cases} \frac{1}{2} + \left(\frac{1}{2} - \frac{1}{\kappa} \right) (2\xi - 1)^{1/3} & \text{if } i = 1, \\ \frac{1}{\kappa} + \left(\frac{1}{2} - \frac{1}{\kappa} \right) 2\xi & \text{if } i = 2. \end{cases} \quad (2.96)$$

- (iii) Generate a new random number ξ .
- (iv) If $\xi > U_i(\epsilon)$, go to step (i).
- (v) Deliver ϵ .

Notice that the quantity $2\xi - 1$ may be negative and, therefore, taking its cube root will lead to a computer error; provision of this fact must be made when programming the algorithm. The efficiency of the algorithm is greater than 70% for energies near the threshold, and increases with increasing photon energies. For $E = 1$ GeV it is of the order of 95% for all the elements in the periodic table.

2.4.1.1 Angular distribution of the produced particles

Actually, the complete DCS for pair production is a function of the directions of the pair of particles. As the final state involves three bodies (the nucleus and the produced pair), the directions of the produced particles cannot be obtained from only their kinetic energies. The polar angles of the directions of movement of the electron and positron (θ_- and θ_+ , Fig. 2.1) relative to the direction of the incident photon are sampled from the leading term of the expression obtained from high-energy theory (Heitler, 1954; Motz et al., 1969)

$$p(\cos \theta_{\pm}) = a (1 - \beta_{\pm} \cos \theta_{\pm})^{-2}, \quad (2.97)$$

where a is a normalisation constant and

$$\beta_{\pm} = \frac{\sqrt{E_{\pm}(E_{\pm} + 2m_e c^2)}}{E_{\pm} + m_e c^2} \quad (2.98)$$

is the particle velocity in units of the speed of light. Random values of $\cos \theta_{\pm}$ are obtained by using the inverse-transform method (see Section 1.2.2), which leads to the sampling formula

$$\cos \theta_{\pm} = \frac{2\xi - 1 + \beta_{\pm}}{(2\xi - 1)\beta_{\pm} + 1}. \quad (2.99)$$

As the directions of the produced particles and the incident photon are not necessarily coplanar, the azimuthal angles ϕ_- and ϕ_+ of the electron and the positron are sampled independently and uniformly in the interval $(0, 2\pi)$.

It is worth stressing the fact that the produced charged particles have ranges that are much smaller than the mean free path of the photons. Moreover, the charged particles immediately enter a multiple elastic scattering process which randomises their directions of movement. As a consequence, there should be little difference between simulation results obtained with the present method and with exact random sampling from a more accurate DCS, differential in the energies and directions of the generated particles.

2.4.1.2 Compound materials

Let us consider a compound $X_x Y_y$, in which the molecules consist of x atoms of the element X and y atoms of the element Y . The number of electrons per molecule is $Z_M = xZ(X) + yZ(Y)$ and the molar mass is $A_M = xA_m(X) + yA_m(Y)$, where $Z(X)$ and $A_m(X)$ stand for the atomic number and molar mass of element X .

In the simulation of pair-production events, we could use the molecular DCSs obtained from the additivity rule. The simulation of each event would then consist of 1) sampling the atom which participates in the interaction and 2) generating a random value of the electron reduced energy ϵ from the corresponding atomic DCS. To save computer time, PENELOPE generates ϵ by considering an “equivalent” single element material of the same mass density ρ as the actual medium, atomic number Z_{eq} and molar mass A_{eq} given by

$$Z_{\text{eq}}A_{\text{M}} = Z_{\text{M}}A_{\text{eq}} = xZ(\text{X})A_{\text{m}}(\text{X}) + yZ(\text{Y})A_{\text{m}}(\text{Y}), \quad (2.100)$$

i.e., its atomic number (molar mass) is the mass-average (Z -average) of the atomic numbers (molar masses) of the constituent atoms. The reduced energy is sampled from the DCS of the element with the atomic number closest to Z_{eq} . Usually, this approximation does not alter the simulation results appreciably and permits a considerable simplification of the program and a reduction of the simulation time.

2.5 Attenuation coefficients

The photon inverse mean free path for a given mechanism is known as the partial attenuation coefficient of that mechanism. Thus, the partial attenuation coefficient for photoelectric absorption is

$$\mu_{\text{ph}} = \mathcal{N}\sigma_{\text{ph}}, \quad (2.101)$$

where $\mathcal{N} = N_{\text{A}}\rho/A_{\text{M}}$ is the number of atoms or molecules per unit volume and σ_{ph} is the atomic or molecular photoelectric cross section. The photoelectric mass attenuation coefficient is defined as μ_{ph}/ρ and, therefore, is independent of the density of the material. Analogous definitions apply for the other interaction processes. The total mass attenuation coefficient is obtained as

$$\frac{\mu}{\rho} = \frac{N_{\text{A}}}{A_{\text{M}}} (\sigma_{\text{Ra}} + \sigma_{\text{Co}} + \sigma_{\text{ph}} + \sigma_{\text{pp}}). \quad (2.102)$$

As mentioned above, PENELOPE uses tables of total cross sections for photoelectric absorption and pair production obtained from the database EPDL (Cullen *et al.*, 1997) and the program XCOM (Berger and Hubbell, 1987), respectively. Photoelectric cross sections for energies different from those in the tables are calculated by linear log-log interpolation. Total cross sections for pair production are evaluated by cubic spline log-log interpolation of the function $(1 - 2m_{\text{e}}c^2/E)^{-3}\sigma_{\text{pp}}$, which varies slowly with the photon energy.

Mean free paths for coherent and incoherent scattering are computed from the DCSs described in Sections 2.1 and 2.3. The resulting values are virtually identical to those given by the XCOM program for E greater than ~ 50 keV. At lower energies, our mean free paths for Compton scattering deviate from those given by XCOM; these were calculated from a different theoretical model (Hubbell *et al.*, 1975), which neglects Doppler

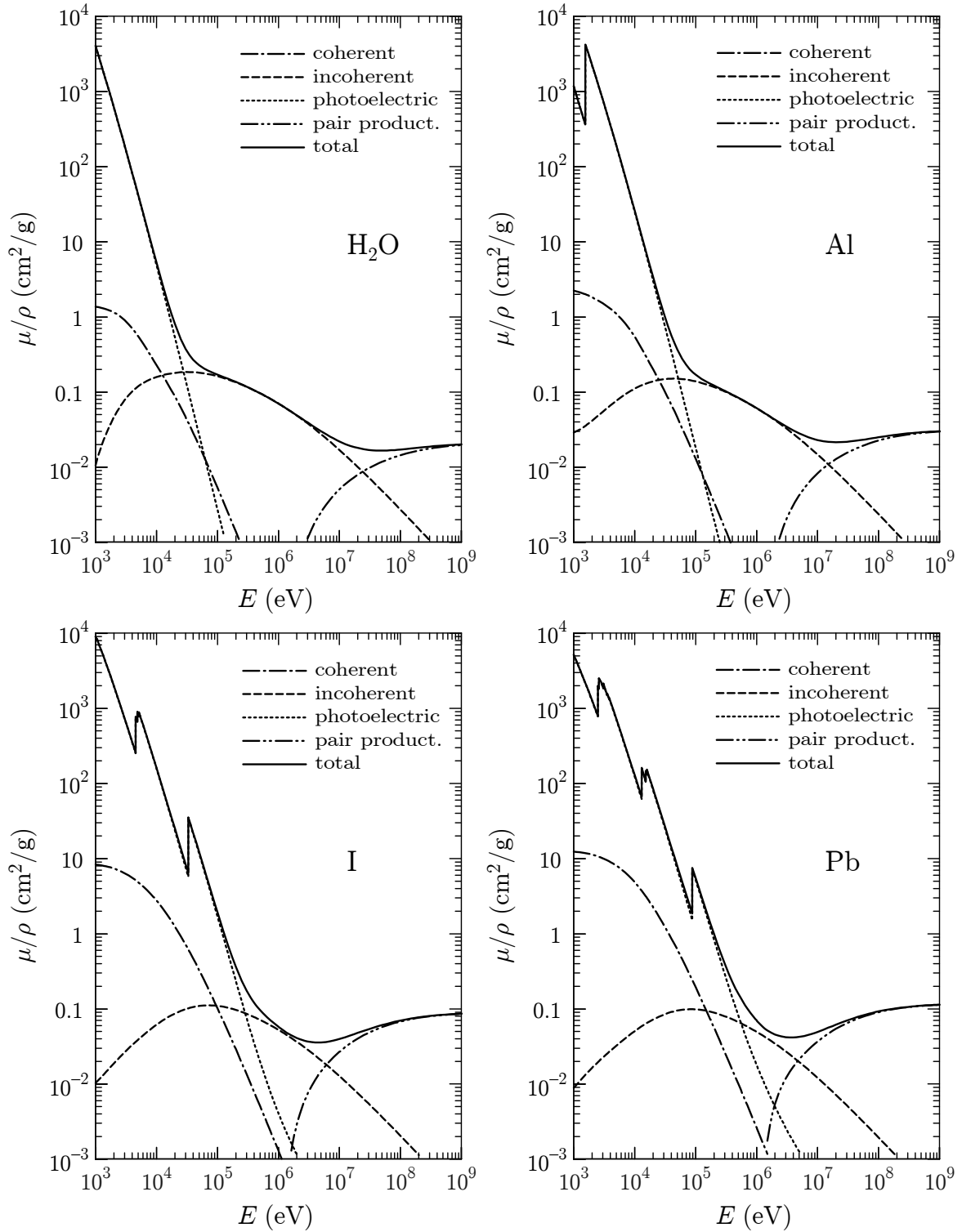


Figure 2.10: Partial and total mass attenuation coefficients of water, aluminium, iodine and lead as functions of the photon energy.

broadening (see, *e.g.*, Brusa *et al.*, 1996). The evaluation of the total atomic cross section for these processes [see Eqs. (2.10) and (2.50)] involves a numerical quadrature, which is performed by using the function **SUMGA** (Appendix B). Notice that for high-energy photons, the integrand in the coherent scattering cross section, Eq. (2.10), is sharply peaked at $\theta = 0$. In such a case, the numerical integration method is not effective. For energies larger than $\sim Z/2$ MeV, we take advantage of the asymptotic behaviour shown by Eq. (2.12) to avoid time-consuming integration. Partial and total mass attenuation coefficients for water and lead, as representatives of low- and high- Z materials, are displayed in Fig. 2.10.

As already mentioned, PENELOPE simulates photon transport by using the conventional detailed (interaction by interaction) method. To sample the path length to the next interaction site and the type of event that occurs there, the values of total and partial attenuation coefficients at the current energy E are required. During simulation, these attenuation coefficients are obtained by interpolation from pre-calculated tables stored in memory. In PENELOPE, energy-dependent quantities are usually tabulated for a logarithmic grid of 200 energies, E_i ($i = 1, \dots, 200$), which spans the complete energy range considered in the simulation and allows accurate linear log-log interpolation. This scheme works well for pair production and for Rayleigh and Compton scattering, whose partial attenuation coefficients vary smoothly with energy. However, it is not applicable to photoelectric absorption, because its partial attenuation coefficient $\mu_{\text{ph}}(E)$ is not continuous (see Fig. 2.10). To ensure accuracy, the grid of energies where $\mu_{\text{ph}}(E)$ is tabulated must include absorption edges and, therefore, cannot be the same for all materials. Consequently, interpolation of the attenuation coefficient for photoelectric absorption is lengthier than for other interactions.

Many interpolations of $\mu_{\text{ph}}(E)$ can be avoided by introducing delta interactions (see Section 1.4.5) as follows. Within the grid interval (E_i, E_{i+1}) , the attenuation coefficient for delta interactions is defined by

$$\mu_{\delta}(E) = \mu_{\text{ph,max}} - \mu_{\text{ph}}(E), \quad (2.103)$$

where $\mu_{\text{ph,max}}$ is the maximum value of $\mu_{\text{ph}}(E)$ in that energy interval. It is convenient to consider photoelectric absorption and delta interactions as a combined process (“ph+ δ ”) with partial attenuation coefficient equal to $\mu_{\text{ph,max}}$, which is constant within each grid interval. To sample the length of the free flight to the next interaction, we use the total attenuation coefficient

$$\mu' = \mu_{\text{Ra}} + \mu_{\text{Co}} + \mu_{\text{pp}} + \mu_{\text{ph,max}}. \quad (2.104)$$

Note that interpolation of μ_{ph} is not required at this point. The kind of interaction that occurs at the end of the free flight is sampled from the relative probabilities of the various interaction mechanisms (“Ra”, “Co”, “pp”, and “ph+ δ ”), which are proportional to the corresponding partial attenuation coefficients,

$$p_{\text{Ra}} = \frac{\mu_{\text{Ra}}}{\mu'}, \quad p_{\text{Co}} = \frac{\mu_{\text{Co}}}{\mu'}, \quad p_{\text{pp}} = \frac{\mu_{\text{pp}}}{\mu'}, \quad p_{\text{ph}+\delta} = \frac{\mu_{\text{ph,max}}}{\mu'} = \frac{\mu_{\delta} + \mu_{\text{ph}}}{\mu'}. \quad (2.105)$$

Only when the outcome is “ph+ δ ”, we need to evaluate the photoelectric attenuation coefficient (to determine whether the interaction is photoelectric absorption or a delta

interaction). Thus, the costly interpolation of $\mu_{\text{ph}}(E)$ is performed only when the probability of photoabsorption is appreciable.

2.6 Atomic relaxation

Atoms are primarily ionised by photon interactions and by electron or positron impact. There is a fundamental difference between the ionising effects of photons and of charged particles. A photon is only able to directly ionise a few atoms. In the case of photoabsorption, when the photon energy is larger than the K-shell binding energy, about 80% of photoabsorptions occur in the K shell, *i.e.*, the resulting ion with a vacancy in the K shell is highly excited. Incoherent scattering is not as highly preferential, but still the probability that an inner shell is ionised is nearly proportional to the number of electrons in the shell. Conversely, fast electrons and positrons (and other charged particles) ionise many atoms along their paths; the ionisations occur preferentially in the less tightly bound atomic shells, or the conduction band in the case of metals (see Section 3.2), so that most of the produced ions are only weakly excited.

Excited ions with a vacancy in an inner shell relax to their ground state through a sequence of radiative and non-radiative transitions. In a radiative transition, the vacancy is filled by an electron from an outer shell and an x ray with characteristic energy is emitted. In a non-radiative transition, the vacancy is filled by an outer electron and the excess energy is released through emission of an electron from a shell that is farther out (Auger effect). Each non-radiative transition generates an additional vacancy that, in turn, migrates “outwards”. The production of vacancies in inner shells and their subsequent relaxation must be simulated in detail, since the energetic x rays and/or electrons emitted during the process may transport energy quite a distance from the excited ion.

PENELOPE simulates the emission of characteristic radiation and Auger electrons that result from vacancies produced in K, L and M shells by photoelectric absorption, Compton scattering and electron/positron impact (see Chapter 3). The relaxation of these vacancies is followed until the K, L and M shells are filled up, *i.e.*, until the vacancies have migrated to N and outer shells. Vacancies in these outer shells originate much less energetic secondary radiation, whose main effect is to spread out the excitation energy of the ion within the surrounding material. To obtain a reliable description of the dose distribution, and other macroscopic transport characteristics, we only have to follow secondary radiation that is able to propagate to distances of the order of, say, 1% of the penetration distance (or range) of the primary radiation. Radiation with lower energy does not need to be followed, since its only effect is to blur the “primary” dose distribution on a small length scale.

To simplify the description of the ionisation processes of outer shells (*i.e.*, photoelectric absorption, Compton scattering and electron/positron impact), we simply assume that, when ionisation occurs in N or outer shells, a secondary (delta) electron is emitted from the parent ion with a kinetic energy E_s equal to the energy deposited by the

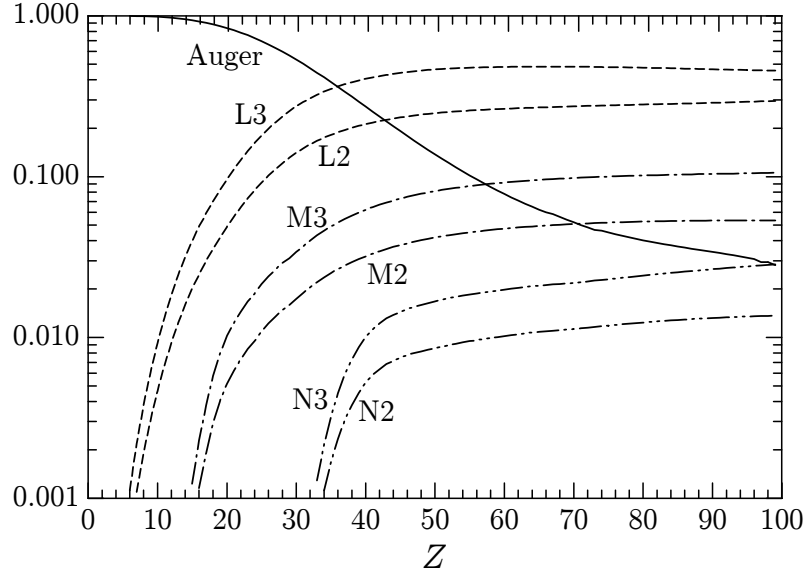


Figure 2.11: Relative probabilities for radiative and non-radiative (Auger) transitions that fill a vacancy in the K-shell of atoms.

primary particle,

$$E_{\text{dep}} = \begin{cases} E - E' & \text{in Compton scattering,} \\ E & \text{in photoelectric absorption,} \\ W & \text{in electron/positron impact (see Chapter 3).} \end{cases} \quad (2.106)$$

That is, the whole excitation energy of the ion is taken up by the ejected electron and no fluorescent radiation is simulated. In reality, the emitted electrons have energies that are less than the (2.106) values and can be followed by characteristic x rays, which have mean free paths that are usually much larger than the Bethe range of photoelectrons. By giving an artificially increased initial energy to the electron we allow it to transport energy further from the ion so as to partially compensate for the neglect of other radiation emitted during the de-excitation cascade.

In the case of ionisation of an inner shell i , *i.e.*, a K, L or M shell, we consider that the electron is ejected with kinetic energy

$$E_s = E_{\text{dep}} - U_i, \quad (2.107)$$

where U_i is the ionisation energy of the active shell, and that the target atom is left with a vacancy in shell i . As mentioned above, we consider only characteristic x rays and Auger electrons emitted in the first stages of the relaxation process. These secondary radiations are assumed to be emitted isotropically from the excited atom. We use the following notation to designate the possible transitions

- Radiative: S0-S1 (an electron from the S1 shell fills the vacancy in the S0 shell, leaving a hole in the S1 shell). The considered radiative transitions (for elements with $Z > 18$

with the M-shell filled) are shown in Fig. 2.3.

- Non-radiative: S0-S1-S2 (an electron from the S1 shell fills the vacancy in the S0 shell, and the released energy is taken away by an electron in the S2 shell; this process leaves two vacancies in the S1 and S2 shells).

Non-radiative transitions of the type $LI-LJ-Xq$, which involve an electron transition between two L-subshells and the ejection of an electron from an outer shell Xq are known as L-shell Coster-Kronig transitions.

The information furnished to PENELOPE for each element consists of a table of possible transitions, transition probabilities and energies of the emitted x rays or electrons for ionised atoms with a single vacancy in the K-shell or in an L- or M-subshell. These data are entered through the material definition file. The transition probabilities are extracted from the LLNL Evaluated Atomic Data Library (Perkins *et al.*, 1991). Figure 2.11 displays transition probabilities for the transitions that fill a vacancy in the K shell as functions of the atomic number Z ; the curve labelled “Auger” corresponds to the totality of non-radiative transitions. We see that for low- Z elements, the relaxation proceeds mostly through non-radiative transitions. It is worth noting that the ratio of probabilities of the radiative transitions K-S2 and K-S3 (where S stands for L, M or N) is approximately 1/2, as obtained from the dipole approximation (see, *e.g.*, Bransden and Joachain, 1983); radiative transitions K-S1 are strictly forbidden (to first order) within the dipole approximation.

The energies of x rays emitted in radiative transitions resulting from single vacancies in the K and L shells were taken from the recent compilation by Deslattes *et al.* (2003). X-ray energies for transitions of vacancies in M shells were taken from Bearden’s (1967) review and reevaluation of experimental x-ray wavelengths. In the case of radiative transitions S0-S1 not included in Bearden’s tabulation, the energy of the x ray is approximated as

$$E_x = U_{S0} - U_{S1}, \quad (2.108)$$

where U_{Si} is the binding energy of an electron in the shell Si of the neutral atom, which is taken from the PENELOPE database. The energy of the electron emitted in the non-radiative transition S0-S1-S2 is set equal to

$$E_e = U_{S0} - U_{S1} - U_{S2}. \quad (2.109)$$

Note that these recipes correspond to assuming that the presence of the vacancy (or vacancies) does not alter the ionisation energies of the active electron shells, which is an approximation. It should be noted that these transition energies are also used to determine the energies of the emitted radiation at any stage of the de-excitation cascade, which means that we neglect the possible relaxation of the ion (see, *e.g.*, Sevier, 1972). Therefore, our approach will not produce L_α and L_β x-ray satellite lines; these arise from the filling of a vacancy in a doubly-ionised L shell (generated, *e.g.*, by a Coster-Kronig transition), which releases energy that is slightly different from the energy liberated when the shell contains only a single vacancy. It is also worth recalling that the adopted transition probabilities are approximate. For K shells they are expected to be accurate to within about 1%, but for other shells they are subject to much larger uncertainties.

Even the L-shell fluorescence yield (the sum of radiative transition probabilities for an L-shell vacancy) is uncertain by about 20% (see, *e.g.*, Hubbell, 1989; Perkins *et al.*, 1991).

The simulation of the relaxation cascade is performed by subroutine **RELAX**. The transition that fills the initial vacancy is randomly selected according to the adopted transition probabilities, by using Walker's aliasing method (Section 1.2.3.1). This transition leaves the ion with one or two vacancies. If the energy of the emitted characteristic x ray or Auger electron is larger than the corresponding absorption energy, the state variables of the particle are stored in the secondary stack (which contains the initial states of all particles produced during the current shower that have not yet been simulated). The generation of the cascade continues by repeating the process for each remaining vacancy. It ends either when the K, L and M subshells have been filled up or when there is not enough energy to produce "active" radiation (with energy larger than the absorption energy). The excitation energy of the residual ion is assumed to be deposited locally.

It is important to bear in mind that we are disregarding the emission and transport of soft x rays and slow electrons. This sets a lower limit to the photon energies for which PENELOPE is applicable. In principle, simulation results are expected to be reliable only for photons with energies larger than the ionisation energy of the N1 subshell of the heaviest element present (95 eV for silver, 759 eV for gold and 1.442 keV for uranium).

Chapter 3

Electron and positron interactions

In this Chapter we consider the interactions of fast electrons and positrons of kinetic energy E with matter. For the sake of simplicity, we start by assuming that the particles move in a single-element medium of atomic number Z and density ρ , with $\mathcal{N}$ atoms per unit volume. The extension to compounds, and mixtures, is normally done on the basis of the additivity approximation, *i.e.*, the molecular DCS is approximated as the *incoherent* sum of the atomic DCSs of all the atoms in a molecule.

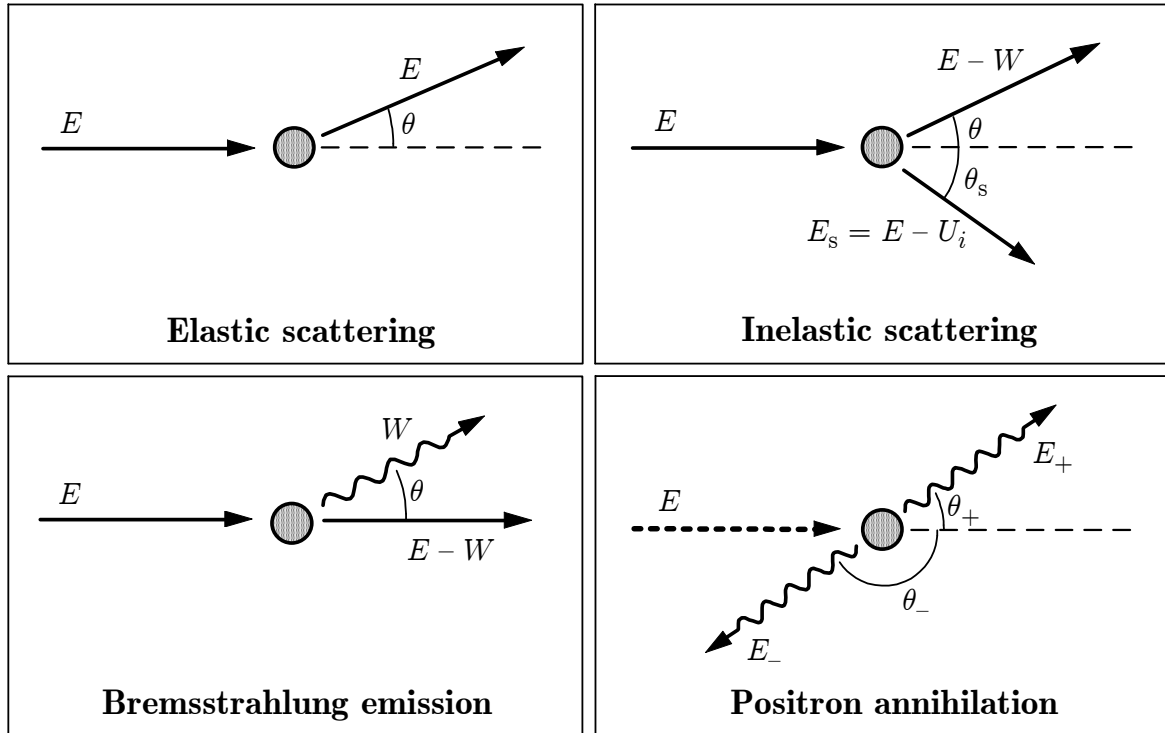


Figure 3.1: Basic interactions of electrons and positrons with matter.

The possible interactions of electrons and positrons with the medium are elastic scattering, inelastic collisions and bremsstrahlung emission; positrons can also undergo annihilation, either in flight or at rest (see Fig. 3.1). The atomic DCSs adopted in PENELOPE are defined either as analytical functions or by means of numerical tables, or as a combination of both. These DCSs, which are sufficiently accurate for most practical simulation purposes, allow fast and accurate random sampling of the individual interactions. It is worth pointing out that multiple-scattering distributions are quite insensitive to the fine details of the single-scattering DCSs. If the adopted DCSs have a physically reasonable shape, only the values of a few integrals of the DCS have a direct influence on the simulation results (Liljequist, 1987; Fernández-Varea *et al.*, 1993b). As a consequence, a general-purpose simulation procedure can be made fairly simple by using approximate DCSs with the proviso that they exactly reproduce the correct values of the relevant integrals. The DCSs described below represent a compromise between reliability and simplicity; they are simple enough to allow the use of fast sampling methods and, at the same time, they are flexible enough to account for the relevant features of the interactions.

Owing to the large number of interactions suffered by a fast electron or positron before coming to rest, detailed simulation is unfeasible at high energies. In PENELOPE we overcome this practical difficulty by using a mixed simulation procedure (see Chapter 4) instead of the habitual condensed simulation schemes adopted in other high-energy simulation codes —*e.g.*, ETRAN (Berger and Seltzer, 1988), ITS3 (Hableib *et al.*, 1992), EGS4 (Nelson *et al.*, 1985), GEANT3 (Brun *et al.*, 1986), EGSnrc (Kawrakow and Rogers, 2001), MCNP (X-5 Monte Carlo Team, 2003), GEANT4 (Agostinelli *et al.*, 2003; Allison *et al.*, 2006), FLUKA (Ferrari *et al.*, 2005), EGS5 (Hirayama *et al.*, 2005),. The formulation of mixed simulation is complicated by the fact that the sampling of hard interactions is done from restricted DCSs, with cutoffs that vary with the particle energy during the evolution of a track. This limits the complexity of the DCSs that can be efficiently used in a simulation code.

3.1 Elastic collisions

In this Section we consider the theoretical description of elastic collisions of electrons and positrons with isolated neutral atoms of atomic number Z at rest. By definition, elastic interactions are those in which the initial and final quantum states of the target atom are the same, normally the ground state. The angular deflections of electron trajectories in matter are mainly (but not completely) due to elastic scattering. Notice that there is a certain energy transfer from the projectile to the target, which causes the recoil of the latter (see Section A.1.1). Because of the large mass of the target ($\sim 3600Zm_e$), the average energy lost by the projectile is a very small fraction of its initial energy (a few meV for scattering of 30 keV electron by aluminium atoms) and is usually neglected, which is equivalent to assuming that the target has an infinite mass and does not recoil.

Elastic collisions of electrons and positrons with kinetic energies larger than a few

hundred eV can be described as scattering of the projectile by the electrostatic field of the target (Mott and Massey, 1965). The charge distribution of the target atom consists of the nucleus and the electron cloud. The density of atomic electrons $\rho_e(\mathbf{r})$ can be calculated by using available Hartree-Fock codes (*e.g.*, the one of Desclaux, 1975). For atoms with closed-shell configurations, the electron distribution is spherically symmetrical; for atoms with open shells, we assume that an average over directions is performed to give a spherical density $\rho_e(r)$. To account for the effect of the finite size of the nucleus on the elastic DCS (which is appreciable only for projectiles with energy E larger than a few MeV), the density of protons within the nucleus may be approximated by the Fermi distribution (Hahn *et al.*, 1956)

$$\rho_n(r) = \frac{\rho_0}{\exp[(r - R_n)(4 \ln 3/t)] + 1}, \quad (3.1)$$

where R_n is the mean radius (or half-density radius) and t is the surface (or “skin”) thickness (defined as the distance over which ρ_n drops from ~ 0.9 to ~ 0.1 of its central value). These two parameters are given by

$$R_n = 1.07 \times 10^{-15} A_m^{1/3} \text{ m}, \quad t = 2.4 \times 10^{-15} \text{ m}, \quad (3.2)$$

where A_m is the molar mass (in g/mol) of the element. The constant ρ_0 , which equals twice the proton density at $r = R_n$, is determined by normalisation,

$$Z = 4\pi \int_0^\infty \rho_n(r) r^2 dr. \quad (3.3)$$

The electrostatic potential of the target atom is

$$\begin{aligned} \varphi(r) = & e 4\pi \left[\frac{1}{r} \int_0^r \rho_n(r') r'^2 dr' + \int_r^\infty \rho_n(r') r' dr' \right] \\ & - e 4\pi \left[\frac{1}{r} \int_0^r \rho_e(r') r'^2 dr' + \int_r^\infty \rho_e(r') r' dr' \right]. \end{aligned} \quad (3.4)$$

Within the static-field approximation (Mott and Massey, 1965; Walker, 1971; Salvat *et al.*, 2005), the DCS for elastic scattering of electrons or positrons is obtained by solving the partial-wave expanded Dirac equation for the motion of the projectile in the field of the target atom. The interaction energy is given by

$$V(r) = z_0 e \varphi(r) + V_{\text{ex}}(r), \quad (3.5)$$

where z_0 is the charge of the projectile in units of e (-1 for electrons, $+1$ for positrons). The term $V_{\text{ex}}(r)$, which applies only for electrons, represents a local approximation to the exchange interaction between the projectile and the atomic electrons (see, *e.g.*, Salvat, 1998; Salvat, 2003). We shall limit our considerations to the case of spin-unpolarised projectiles, *i.e.*, their spin is randomly oriented. Then, the effect of elastic interactions can be described as a deflection of the projectile trajectory, characterised by the polar

and azimuthal scattering angles θ and ϕ . For a central field, the angular distribution of singly-scattered electrons is axially symmetric about the direction of incidence, *i.e.*, independent of ϕ . The DCS (per unit solid angle) for elastic scattering of a projectile with kinetic energy E into the solid angle element $d\Omega$ about the direction (θ, ϕ) is given by (Walker, 1971; Salvat *et al.*, 2005)

$$\frac{d\sigma_{\text{el}}}{d\Omega} = |f(\theta)|^2 + |g(\theta)|^2, \quad (3.6)$$

where

$$\begin{aligned} f(\theta) &= \frac{1}{2ik} \sum_{\ell=0}^{\infty} \{(\ell+1) [\exp(2i\delta_{\ell+}) - 1] + \ell [\exp(2i\delta_{\ell-}) - 1]\} P_{\ell}(\cos \theta), \\ g(\theta) &= \frac{1}{2ik} \sum_{\ell=0}^{\infty} \{\exp(2i\delta_{\ell-}) - \exp(2i\delta_{\ell+})\} P_{\ell}^1(\cos \theta) \end{aligned} \quad (3.7)$$

are the direct and spin-flip scattering amplitudes, respectively. The quantity

$$k \equiv \frac{p}{\hbar} = \frac{1}{\hbar c} [E(E + 2m_e c^2)]^{1/2} \quad (3.8)$$

is the wave number of the projectile, $P_{\ell}(\cos \theta)$ are Legendre polynomials, $P_{\ell}^1(\cos \theta)$ are associated Legendre functions and $\delta_{\ell\pm}$ are the phase shifts. These are determined from the asymptotic behaviour of the Dirac radial functions for large r (Walker, 1971; Salvat *et al.*, 2005). Thus, to determine each phase shift we must solve the radial Dirac equations for the potential $V(r)$. The convergence of the partial-wave series (3.7) slows down when the energy of the projectile increases. This makes the calculation difficult for energies larger than a few MeV (in the case of scattering by gold atoms, about 10,000 phase shifts are required at $E = 10$ MeV). The partial-wave DCS, Eq. (3.6), rigorously accounts for spin and other relativistic effects, as well as for finite nuclear size effects.

Single elastic collisions are determined by the values of the polar and azimuthal scattering angles, θ and ϕ , respectively. Owing to the assumed spherical symmetry of the scattering centres, single- and multiple-scattering angular distributions are axially symmetrical about the direction of incidence, *i.e.*, they are independent of the azimuthal scattering angle ϕ . For simulation purposes, it is convenient to measure polar angular deflections produced by single-scattering events in terms of the variable [see Eq. (1.76)]

$$\mu = (1 - \cos \theta)/2 \quad (3.9)$$

instead of the scattering angle θ . Note that μ varies from 0 (forward scattering) to 1 (backward scattering). The DCS per unit angular deflection is

$$\frac{d\sigma_{\text{el}}}{d\mu} = 4\pi \frac{d\sigma_{\text{el}}}{d\Omega}. \quad (3.10)$$

The total elastic cross section is given by

$$\sigma_{\text{el}} = \int \frac{d\sigma_{\text{el}}}{d\Omega} d\Omega = \int \frac{d\sigma_{\text{el}}}{d\mu} d\mu. \quad (3.11)$$

We can also write

$$\frac{d\sigma_{\text{el}}}{d\mu} = \sigma_{\text{el}} p_{\text{el}}(\mu), \quad (3.12)$$

where $p_{\text{el}}(\mu)$ is the normalised PDF of μ in a single collision. The mean free path between consecutive elastic events in a homogeneous single-element medium is

$$\lambda_{\text{el}} = 1/(\mathcal{N}\sigma_{\text{el}}), \quad (3.13)$$

where $\mathcal{N}$ is the number of atoms per unit volume.

Other important quantities (see Section 4.1) are the transport cross sections

$$\sigma_{\text{el},\ell} \equiv \int [1 - P_{\ell}(\cos \theta)] \frac{d\sigma_{\text{el}}}{d\Omega} d\Omega, \quad (3.14)$$

where $P_{\ell}(x)$ are Legendre polynomials. Those of lowest degrees are

$$P_0(x) = 1, \quad P_1(x) = x, \quad P_2(x) = \frac{1}{2}(3x^2 - 1). \quad (3.15)$$

The ℓ -th transport mean free path is defined by

$$\lambda_{\text{el},\ell} \equiv 1/(\mathcal{N}\sigma_{\text{el},\ell}). \quad (3.16)$$

The first and second transport cross sections, $\sigma_{\text{el},1}$ and $\sigma_{\text{el},2}$, are given by

$$\sigma_{\text{el},1} = \int (1 - \cos \theta) \frac{d\sigma_{\text{el}}}{d\Omega} d\Omega = 2\sigma_{\text{el}} \int_0^1 \mu p_{\text{el}}(\mu) d\mu = 2\sigma_{\text{el}} \langle \mu \rangle \quad (3.17)$$

and

$$\begin{aligned} \sigma_{\text{el},2} &= \int \frac{3}{2}(1 - \cos^2 \theta) \frac{d\sigma_{\text{el}}}{d\Omega} d\Omega \\ &= 6\sigma_{\text{el}} \int_0^1 (\mu - \mu^2) p_{\text{el}}(\mu) d\mu = 6\sigma_{\text{el}} (\langle \mu \rangle - \langle \mu^2 \rangle), \end{aligned} \quad (3.18)$$

where $\langle \dots \rangle$ indicates the average value in a single collision. The quantities $\lambda_{\text{el},1}$ and $\lambda_{\text{el},2}$, Eq. (3.16), determine the first and second moments of the multiple-scattering distributions (see Section 4.1). The inverse of the first transport mean free path,

$$\lambda_{\text{el},1}^{-1} = \mathcal{N}\sigma_{\text{el},1} = \frac{2}{\lambda_{\text{el}}} \langle \mu \rangle, \quad (3.19)$$

gives a measure of the average angular deflection per unit path length. By analogy with the “stopping power”, which is defined as the mean energy loss per unit path length (see Section 3.2.3), the quantity $2\lambda_{\text{el},1}^{-1}$ is sometimes called the “scattering power”¹.

¹At high energies, where the scattering is concentrated at very small angles, $\langle \mu \rangle \simeq \langle \theta^2 \rangle / 4$ and $\lambda_{\text{el},1}^{-1} \simeq \langle \theta^2 \rangle / (2\lambda_{\text{el}})$.

3.1.1 Partial-wave cross sections

In the present version of PENELOPE, elastic collisions of electrons and positrons are simulated by using numerical partial-wave cross sections for free neutral atoms. These cross sections were calculated with the program ELSEPA (ELastic Scattering of Electrons and Positrons by Atoms) written by Salvat *et al.* (2005). This program performs accurate relativistic (Dirac) partial-wave calculations of elastic scattering of electrons and positrons by free atoms, and allows consideration of different scattering-potential models. For the calculation of the elastic-scattering database, we used the static-field approximation. The electrostatic potential of the target atom was obtained by considering the Fermi nuclear charge distribution, Eq. (3.1), and the self-consistent Dirac-Fock electron density for free atoms generated by the code of Desclaux (1975). For electron projectiles, the exchange potential of Furness and McCarthy (1973) was adopted. The program ELSEPA is able to perform partial-wave calculations for projectiles with energies up to about 100 MeV. At higher energies, the partial-wave calculation of DCSs is impractical, because of the slow convergence of the series and the accumulation of numerical round-off errors. At these energies, ELSEPA computes elastic-scattering cross sections using a combination of the Born approximation for scattering by the bare finite nucleus and a numerical high-energy screening correction. The cross sections generated by ELSEPA provide the most reliable theoretical description available of elastic scattering for projectiles with kinetic energies higher than about 10 keV.

When the energy of the projectile decreases, the accuracy of the static-field approximation worsens progressively, due to the combined effects of inelastic absorption (which causes a depletion of the elastic-channel wave function) and atomic-charge polarisability (the target atom is polarised by the electric field of the projectile, and the field of the induced dipole acts back on the projectile). These effects can be accounted for by introducing local corrections to the static potential (3.5) [see, *e.g.*, Salvat (2003) and references therein]. Calculations with ELSEPA, including these low-energy effects, yield elastic-scattering DCSs that, at large angles, differ from the static-field DCSs by about 20% at $E = 1$ keV and by up to 50% at $E = 100$ eV. However, differences between total cross sections are much smaller, due to a partial cancellation of the two effects. Unfortunately, both inelastic absorption and dipole polarisation depend strongly on the state of aggregation of the target atom. Owing to the lack of a general theory for the evaluation of these low-energy effects for scattering in condensed materials, the DCSs in the elastic-scattering database were calculated from the static-field approximation. This simplification implies that PENELOPE provides a faithful description of multiple elastic scattering of electrons and positrons with energies higher than about 1 keV; for lower energies, the simulation results should be considered as semi-quantitative.

An extensive database of elastic-scattering DCSs was generated by running the program ELSEPA using the static-field approximation with the electrostatic and exchange potentials described above². This database, which hereafter will be referred to as the

² A similar database (covering the energy range from 50 eV to 300 keV, and generated with a preliminary version of ELSEPA) is distributed by the NIST (Jablonski *et al.*, 2003).

ELSEPA database, consists of 198 ASCII files that contain DCSs, $d\sigma_{\text{el}}/d\Omega$, total cross sections, σ_{el} , and first and second transport cross sections, $\sigma_{\text{el},1}$ and $\sigma_{\text{el},2}$, for elastic scattering of electrons and positrons by neutral atoms of the elements $Z = 1 - 99$. The database files for electrons and positrons are named `eeldxZZZ.p06` and `peldxZZZ.p06`, where `ZZZ` (three digits) is the atomic number of the target atom. The DCSs are tabulated at fixed grids of kinetic energies E and angles θ (the same for all elements). The E -grid is nearly logarithmic, with 96 points (15 points per decade), and spans the interval from 50 eV up to 100 MeV. The angular grid consists of 606 angles, distributed with a higher density near $\theta = 0$ to allow accurate interpolation of the DCS for the highest energies considered. These grids are appropriate for interpolation of the tabulated functions using cubic spline log-log interpolation in E and linear log-log interpolation in $\mu \equiv (1 - \cos \theta)/2$.

Figure 3.2 displays DCSs from the ELSEPA database for elastic scattering of electrons and positrons of various energies by atoms of carbon, silicon, silver and gold. The plots illustrate the variation of the DCS with the atomic number Z , the charge of the projectile and the energy E . Since the interaction $V(r)$ is attractive for electrons and repulsive for positrons, the scattering is more intense for electrons (which can fall deeply into the potential well of the atom) than for positrons (which are repelled from the nucleus and cannot “feel” the inner part of the atom). The DCS for low-energy electrons exhibits a diffraction-like structure, while the DCS for positrons decreases monotonously with the deflection θ , and μ . The Born approximation (see, *e.g.*, Mott and Massey, 1965) predicts a structureless DCS that decreases with μ and is proportional to the squared charge of the projectile (*i.e.*, the same DCS for electrons and positrons). This approximation considers the scattering field as a perturbation (to first order) and, hence, it is valid only for weak fields, *i.e.*, for low- Z elements or, in the case of heavy elements, for large impact parameters (which correspond to small scattering angles). The difference between the (partial-wave) DCSs for electrons and positrons gives a clear indication of the applicability of the Born approximation.

Figure 3.3 shows elastic mean free paths and transport mean free paths for electrons in aluminium and gold. At low energies, the differences between the DCSs of the two elements (see Fig. 3.2) produce very visible differences between the corresponding transport mean free paths. When E increases, the DCS becomes strongly peaked in the forward direction and $\langle \mu^2 \rangle$ becomes much smaller than $\langle \mu \rangle$. In the high-energy limit, $\sigma_{\text{el},2} \simeq 3\sigma_{\text{el},1}$ ($\lambda_{\text{el},2} \simeq \lambda_{\text{el},1}/3$). The total cross section, $\propto 1/(\rho\lambda_{\text{el}})$, decreases monotonously with E to reach a constant value at high energies. This saturation is a relativistic effect: the total cross section measures the interaction probability, which is proportional to the time spent by the projectile within the region where the scattering field is appreciable. This time is determined by the speed of the projectile, which approaches c from below when the projectile energy increases. In the non-relativistic theory, the speed $v_{\text{n.r.}} = (2E/m_e)^{1/2}$ increases without limit with E and the calculated non-relativistic total cross section tends to zero at high energies.

In PENELOPE, the DCSs for compounds (and mixtures) are calculated from atomic total and transport cross sections by means of the additivity approximation (incoherent

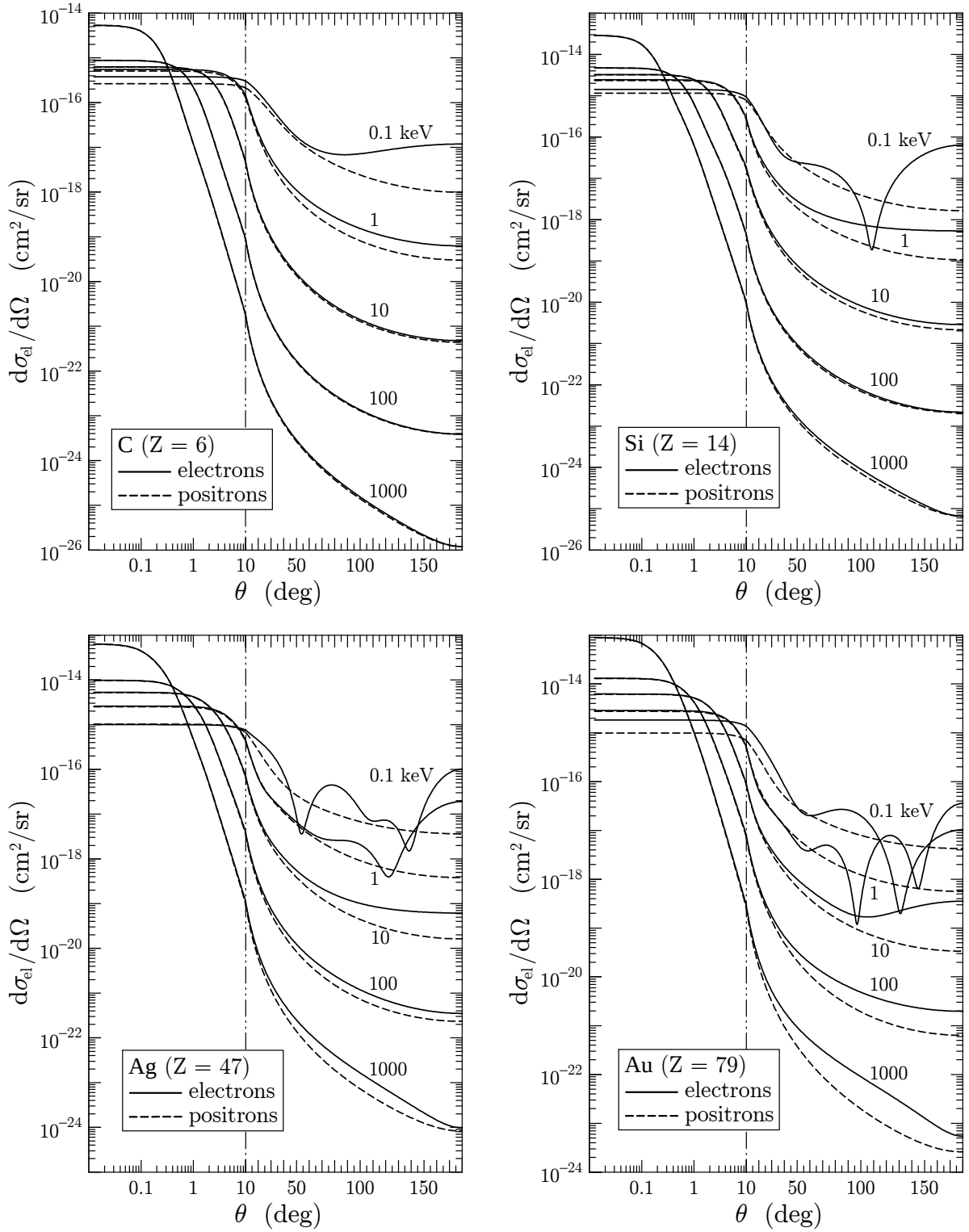


Figure 3.2: DCS for elastic scattering of electrons and positrons by carbon, silicon, silver, and gold atoms as a function of the polar deflection angle θ . Notice the change from logarithmic to linear scale at $\theta = 10$ deg.

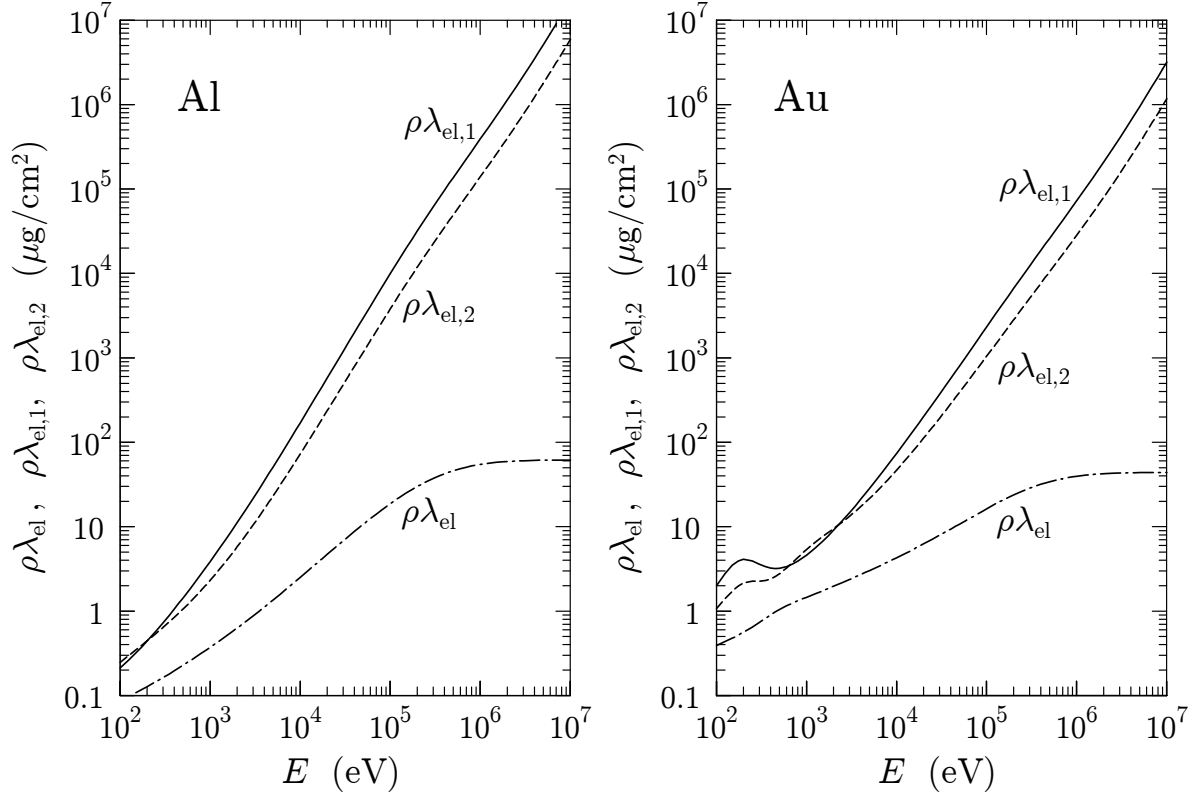


Figure 3.3: Elastic mean free path, λ_{el} , and first and second transport mean free paths, $\lambda_{\text{el},1}$ and $\lambda_{\text{el},2}$, for electrons scattered in aluminium and gold as functions of the kinetic energy of the projectile.

sum of scattered intensities). This amounts to neglecting chemical binding effects. A more accurate approach, which yields a good estimate of these effects, is provided by the following independent-atom approximation (Walker, 1968; Yates, 1968). Assume that the interaction of the projectile with each atom is still given by the free-atom static potential (3.5). The molecular DCS may then be evaluated by adding the waves (not the currents) scattered from the various atoms in the molecule and averaging over molecular orientations. The resulting DCS is given by

$$\frac{d\sigma_{\text{el}}}{d\Omega} = \sum_{i,j} \frac{\sin(qa_{ij}/\hbar)}{qa_{ij}/\hbar} [f_i(\theta)f_j^*(\theta) + g_i(\theta)g_j^*(\theta)], \quad (3.20)$$

where $q = 2\hbar k \sin(\theta/2)$ is the momentum transfer, a_{ij} is the distance between the atoms i and j and f_i , g_i are the scattering amplitudes, Eq. (3.7), for the atom i . It has been claimed that DCSs obtained from this formulation agree with experiments to within $\sim 2\%$ (Walker, 1968; Yates, 1968). DCSs for scattering of 100 eV and 2.5 keV electrons in water vapour, obtained from the simple additivity rule and computed from Eq. (3.20), are compared in Fig. 3.4. It is seen that, for energies above a few keV, chemical binding causes a slight distortion of the DCS at small angles, and a slight

rippling for intermediate angles. Therefore, the use of the additivity approximation (*i.e.*, neglecting chemical binding effects) in Monte Carlo simulation at these energies is justified.

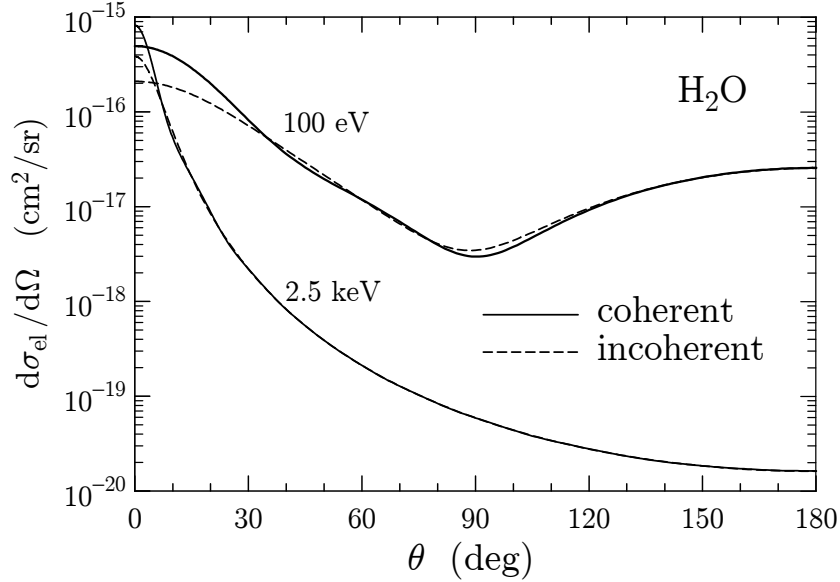


Figure 3.4: DCSs for elastic scattering of electrons by water molecules, calculated as the coherent sum of scattered waves, Eq. (3.20), and from the additivity approximation (incoherent sum).

3.1.1.1 Simulation of single-scattering events

As mentioned above, the angular distribution in single elastic events is axially symmetrical about the direction of incidence. Hence, the azimuthal scattering angle ϕ is sampled uniformly in the interval $(0, 2\pi)$ using the sampling formula $\phi = 2\pi\xi$. The polar deflection μ in each individual collision is generated from the DCSs in the ELSEPA database, by using an interpolation scheme similar to the one described by (Benedito *et al.*, 2001). Partial-wave DCSs will also be utilised for mixed simulations (see Chapter 4), in which only hard events, with deflection μ larger than a given cutoff value μ_c , are sampled individually.

At initialisation time, PENELOPE reads a table of partial-wave DCSs, total cross sections and transport cross sections that, for elemental media, is a duplicate of the corresponding ELSEPA database files. In the case of compounds the table has the same format, but contains molecular cross sections (generated from the atomic cross sections by using the additivity rule). These input cross sections are tabulated at the energies of the ELSEPA grid (which is nearly logarithmic, with 15 points per decade). The first action of PENELOPE is to produce a new table of cross sections for a denser grid of energies, with 200 logarithmically-spaced points (E_i ; $i = 1, \dots, 200$), which spans the

complete energy range used in the simulation. This table is obtained from the input table by means of log-log cubic spline interpolation in E . For each grid energy E_i the program determines the PDF for the angular deflection μ ,

$$p_{\text{el}}(E_i; \mu) = \frac{1}{\sigma_{\text{el}}} \frac{d\sigma_{\text{el}}}{d\mu} = \frac{4\pi}{\sigma_{\text{el}}} \frac{d\sigma_{\text{el}}}{d\Omega}, \quad (3.21)$$

which is tabulated at the 606 angular deflections μ_j in the ELSEPA database. With the aid of log-log cubic spline interpolation in μ , these numerical PDFs are transformed into continuous distributions, from which random values of the angular deflection μ are generated by using the RITA method (see Section 1.2.4).

In detailed simulations (including energy loss events), the kinetic energy of the projectile varies along the track and we need to simulate elastic events for energies E different from those in the simulation grid. In principle, when $E_i < E < E_{i+1}$, we should first interpolate along the energy axis to obtain the PDF $p_{\text{int}}(E, \mu)$ and then perform the random sampling of μ from the interpolated PDF. Evidently, this procedure would be too time consuming. A convenient alternative is to use linear interpolation of the PDF in $\ln E$, which largely simplifies the numerical operations (see below). The accuracy of this interpolation is determined by the spacing of the adopted grid of energies E_i . With the simulation grid used in PENELOPE (200 logarithmically spaced points), interpolation errors are typically smaller than 0.1% in the less favourable cases, *i.e.*, when the energy E lies close to the midpoint of the interval (E_i, E_{i+1}) . In general, these interpolation errors are much smaller than the intrinsic uncertainties of the partial-wave DCS (associated with the simplifications in the underlying physics model) and have a negligible impact on the simulation results.

The interpolated PDF at E can be regarded as a mixture of the PDFs at E_i and E_{i+1} ,

$$p_{\text{el,int}}(E; \mu) = \pi_i p_{\text{el}}(E_i; \mu) + \pi_{i+1} p_{\text{el}}(E_{i+1}; \mu) \quad (3.22)$$

with interpolation weights

$$\pi_i = \frac{\ln E_{i+1} - \ln E}{\ln E_{i+1} - \ln E_i} \quad \text{and} \quad \pi_{i+1} = \frac{\ln E - \ln E_i}{\ln E_{i+1} - \ln E_i}, \quad (3.23)$$

respectively. Note that these weights are positive and add to unity. Therefore the random sampling of μ from $p_{\text{int}}(E; \mu)$ can be performed by using the composition method (see Section 1.2.6). The sampling algorithm proceeds as follows:

- (i) Sample the variable k , which can take the values i or $i+1$ with point probabilities π_i and π_{i+1} , respectively, and
- (ii) Sample μ from the distribution $p_{\text{el}}(E_k; \mu)$ by using the RITA method (Section 1.2.4).

Notice that with this “interpolation by weight” method we only need to sample μ from the tabulated angular PDFs, *i.e.*, for the energies of the grid. This requires storing in

memory only the set of parameters and flags used by the RITA sampling routine for each E_i .

In mixed simulations (see Chapter 4), individual hard events cause angular deflections μ in a certain interval $(\mu_c, 1)$. With the help of a slight modification, the algorithm described above can be used to generate μ -values restricted to that interval. We recall that, with the inverse-transform method, for a given random number ξ we generate an angular deflection $\mu(\xi)$ by solving the sampling equation

$$\int_0^\mu p_{\text{el}}(E; \mu') d\mu' = \xi, \quad (3.24)$$

where the PDF $p_{\text{el}}(E; \mu)$ is assumed to be normalised to unity. Evidently, $\mu(\xi)$ increases monotonically with ξ . A mixed-simulation algorithm is defined by specifying the mean free path $\lambda_{\text{el}}^{(\text{h})}$ between hard elastic events as a function of energy (see Section 4.3). The relation

$$\frac{1}{\lambda_{\text{el}}^{(\text{h})}} = \frac{1}{\lambda_{\text{el}}} \int_{\mu_c}^1 p_{\text{el}}(E; \mu) d\mu, \quad (3.25)$$

then determines the cutoff deflection $\mu_c(E)$. We note that

$$\int_0^{\mu_c} p_{\text{el}}(E; \mu) d\mu = 1 - \frac{\lambda_{\text{el}}}{\lambda_{\text{el}}^{(\text{h})}} \equiv \xi_c. \quad (3.26)$$

Therefore, to generate random values of μ , distributed according to the PDF $p_{\text{el}}(E; \mu)$ restricted to the interval $(\mu_c, 1)$, we can still use the inverse transform (3.24) but with random numbers ξ in the interval $(\xi_c, 1)$. That is, the sampling equation (restricted inverse transform)

$$\int_0^\mu p_{\text{el}}(E; \mu) d\mu = \xi_c + \xi(1 - \xi_c), \quad (3.27)$$

yields values of μ distributed according to $p_{\text{el}}(E; \mu)$ in the interval $(\mu_c, 1)$. In PENELOPE, this restricted inverse transform is performed by using the RITA method (without Walker's aliasing).

3.1.2 The modified Wentzel (MW) model

As mentioned above, the ELSEPA database extends only up to $E = 100$ MeV. Therefore, we need an alternative model to describe elastic events at higher energies. In earlier versions of PENELOPE, elastic scattering was simulated by means of the modified Wentzel (MW) model, which corresponds to an approximate DCS given by a simple analytical expression. The MW model is determined by the values of the total cross section $\sigma_{\text{el}}(E)$ and the first and second transport cross sections $\sigma_{\text{el},1}$ and $\sigma_{\text{el},2}$. Using the code ELSEPA (Salvat *et al.*, 2005), we generated a database that contains these cross sections for electrons and positrons, for atoms of the elements $Z = 1$ –99 and for a grid of energies that ranges from 50 eV to 1 GeV and is dense enough to permit accurate cubic spline log-log interpolation in E . In the energy range below 100 MeV, total cross

sections were calculated by the partial-wave method (they are the same as those in the ELSEPA database); total cross sections for higher energies were generated by using the Born approximation with a high-energy screening correction. In the present version of PENELOPE, the MW model is used to describe elastic events for energies larger than about 100 MeV. It is worth mentioning that, while simulations with partial-wave DCSs require preparatory calculations (which take several seconds for each material), the initialisation of the MW model is very fast. Therefore, the MW model can be useful for trial calculations, when a fast response is desirable and high accuracy is not required. For instance, the code SHOWER (see Section 6.4), which displays radiation showers on the computer screen, uses the MW DCS to simulate elastic events for any energy.

The MW DCS is expressed as

$$\frac{d\sigma_{\text{el}}^{(\text{MW})}}{d\mu} = \sigma_{\text{el}} p_{\text{MW}}(\mu), \quad (3.28)$$

where the single-scattering distribution $p_{\text{MW}}(\mu)$ is defined by a simple analytical expression, with a physically plausible form, depending on two adjustable parameters. These parameters are determined in such a way that the values of $\langle\mu\rangle$ and $\langle\mu^2\rangle$ obtained from $p_{\text{MW}}(\mu)$ are equal to those of the actual (partial-wave) DCS:

$$\langle\mu\rangle_{\text{MW}} \equiv \int_0^1 \mu p_{\text{MW}}(\mu) d\mu = \langle\mu\rangle = \frac{1}{2} \frac{\sigma_{\text{el},1}}{\sigma_{\text{el}}} \quad (3.29)$$

and

$$\langle\mu^2\rangle_{\text{MW}} \equiv \int_0^1 \mu^2 p_{\text{MW}}(\mu) d\mu = \langle\mu^2\rangle = \frac{1}{2} \frac{\sigma_{\text{el},1}}{\sigma_{\text{el}}} - \frac{1}{6} \frac{\sigma_{\text{el},2}}{\sigma_{\text{el}}}. \quad (3.30)$$

Thus, the MW model will give the same mean free path and the same first and second transport mean free paths as the partial-wave DCS. As a consequence (see Chapter 4), detailed simulations using this model will yield multiple-scattering distributions that do not differ significantly from those obtained from the partial-wave DCS, quite irrespectively of other details of the “artificial” distribution $p_{\text{MW}}(\mu)$.

To set the distribution $p_{\text{MW}}(\mu)$, we start from the Wentzel (1927) angular distribution,

$$p_{\text{W},A_0}(\mu) \equiv \frac{A_0(1+A_0)}{(\mu+A_0)^2}, \quad (A_0 > 0) \quad (3.31)$$

which describes the scattering by an exponentially-screened Coulomb field within the Born approximation (see, *e.g.*, Mott and Massey, 1965), that is, it provides a physically plausible angular distribution, at least for light elements or high-energy projectiles. It is also worth mentioning that the multiple-scattering theory of Molière (1947, 1948) can be derived by assuming that electrons scatter according to the Wentzel distribution (see Fernández-Varea *et al.*, 1993b). The first moments of the Wentzel distribution are

$$\langle\mu\rangle_{\text{W},A_0} = \int_0^1 \mu \frac{A_0(1+A_0)}{(\mu+A_0)^2} d\mu = A_0 \left[(1+A_0) \ln \left(\frac{1+A_0}{A_0} \right) - 1 \right] \quad (3.32)$$

and

$$\langle \mu^2 \rangle_{W,A_0} = \int_0^1 \mu^2 \frac{A_0(1+A_0)}{(\mu+A_0)^2} d\mu = A_0 [1 - 2\langle \mu \rangle_{W,A_0}]. \quad (3.33)$$

Let us define the value of the screening constant A_0 so that $\langle \mu \rangle_{W,A_0} = \langle \mu \rangle$. The value of A_0 can be easily calculated by solving Eq. (3.32) numerically, *e.g.*, by the Newton-Raphson method. Usually, we shall have $\langle \mu^2 \rangle_{W,A_0} \neq \langle \mu^2 \rangle$. At low energies, the Wentzel distribution that gives the correct average deflection is too “narrow” [$\langle \mu^2 \rangle_{W,A_0} < \langle \mu^2 \rangle$] for both electrons and positrons and for all the elements]. At high energies, the angular distribution is strongly peaked in the forward direction and the Wentzel distribution becomes too “wide”. This suggests using a modified Wentzel (MW) model obtained by combining a Wentzel distribution with a simple distribution, which takes different forms in these two cases,

- Case I. If $\langle \mu^2 \rangle_{W,A_0} > \langle \mu^2 \rangle$ (the Wentzel distribution is too wide), we take $p_{MW}(\mu)$ as a statistical admixture of the Wentzel distribution and a delta distribution (a zero-width, fixed-scattering-angle process)

$$p_{MW-I}(\mu) = (1-B) p_{W,A}(\mu) + B \delta(\mu - \langle \mu \rangle) \quad (3.34)$$

with

$$A = A_0 \quad \text{and} \quad B = \frac{\langle \mu^2 \rangle_{W,A} - \langle \mu^2 \rangle}{\langle \mu^2 \rangle_{W,A} - \langle \mu \rangle^2}. \quad (3.35)$$

Notice that in this case we usually have $\langle \mu \rangle \ll 1$, so that the delta distribution is at very small angles. Although we have introduced a discrete peak in the DCS, its effect is smeared out by the successive collisions and not visible in the multiple-scattering angular distributions.

- Case II. If $\langle \mu^2 \rangle_{W,A_0} < \langle \mu^2 \rangle$ (the Wentzel distribution is too narrow), we express $p_{MW}(\mu)$ as a statistical admixture of a Wentzel distribution (with A not necessarily equal to A_0) and a triangle distribution in the interval $(1/2, 1)$,

$$p_{MW-II}(\mu) = (1-B) p_{W,A}(\mu) + B 8(\mu - 1/2) \Theta(\mu - 1/2). \quad (3.36)$$

The parameters A and B are obtained from the conditions (3.29) and (3.30), which give

$$\begin{aligned} (1-B) \langle \mu \rangle_{W,A} + B \frac{5}{6} &= \langle \mu \rangle \\ (1-B) \langle \mu^2 \rangle_{W,A} + B \frac{17}{24} &= \langle \mu^2 \rangle. \end{aligned} \quad (3.37)$$

From the first of these equations,

$$B = \frac{\langle \mu \rangle - \langle \mu \rangle_{W,A}}{(5/6) - \langle \mu \rangle_{W,A}}. \quad (3.38)$$

Inserting this value in the second of Eqs. (3.37), we obtain

$$\left(\frac{17}{24} - \langle \mu^2 \rangle \right) \langle \mu \rangle_{W,A} - \left(\frac{5}{6} - \langle \mu \rangle \right) \langle \mu^2 \rangle_{W,A} = \frac{17}{24} \langle \mu \rangle - \frac{5}{6} \langle \mu^2 \rangle. \quad (3.39)$$

For all situations of interest, this equation has a single root A in the interval $(0, A_0)$ and can be easily solved by means of the bisection method. The value of B given by Eq. (3.38) is then positive and less than unity, as required.

In Fig. 3.5 we compare partial-wave DCSs and MW model DCSs for elastic scattering of electrons of various energies by gold atoms. The considered energies correspond to the case-II MW model [so that the distribution $p_{\text{MW}}(\mu)$ is continuous]. We see that the MW model does imitate the partial-wave DCSs, but the differences are significant. Nevertheless, the important fact here is that both DCSs give exactly the same values of σ_{el} , $\langle\mu\rangle$ and $\langle\mu^2\rangle$.

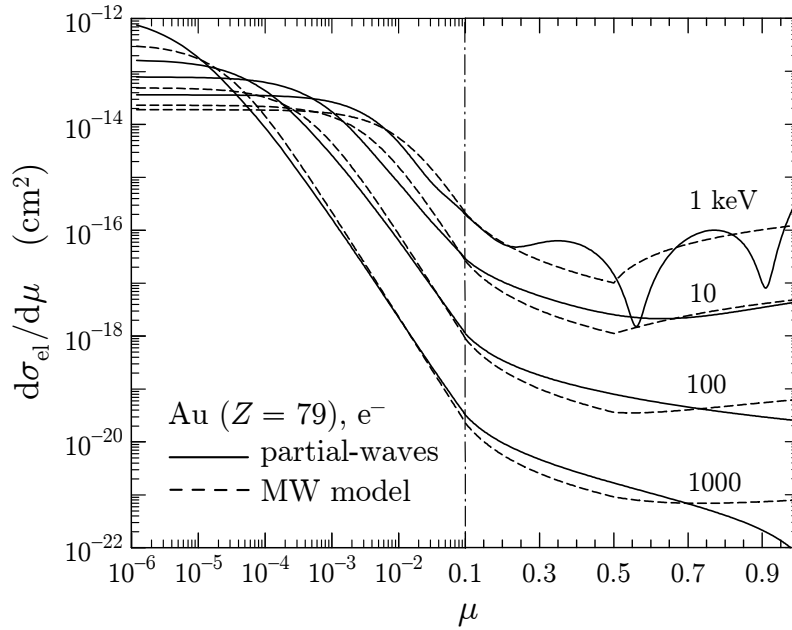


Figure 3.5: Partial-wave and MW model DCSs for elastic scattering of electrons by gold atoms. Notice the change from logarithmic to linear scale at $\mu = 0.1$.

As indicated above, the parameters of the MW model are determined by the characteristic functions $\sigma_{\text{el}}(E)$, $\sigma_{\text{el},1}(E)$ and $\sigma_{\text{el},2}(E)$. PENELOPE reads these functions from the MW database files [99 files named `pdee1ZZ.p06`, where `ZZ` (two digits) is the atomic number of the target atom]. Notice that, with the help of the additivity rule, the MW model can readily be applied to compounds and mixtures. Before starting the simulation, PENELOPE evaluates a table of the parameters A and B , and stores it in the computer memory. Instead of B , PENELOPE tabulates the quantity $B' = +B$ (case I) and $B' = -B$ (case II); this avoids the need to specify the case, which can be inferred from the sign of B' . It is worth noting that A and B' are continuous functions of energy and, therefore, can be rapidly evaluated, for any energy, by interpolation in the stored table. In case I, $\langle\mu\rangle$ coincides with $\langle\mu\rangle_{\text{W},A}$, which is determined by A , Eq. (3.32). Figure 3.6 displays the MW model parameters for aluminium and gold, as representative of low- and high- Z elements. Notice that at high energies, where the case I model applies,

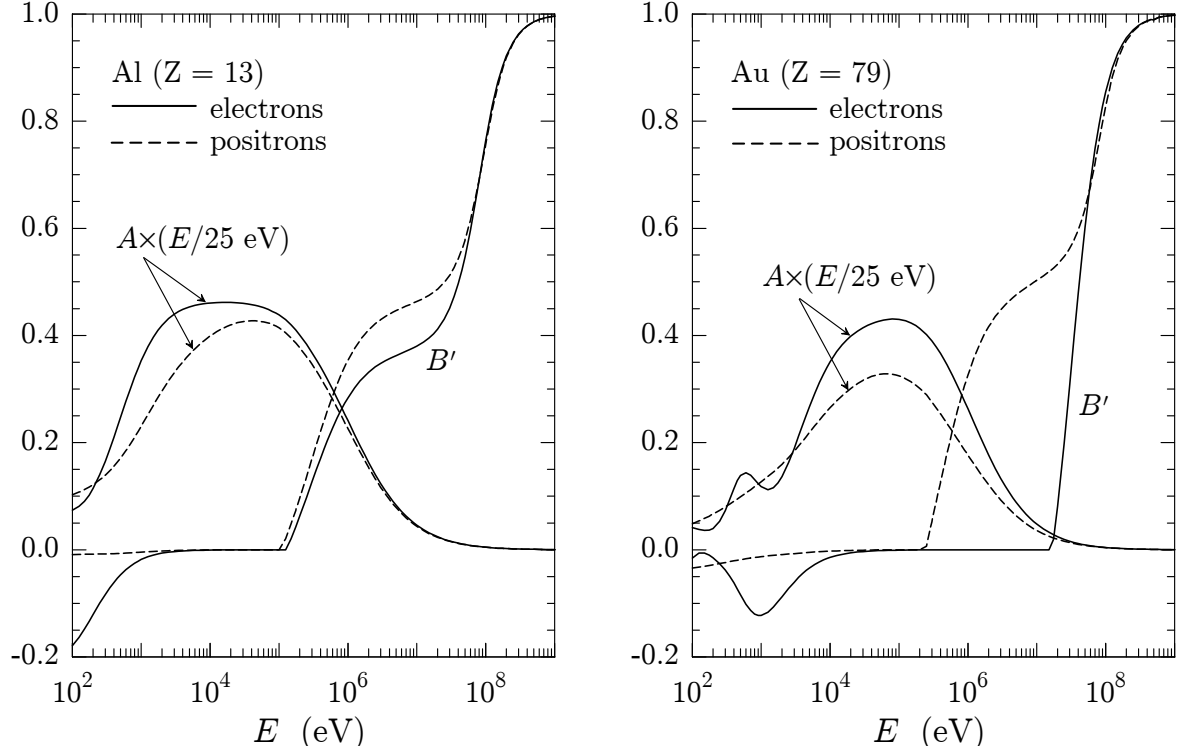


Figure 3.6: Parameters of the MW model for scattering of electrons and positrons by aluminium and gold atoms.

the strength of the delta contribution increases rapidly with energy, indicating that the partial-wave DCS is much narrower than the Wentzel distribution.

3.1.2.1 Simulation of single elastic events with the MW model

The main advantage of using the MW model is that the random sampling of the polar deflection μ can be performed analytically. In the case of mixed simulations (see Chapter 4), only hard events, with deflection μ larger than a given cutoff value μ_c , are sampled individually. In this Section we describe analytical (*i.e.*, exact) methods for random sampling of μ in the restricted interval $(\mu_c, 1)$. The azimuthal scattering angle ϕ is sampled uniformly in the interval $(0, 2\pi)$.

- Case I. The cumulative distribution function of $p_{\text{MW-I}}(\mu)$ is

$$\mathcal{P}_{\text{MW-I}}(\mu) \equiv \int_0^\mu p_{\text{MW-I}}(\mu') d\mu' = \begin{cases} (1-B) \frac{(1+A)\mu}{A+\mu} & \text{if } 0 \leq \mu < \langle \mu \rangle, \\ B + (1-B) \frac{(1+A)\mu}{A+\mu} & \text{if } \langle \mu \rangle \leq \mu \leq 1. \end{cases} \quad (3.40)$$

Because of the analytical simplicity of this function, the random sampling of μ can be performed by using the inverse-transform method (Section 1.2.2). The sampling

equation for μ in (0,1) reads

$$\mu = \mathcal{P}_{\text{MW-I}}^{-1}(\xi), \quad (3.41)$$

where $\mathcal{P}_{\text{MW-I}}^{-1}(\xi)$ is the inverse of the cumulative distribution function, which is given by

$$\mathcal{P}_{\text{MW-I}}^{-1}(\xi) = \begin{cases} \frac{\xi A}{(1-B)(1+A) - \xi} & \text{if } 0 \leq \xi < \xi_0, \\ \langle \mu \rangle & \text{if } \xi_0 \leq \xi < \xi_0 + B, \\ \frac{(\xi - B)A}{(1-B)(1+A) - (\xi - B)} & \text{if } \xi_0 + B \leq \xi \leq 1, \end{cases} \quad (3.42)$$

with

$$\xi_0 = (1-B) \frac{(1+A)\langle \mu \rangle}{A + \langle \mu \rangle}. \quad (3.43)$$

To sample μ in the restricted interval $(\mu_c, 1)$, we can still use the inverse-transform method, Eq. (3.41), but with the random number ξ sampled uniformly in the interval $(\xi_c, 1)$ with

$$\xi_c = \mathcal{P}_{\text{MW-I}}(\mu_c). \quad (3.44)$$

- Case II. The cumulative distribution function is

$$\begin{aligned} \mathcal{P}_{\text{MW-II}}(\mu) &\equiv \int_0^\mu p_{\text{MW-II}}(\mu') d\mu' \\ &= \begin{cases} (1-B) \frac{(1+A)\mu}{A + \mu} & \text{if } 0 \leq \mu < \frac{1}{2}, \\ (1-B) \frac{(1+A)\mu}{A + \mu} + B 4 \left[\mu^2 - \mu + \frac{1}{4} \right] & \text{if } \frac{1}{2} \leq \mu \leq 1. \end{cases} \end{aligned} \quad (3.45)$$

In principle, to sample μ in (0,1), we can adopt the inverse-transform method. The sampling equation

$$\xi = \mathcal{P}_{\text{MW-II}}(\mu) \quad (3.46)$$

can be cast in the form of a cubic equation. This equation can be solved either by using the analytical solution formulas for the cubic equation, which are somewhat complicated, or numerically, *e.g.*, by the Newton-Raphson method. We employ this last procedure to determine the cutoff deflection for mixed simulation (see Section 4.1). To sample μ in the restricted interval $(\mu_c, 1)$ we use the composition method, which is easier than solving Eq. (3.46). Notice that the sampling from the (restricted) Wentzel and from the triangle distributions can be performed analytically by the inverse-transform method.

3.2 Inelastic collisions

The dominant energy loss mechanisms for electrons and positrons with intermediate and low energies are inelastic collisions, *i.e.*, interactions that produce electronic excitations

and ionisations in the medium. The quantum theory of inelastic collisions of charged particles with individual atoms and molecules was first formulated by Bethe (1930, 1932) on the basis of the first-order (plane-wave) Born approximation. The extension of the theory to inelastic collisions in condensed materials has been discussed by Fano (1963). The formal aspects of the quantum theory for condensed matter are quite complicated. Fortunately, the results are essentially equivalent to those from classical dielectric theory.

The effect of individual inelastic collisions *on the projectile* is completely specified by giving the energy loss W and the polar and azimuthal scattering angles θ and ϕ , respectively. For amorphous media with randomly oriented atoms (or molecules), the DCS for inelastic collisions is independent of the azimuthal scattering angle ϕ . Instead of the polar scattering angle θ , it is convenient to use the recoil energy Q [see Eqs. (A.29) and (A.30)], defined by

$$Q(Q + 2m_e c^2) = (cq)^2. \quad (3.47)$$

The quantity q is the magnitude of the momentum transfer $\mathbf{q} \equiv \mathbf{p} - \mathbf{p}'$, where $\mathbf{p}$ and $\mathbf{p}'$ are the linear momenta of the projectile before and after the collision. Notice that Q is the kinetic energy of an electron that moves with a linear momentum equal to q .

Let us first consider the inelastic interactions of electrons or positrons ($z_0^2 = 1$) with an isolated atom (or molecule) containing Z electrons in its ground state. The DCS for collisions with energy loss W and recoil energy Q , obtained from the first Born approximation, can be written in the form (Fano, 1963)

$$\frac{d^2\sigma_{\text{in}}}{dW dQ} = \frac{2\pi z_0^2 e^4}{m_e v^2} \left(\frac{2m_e c^2}{WQ(Q + 2m_e c^2)} + \frac{\beta^2 \sin^2 \theta_r W 2m_e c^2}{[Q(Q + 2m_e c^2) - W^2]^2} \right) \frac{df(Q, W)}{dW}, \quad (3.48)$$

where $v = \beta c$ is the velocity of the projectile. θ_r is the angle between the initial momentum of the projectile and the momentum transfer, which is given by Eq. (A.42),

$$\cos^2 \theta_r = \frac{W^2/\beta^2}{Q(Q + 2m_e c^2)} \left(1 + \frac{Q(Q + 2m_e c^2) - W^2}{2W(E + m_e c^2)} \right)^2. \quad (3.49)$$

The result (3.48) is obtained in the Coulomb gauge (Fano, 1963); the two terms on the right-hand side are the contributions from interactions through the instantaneous (longitudinal) Coulomb field and through the exchange of virtual photons (transverse field), respectively. The factor $df(Q, W)/dW$ is the atomic generalised oscillator strength (GOS), which completely determines the effect of inelastic interactions *on the projectile*, within the Born approximation. Notice, however, that knowledge of the GOS does not suffice to describe the energy spectrum and angular distribution of secondary knock-on electrons (delta rays).

The GOS can be represented as a surface over the (Q, W) plane, which is called the Bethe surface (see Inokuti, 1971; Inokuti *et al.*, 1978). Unfortunately, the GOS is known in analytical form only for two simple systems, namely, the (non-relativistic) hydrogenic ions (see Fig. 3.7) and the free-electron gas. Even in these cases, the analytical expressions of the GOSs are too complicated for simulation purposes. For ionisation of inner shells, the GOS can be computed numerically from first principles (see, *e.g.*, Manson,

1972), but using GOSs defined through extensive numerical tables is impractical for Monte Carlo simulation. Fortunately, the physics of inelastic collisions is largely determined by a few global features of the Bethe surface. Relatively simple GOS models can be devised that are consistent with these features and, therefore, lead to a fairly realistic description of inelastic interactions (see, *e.g.*, Salvat and Fernández-Varea, 1992).

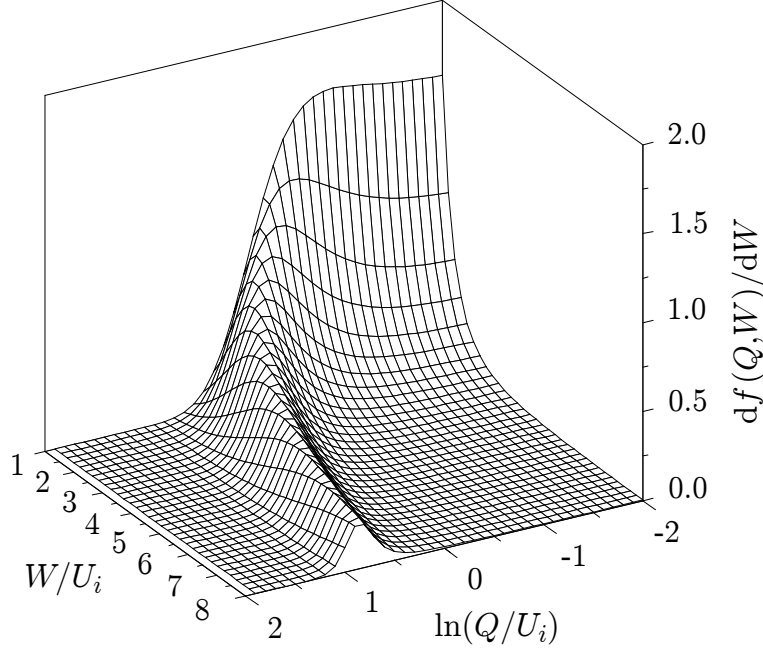


Figure 3.7: The GOS for ionisation of the hydrogen atom ($Z = 1$) in the ground state. All energies are in units of the ionisation energy $U_i = 13.6$ eV. The GOS for ionisation of (non-relativistic) hydrogenic ions is independent of Z if energies are expressed in units of the ionisation energy.

As mentioned above, the “atomic” DCS for inelastic interactions in dense media can be obtained from a semiclassical treatment in which the medium is considered as a dielectric, characterised by a complex dielectric function $\epsilon(k, \omega)$, which depends on the wave number k and the frequency ω . In the classical picture, the (external) electric field of the projectile polarizes the medium producing an induced electric field that causes the slowing down of the projectile. The dielectric function relates the Fourier components of the total (external+induced) and the external electric potentials. It is convenient to interpret the quantities $q = \hbar k$ and $W = \hbar \omega$ as the momentum and energy transfers and consider that the dielectric function depends on the variables Q [defined by Eq. (3.47)] and W . The DCSs obtained from the dielectric and quantum treatments are consistent (*i.e.*, the former reduces to the latter for a low-density medium) if one assumes the identity

$$\frac{df(Q, W)}{dW} \equiv W \frac{Q + m_e c^2}{m_e c^2} \frac{2Z}{\pi \Omega_p^2} \text{Im} \left(\frac{-1}{\epsilon(Q, W)} \right), \quad (3.50)$$

where Ω_p is the plasma energy of a free-electron gas with the electron density of the

medium, given by

$$\Omega_p^2 = 4\pi\mathcal{N}Z\hbar^2e^2/m_e. \quad (3.51)$$

Eq. (3.50) establishes the connection between the atomic GOS (a property of individual atoms) and the dielectric function (a macroscopic concept). The DCS for the condensed medium can be expressed in the form [cf. Eq. (3.48)],

$$\begin{aligned} \frac{d^2\sigma_{\text{in}}}{dW dQ} = & \frac{2\pi z_0^2 e^4}{m_e v^2} \frac{df(Q, W)}{dW} \left(\frac{2m_e c^2}{WQ(Q + 2m_e c^2)} \right. \\ & \left. + \left\{ \frac{\beta^2 \sin^2 \theta_r W 2m_e c^2}{[Q(Q + 2m_e c^2) - W^2]^2} - \mathcal{D}(Q, W) \right\} \right), \end{aligned} \quad (3.52)$$

where the term $\mathcal{D}(Q, W)$, which is appreciable only for small Q , accounts for the so-called density-effect correction (Sternheimer, 1952). The origin of this term is the polarisability of the medium, which “screens” the distant transverse interaction causing a net reduction of its contribution to the stopping power. The density-effect correction $\mathcal{D}(Q, W)$ is determined by the dielectric function that, in turn, is related to the GOS. Thus, the GOS contains all the information needed to compute the DCS for electron/positron inelastic interactions in condensed media.

In the limit of very large recoil energies, the binding and momentum distribution of the target electrons have a small effect on the interaction. Therefore, in the large- Q region, the target electrons behave as if they were essentially free and at rest and, consequently, the GOS reduces to a ridge along the line $W = Q$, which was named the Bethe ridge by Inokuti (1971). In the case of hydrogenic ions in the ground state, Fig. 3.7, the Bethe ridge becomes clearly visible at relatively small recoil energies, of the order of the ionisation energy U_i . For smaller Q 's, the structure of the Bethe surface is characteristic of the material. In the limit $Q \rightarrow 0$, the GOS reduces to the optical oscillator strength (OOS),

$$\frac{df(W)}{dW} \equiv \frac{df(Q = 0, W)}{dW}, \quad (3.53)$$

which is closely related to the (dipole) photoelectric cross section for photons of energy W (Fano, 1963). Experimental information on the OOS is provided by measurements of either photoelectric cross sections or dielectric functions (see, *e.g.*, Fernández-Varea *et al.*, 1993a). The GOS satisfies the Bethe sum rule (Inokuti, 1971)

$$\int_0^\infty \frac{df(Q, W)}{dW} dW = Z \quad \text{for any } Q. \quad (3.54)$$

This sum rule, which is a result from non-relativistic theory (see, *e.g.*, Mott and Massey, 1965), is assumed to be generally satisfied. It leads to the interpretation of the GOS as the effective number of electrons per unit energy transfer that participate in interactions with given recoil energy Q . The mean excitation energy I , defined by (Fano, 1963; Inokuti, 1971)

$$Z \ln I = \int_0^\infty \ln W \frac{df(W)}{dW} dW, \quad (3.55)$$

plays a central role in the Bethe stopping power formula [Eq. (3.112)]. This quantity has been determined empirically for a large number of materials (see Berger and Seltzer, 1982, and references therein) from measurements of the stopping power of heavy charged particles and/or from experimental optical dielectric functions. In the following, we shall assume that the mean excitation energy of the stopping medium is known.

3.2.1 GOS model

The simulation of inelastic collisions of electrons and positrons in PENELOPE is performed on the basis of the following GOS model, which is tailored to allow fast random sampling of W and Q . We assume that the GOS splits into contributions from the different atomic electron shells. Each atomic shell k is characterised by the number Z_k of electrons in the shell and the ionisation energy U_k . To model the contribution of a shell to the GOS, we refer to the example of the hydrogen atom (Fig. 3.7) and observe that for $Q > U_k$ the GOS reduces to the Bethe ridge, whereas for $Q < U_k$ it is nearly constant with Q and decreases rapidly with W ; a large fraction of the OOS concentrates in a relatively narrow W -interval. Consideration of other well-known systems, such as inner shells of heavy atoms (Manson, 1972) and the free-electron gas (Lindhard and Winther, 1964), shows that these gross features of the GOS are universal. Liljequist (1983) proposed modelling the GOS of each atomic electron shell as a single “ δ -oscillator”, which is an entity with a simple GOS given by (see Fig. 3.8)

$$F(W_k; Q, W) = \delta(W - W_k)\Theta(W_k - Q) + \delta(W - Q)\Theta(Q - W_k), \quad (3.56)$$

where $\delta(x)$ is the Dirac delta function and $\Theta(x)$ is the step function. The first term represents resonant low- Q (distant) interactions, which are described as a single resonance at the energy W_k . The second term corresponds to large- Q (close) interactions, in which the target electrons react as if they were free and at rest ($W = Q$). Notice that the oscillator GOS satisfies the sum rule

$$\int_0^\infty F(W_k; Q, W) dW = 1 \quad \text{for any } Q \quad (3.57)$$

and, consequently, a δ -oscillator corresponds to one electron in the target. The Liljequist GOS model for the whole atom is given by

$$\frac{df(Q, W)}{dW} = \sum_k f_k [\delta(W - W_k)\Theta(W_k - Q) + \delta(W - Q)\Theta(Q - W_k)]. \quad (3.58)$$

where the summation in k extends over all bound electron shells (and the conduction band, in the case of conductors) and the partial oscillator strength f_k is identified with the number of electrons in the k -th shell, *i.e.*, $f_k = Z_k$. The corresponding OOS reduces to

$$\frac{df(W)}{dW} = \sum_k f_k \delta(W - W_k), \quad (3.59)$$

which has the same form (a superposition of resonances) as the OOS used by Sternheimer (1952) in his calculations of the density effect correction. In order to reproduce the high-energy stopping power given by the Bethe formula (Berger and Seltzer, 1982), the oscillator strengths must satisfy the Bethe sum rule (3.54),

$$\sum_k f_k = Z, \quad (3.60)$$

and the excitation energies must be defined in such a way that the GOS model leads, through Eq. (3.55), to the accepted value of the mean excitation energy I ,

$$\sum_k f_k \ln W_k = Z \ln I. \quad (3.61)$$

As the partial oscillator strength f_k has been set equal to the number of electrons in the k -th shell, the Bethe sum rule is automatically satisfied.

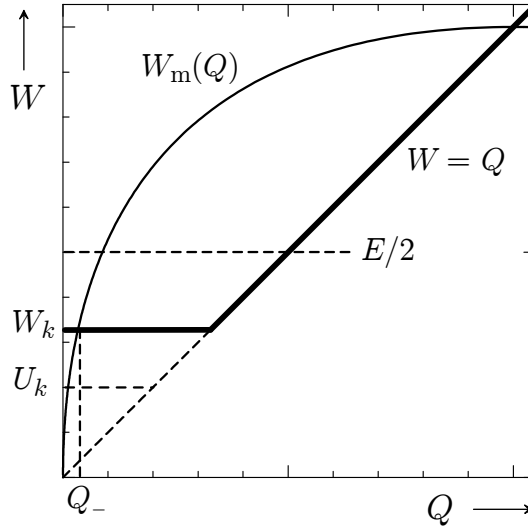


Figure 3.8: Oscillator model for the GOS of an inner shell with $U_k = 2$ keV. The continuous curve represents the maximum allowed energy loss as a function of the recoil energy, $W_m(Q)$, for electrons/positrons with $E = 10$ keV. For distant interactions the possible recoil energies lie in the interval from Q_- to W_k . Recoil energies larger than W_k correspond to close interactions. The largest allowed energy loss $W_{\max}$ is $E/2$ for electrons and E for positrons (see text).

The largest contribution to the total cross section arises from low- W (soft) excitations. Therefore, the total cross section is mostly determined by the OOS of weakly bound electrons, which is strongly dependent on the state of aggregation. In the case of conductors and semiconductors, electrons in the outermost shells form the conduction band (cb). These electrons can move quite freely through the medium and, hence, their binding energy is set to zero, $U_{cb} = 0$. Excitations of the conduction band will be described by a single oscillator, with oscillator strength f_{cb} and resonance energy W_{cb} . These parameters should be identified with the effective number of electrons (per atom

or molecule) that participate in plasmon excitations and the plasmon energy, respectively. They can be estimated, *e.g.*, from electron energy-loss spectra or from measured optical data. When this information is not available, we will simply fix the value of f_{cb} (as the number of electrons with ionisation energies less than, say, 15 eV) and set the resonance energy W_{cb} equal to the plasmon energy of a free-electron gas with the same density as that of conduction electrons,

$$W_{\text{cb}} = \sqrt{4\pi\mathcal{N}f_{\text{cb}}\hbar^2e^2/m_e} = \sqrt{\frac{f_{\text{cb}}}{Z}} \Omega_p. \quad (3.62)$$

This gives a fairly realistic model for free-electron-like metals (such as aluminium), because the resonance energy is set equal to the plasmon energy of the free-electron gas (see, *e.g.*, Kittel, 1976). A similar approach, with f_{cb} set equal to the lowest chemical valence of an element, was adopted by Sternheimer *et al.* (1982, 1984) in their calculations of the density effect correction for single-element metals.

Following Sternheimer (1952), the resonance energy of a bound-shell oscillator is expressed as

$$W_j = \sqrt{(aU_j)^2 + \frac{2}{3} \frac{f_j}{Z} \Omega_p^2}, \quad (3.63)$$

where U_j is the ionisation energy and Ω_p is the plasma energy corresponding to the total electron density in the material, Eq. (3.51). The term $2f_j\Omega_p^2/3Z$ under the square root accounts for the Lorentz-Lorenz correction (the resonance energies of a condensed medium differ from those of a free atom/molecule. The empirical adjustment factor a in Eq. (3.63) (the same for all bound shells) is determined from the condition (3.61), *i.e.*, from

$$Z \ln I = f_{\text{cb}} \ln W_{\text{cb}} + \sum_j f_j \ln \sqrt{(aU_j)^2 + \frac{2}{3} \frac{f_j}{Z} \Omega_p^2}. \quad (3.64)$$

For a one-shell system, such as the hydrogen atom, relations (3.60) and (3.61) imply that the resonance energy W_i is equal to I . Considering the $\sim W^{-3}$ dependence of the hydrogenic OOS, it is concluded that a should be of the order of $\exp(1/2) = 1.65$ (Sternheimer *et al.*, 1982). It is worth noting that the Sternheimer adjustment factor a is a characteristic of the considered medium; therefore, the DCSs for excitation of a shell of a given element in two different compounds may be slightly different.

The present GOS model is completely determined by the ionisation energy U_i and the number of electrons Z_i of each electron shell, and by the empirical value of the mean excitation energy I . In PENELOPE, the target atoms are assumed to be in their ground state configuration, and the experimental ionisation energies compiled by Lederer and Shirley (1978) are used. The adopted I -values are those proposed by Berger and Seltzer (1982), which were also used to generate the ICRU (1984) tables of stopping powers for electrons and positrons. Therefore, our inelastic collision model yields stopping powers that are consistent with the ICRU recommended values (see below).

It should be mentioned that the oscillator model gives a Bethe ridge with zero width, *i.e.*, the broadening caused by the momentum distribution of the target electrons is ne-

glected. This is not a serious drawback for light projectiles (electrons and positrons), but it can introduce sizeable errors in the computed cross sections for slow heavy projectiles with $m \gg m_e$. The oscillator model also disregards the fact that, for low- Q interactions, there is a transfer of oscillator strength from inner to outer shells (see, *e.g.*, Shiles *et al.*, 1980). As a consequence, the shell ionisation cross sections obtained from this GOS model are only roughly approximate. Their use in a Monte Carlo code is permissible only because the ionisation of inner shells is a low-probability process (see Fig. 3.9 below) that has a very weak effect on the global transport properties. In what follows, K, L and M shells with ionisation energies U_j larger than the absorption energies of electrons and photons, **EABS(1)** and **EABS(2)** (see Section 6.1.2), will be referred to as “inner” shells. Electron shells other than K, L and M shells, or with $U_j < \min(\mathbf{EABS}(1), \mathbf{EABS}(2))$, will be referred to as “outer” shells.

In mixed (class II) simulations, only hard collisions, with energy loss larger than a specified cutoff value W_{cc} , are simulated (see Chapter 4). The effect of soft interactions (with $W < W_{cc}$) is described by means of a multiple-scattering approximation, which does not require detailed knowledge of the shell DCSs. Hard collisions may produce ionisations in deep electron shells, which leave the target atom in a highly excited state (with a vacancy in an inner shell) that decays by emission of energetic x-rays and Auger electrons. In PENELOPE we use the GOS model only to describe the effect of the interactions on the projectile and the emission of knock-on secondary electrons. The production of vacancies in inner shells, and their subsequent de-excitation, is simulated by using more accurate ionisation cross sections (see Section 3.2.6).

The present GOS model is directly applicable to compounds (and mixtures), since the oscillators may pertain either to atoms or molecules. When the value of the mean excitation energy of the compound is not known, it may be estimated from Bragg’s additivity rule as follows. Consider a compound X_xY_y , in which the molecules consist of x atoms of the element X and y atoms of the element Y . The number of electrons per molecule is $Z_M = xZ_X + yZ_Y$, where Z_X stands for the atomic number of element X . According to the additivity rule, the GOS of the compound is approximated as the sum of the atomic GOSs of the atoms so that

$$Z_M \ln I = xZ_X \ln I_X + yZ_Y \ln I_Y, \quad (3.65)$$

where I_X denotes the mean excitation energy of element X .

For heavy elements, and also for compounds and mixtures with several elements, the number of electron shells may be fairly large (of the order of sixty for an alloy of two heavy metals). In these cases, it would be impractical to treat all shells with the same detail/accuracy. In fact, the description of the outer shells can be simplified without sacrificing the reliability of the simulation results. In PENELOPE, the maximum number of oscillators for each material is limited. When the number of actual shells is too large, oscillators with similar resonance energies are grouped together and replaced by a single oscillator with oscillator strength equal to the sum of strengths of the original oscillators. The resonance energy of the group oscillator is set by requiring that its contribution to the mean excitation energy I equals the sum of contributions of the grouped oscillators;

this ensures that grouping will not alter the stopping power of fast particles (with E substantially greater than the ionisation energy of the grouped oscillators).

3.2.2 Differential cross sections

The DCS for inelastic collisions obtained from our GOS model can be split into contributions from distant longitudinal, distant transverse and close interactions,

$$\frac{d^2\sigma_{\text{in}}}{dW dQ} = \frac{d^2\sigma_{\text{dis,l}}}{dW dQ} + \frac{d^2\sigma_{\text{dis,t}}}{dW dQ} + \frac{d^2\sigma_{\text{clo}}}{dW dQ}. \quad (3.66)$$

The DCS for distant longitudinal interactions is given by the first term in Eq. (3.52),

$$\frac{d^2\sigma_{\text{dis,l}}}{dW dQ} = \frac{2\pi e^4}{m_e v^2} \sum_k f_k \frac{1}{W} \frac{2m_e c^2}{Q(Q + 2m_e c^2)} \delta(W - W_k) \Theta(W_k - Q). \quad (3.67)$$

As mentioned above, the DCS for distant transverse interactions has a complicated expression. To simplify it, we shall ignore the (very small) angular deflections of the projectile in these interactions and replace the expression in curly brackets in Eq. (3.52) by an averaged W -independent value that gives the exact contribution of the distant transverse interactions to the high-energy stopping power (Salvat and Fernández-Varea, 1992). This yields the following approximate expression for the DCS of distant transverse interactions,

$$\begin{aligned} \frac{d^2\sigma_{\text{dis,t}}}{dW dQ} &= \frac{2\pi e^4}{m_e v^2} \sum_k f_k \frac{1}{W} \left\{ \ln \left(\frac{1}{1 - \beta^2} \right) - \beta^2 - \delta_F \right\} \\ &\times \delta(W - W_k) \Theta(W_k - Q) \delta(Q - Q_-), \end{aligned} \quad (3.68)$$

where Q_- is the minimum recoil energy³ for the energy transfer W , Eq. (A.31), and δ_F is the Fermi density effect correction on the stopping power, which has been studied extensively in the past (Sternheimer, 1952; Fano, 1963). δ_F can be computed as (Fano, 1963)

$$\delta_F \equiv \frac{1}{Z} \int_0^\infty \frac{df(Q=0, W)}{dW} \ln \left(1 + \frac{L^2}{W^2} \right) dW - \frac{L^2}{\Omega_p^2} (1 - \beta^2), \quad (3.69)$$

where L is a real-valued function of β^2 defined as the positive root of the following equation (Inokuti and Smith, 1982):

$$\mathcal{F}(L) \equiv \frac{1}{Z} \Omega_p^2 \int_0^\infty \frac{1}{W^2 + L^2} \frac{df(Q=0, W)}{dW} dW = 1 - \beta^2. \quad (3.70)$$

The function $\mathcal{F}(L)$ decreases monotonically with L , and hence, the root $L(\beta^2)$ exists only when $1 - \beta^2 < \mathcal{F}(0)$; otherwise it is $\delta_F = 0$. Therefore, the function $L(\beta^2)$ starts

³The recoil energy Q_- corresponds to $\theta = 0$, *i.e.*, we consider that the projectile is not deflected by distant transverse interactions.

with zero at $\beta^2 = 1 - \mathcal{F}(0)$ and grows monotonically with increasing β^2 . With the OOS, given by Eq. (3.59), we have

$$\mathcal{F}(L) = \frac{1}{Z} \Omega_p^2 \sum_k \frac{f_k}{W_k^2 + L^2} \quad (3.71)$$

and

$$\delta_F \equiv \frac{1}{Z} \sum_k f_k \ln \left(1 + \frac{L^2}{W_k^2} \right) - \frac{L^2}{\Omega_p^2} (1 - \beta^2). \quad (3.72)$$

In the high-energy limit ($\beta \rightarrow 1$), the L value resulting from Eq. (3.70) is large ($L \gg W_k$) and can be approximated as $L^2 = \Omega_p^2 / (1 - \beta^2)$. Then, using the Bethe sum rule ($\sum f_k = Z$) and the relation (3.61), we obtain

$$\delta_F \simeq \ln \left(\frac{\Omega_p^2}{(1 - \beta^2) I^2} \right) - 1, \quad \text{when } \beta \rightarrow 1. \quad (3.73)$$

The DCS for close collisions is given by

$$\frac{d^2 \sigma_{\text{clo}}}{dW dQ} = \frac{2\pi e^4}{m_e v^2} \sum_k f_k \frac{1}{W} \left(\frac{2m_e c^2}{W(W + 2m_e c^2)} + \frac{\beta^2 \sin^2 \theta_{\text{clo}}}{2m_e c^2} \right) \delta(W - Q) \Theta(W - W_k),$$

where θ_{clo} is the recoil angle, defined by Eq. (3.49) with $Q = W$,

$$\cos^2 \theta_{\text{clo}} = \frac{W}{E} \frac{E + 2m_e c^2}{W + 2m_e c^2}. \quad (3.74)$$

We have

$$\frac{d^2 \sigma_{\text{clo}}}{dW dQ} = \frac{2\pi e^4}{m_e v^2} \sum_k f_k \frac{1}{W^2} \left(1 + \frac{\beta^2 (E - W)W - EW}{E(W + 2m_e c^2)} \right) \delta(W - Q) \Theta(W - W_k). \quad (3.75)$$

3.2.2.1 DCS for close collisions of electrons

When the projectile is an electron, the DCS must be corrected to account for the indistinguishability of the projectile and the target electrons. For distant interactions, the effect of this correction is small (much smaller than the distortion introduced by our modelling of the GOS) and will be neglected. The energy loss DCS for binary collisions of electrons with free electrons at rest, obtained from the Born approximation with proper account of exchange, is given by the Møller (1932) formula,

$$\begin{aligned} \frac{d^2 \sigma_M}{dW dQ} = & \frac{2\pi e^4}{m_e v^2} \frac{1}{W^2} \left[1 + \left(\frac{W}{E - W} \right)^2 - \frac{W}{E - W} \right. \\ & \left. + a \left(\frac{W}{E - W} + \frac{W^2}{E^2} \right) \right] \delta(W - Q), \end{aligned} \quad (3.76)$$

where

$$a = \left(\frac{E}{E + m_e c^2} \right)^2 = \left(\frac{\gamma - 1}{\gamma} \right)^2. \quad (3.77)$$

To introduce exchange effects in the DCS for close interactions of electrons, we replace the factor in parenthesis in Eq. (3.75) by the analogous factor in Møller's formula, *i.e.*, we take

$$\frac{d^2 \sigma_{\text{clo}}^{(-)}}{dW dQ} = \frac{2\pi e^4}{m_e v^2} \sum_k f_k \frac{1}{W^2} F^{(-)}(E, W) \delta(W - Q) \Theta(W - W_k), \quad (3.78)$$

with

$$F^{(-)}(E, W) \equiv 1 + \left(\frac{W}{E - W} \right)^2 - \frac{W}{E - W} + a \left(\frac{W}{E - W} + \frac{W^2}{E^2} \right). \quad (3.79)$$

In the final state we have two indistinguishable free electrons, and it is natural to consider the fastest one as the “primary”. Accordingly, the maximum allowed energy transfer in close collisions is

$$W_{\text{max}} = E/2. \quad (3.80)$$

3.2.2.2 DCS for close collisions of positrons

Positrons in matter are unstable particles that annihilate with electrons giving photons (see Section 3.4). On the other hand, electron-positron pairs can be created if enough electromagnetic energy ($> 2m_e c^2$) is available (either from real or virtual photons). A positron does not interact with matter as a usual (stable) positively charged particle, since the competing process of annihilation followed by re-creation can cause the same transitions as “direct” scattering (see, *e.g.*, Sakurai, 1967). The DCS for binary collisions of positrons with free electrons at rest, obtained from the first Born approximation including the “annihilation/creation” mechanism, is given by the Bhabha (1936) formula,

$$\frac{d^2 \sigma_B}{dW dQ} = \frac{2\pi e^4}{m_e v^2} \frac{1}{W^2} \left[1 - b_1 \frac{W}{E} + b_2 \left(\frac{W}{E} \right)^2 - b_3 \left(\frac{W}{E} \right)^3 + b_4 \left(\frac{W}{E} \right)^4 \right] \delta(W - Q), \quad (3.81)$$

where

$$\begin{aligned} b_1 &= \left(\frac{\gamma - 1}{\gamma} \right)^2 \frac{2(\gamma + 1)^2 - 1}{\gamma^2 - 1}, & b_2 &= \left(\frac{\gamma - 1}{\gamma} \right)^2 \frac{3(\gamma + 1)^2 + 1}{(\gamma + 1)^2}, \\ b_3 &= \left(\frac{\gamma - 1}{\gamma} \right)^2 \frac{2\gamma(\gamma - 1)}{(\gamma + 1)^2}, & b_4 &= \left(\frac{\gamma - 1}{\gamma} \right)^2 \frac{(\gamma - 1)^2}{(\gamma + 1)^2}. \end{aligned} \quad (3.82)$$

To account approximately for the effect of annihilation/creation on the DCS for close inelastic interactions of positrons, we shall use the expression (3.75), with the factor in

parenthesis replaced by the Bhabha factor,

$$F^{(+)}(E, W) = 1 - b_1 \frac{W}{E} + b_2 \left(\frac{W}{E} \right)^2 - b_3 \left(\frac{W}{E} \right)^3 + b_4 \left(\frac{W}{E} \right)^4. \quad (3.83)$$

That is,

$$\frac{d^2 \sigma_{\text{clo}}^{(+)}}{dW dQ} = \frac{2\pi e^4}{m_e v^2} \sum_k f_k \frac{1}{W^2} F^{(+)}(E, W) \delta(W - Q) \Theta(W - W_k). \quad (3.84)$$

Notice that the maximum energy loss in collisions of positrons with energy E is $W_{\text{max}} = E$.

3.2.3 Integrated cross sections

The energy-loss DCS is defined as

$$\frac{d\sigma_{\text{in}}}{dW} \equiv \int_{Q_-}^{Q_+} \frac{d^2 \sigma_{\text{in}}}{dW dQ} dQ = \frac{d\sigma_{\text{dis,l}}}{dW} + \frac{d\sigma_{\text{dis,t}}}{dW} + \frac{d\sigma_{\text{clo}}}{dW}, \quad (3.85)$$

where Q_- and Q_+ are the minimum and maximum kinematically allowed recoil energies given by Eq. (A.31). The contributions from distant longitudinal and transverse interactions are

$$\frac{d\sigma_{\text{dis,l}}}{dW} = \frac{2\pi e^4}{m_e v^2} \sum_k f_k \frac{1}{W_k} \ln \left(\frac{W_k Q_- + 2m_e c^2}{Q_- W_k + 2m_e c^2} \right) \delta(W - W_k) \Theta(W_k - Q_-) \quad (3.86)$$

and

$$\frac{d\sigma_{\text{dis,t}}}{dW} = \frac{2\pi e^4}{m_e v^2} \sum_k f_k \frac{1}{W_k} \left\{ \ln \left(\frac{1}{1 - \beta^2} \right) - \beta^2 - \delta_F \right\} \delta(W - W_k) \Theta(W_k - Q_-), \quad (3.87)$$

respectively. The energy-loss DCS for close collisions is

$$\frac{d\sigma_{\text{clo}}^{(\pm)}}{dW} = \frac{2\pi e^4}{m_e v^2} \sum_k f_k \frac{1}{W^2} F^{(\pm)}(E, W) \Theta(W - W_k). \quad (3.88)$$

Our analytical GOS model provides quite an accurate *average* description of inelastic collisions (see below). However, the continuous energy loss spectrum associated with single distant excitations of a given atomic electron shell is approximated here as a single resonance (a δ -distribution). As a consequence, the simulated energy loss spectra show unphysical narrow peaks at energy losses that are multiples of the resonance energies. These spurious peaks are automatically smoothed out after multiple inelastic collisions and also when the bin width used to tally the energy loss distributions is larger than the difference between resonance energies of neighbouring oscillators.

The PDF of the energy loss in a single inelastic collision is given by

$$p_{\text{in}}(W) = \frac{1}{\sigma_{\text{in}}} \frac{d\sigma_{\text{in}}}{dW}, \quad (3.89)$$

where

$$\sigma_{\text{in}} = \int_0^{W_{\text{max}}} \frac{d\sigma_{\text{in}}}{dW} dW \quad (3.90)$$

is the total cross section for inelastic interactions. It is convenient to introduce the quantities

$$\sigma_{\text{in}}^{(n)} \equiv \int_0^{W_{\text{max}}} W^n \frac{d\sigma_{\text{in}}}{dW} dW = \sigma_{\text{in}} \int_0^{W_{\text{max}}} W^n p_{\text{in}}(W) dW = \sigma_{\text{in}} \langle W^n \rangle, \quad (3.91)$$

where $\langle W^n \rangle$ denotes the n -th moment of the energy loss in a single collision (notice that $\sigma_{\text{in}}^{(0)} = \sigma_{\text{in}}$). $\sigma_{\text{in}}^{(1)}$ and $\sigma_{\text{in}}^{(2)}$ are known as the stopping cross section and the energy straggling cross section (for inelastic collisions), respectively.

The mean free path λ_{in} for inelastic collisions is

$$\lambda_{\text{in}}^{-1} = \mathcal{N} \sigma_{\text{in}}, \quad (3.92)$$

where $\mathcal{N}$ is the number of scattering centres (atoms or molecules) per unit volume. The stopping power S_{in} and the energy straggling parameter Ω_{in}^2 are defined by

$$S_{\text{in}} = \mathcal{N} \sigma_{\text{in}}^{(1)} = \frac{\langle W \rangle}{\lambda_{\text{in}}} \quad (3.93)$$

and

$$\Omega_{\text{in}}^2 = \mathcal{N} \sigma_{\text{in}}^{(2)} = \frac{\langle W^2 \rangle}{\lambda_{\text{in}}}. \quad (3.94)$$

Notice that the stopping power gives the average energy loss per unit path length⁴. The physical meaning of the straggling parameter is less direct. Consider a monoenergetic electron (or positron) beam of energy E that impinges normally on a foil of material of (small) thickness ds , and assume that the electrons do not scatter (*i.e.*, they are not deflected) in the foil. The product $\Omega_{\text{in}}^2 ds$ then gives the variance of the energy distribution of the beam after traversing the foil (see also Section 4.2).

The integrated cross sections $\sigma_{\text{in}}^{(n)}$ can be calculated as

$$\sigma_{\text{in}}^{(n)} = \sigma_{\text{dis,l}}^{(n)} + \sigma_{\text{dis,t}}^{(n)} + \sigma_{\text{clo}}^{(n)}. \quad (3.95)$$

The contributions from distant longitudinal and transverse interactions are

$$\sigma_{\text{dis,l}}^{(n)} = \frac{2\pi e^4}{m_e v^2} \sum_k f_k W_k^{n-1} \ln \left(\frac{W_k Q_- + 2m_e c^2}{Q_- W_k + 2m_e c^2} \right) \Theta(W_{\text{max}} - W_k) \quad (3.96)$$

and

$$\sigma_{\text{dis,t}}^{(n)} = \frac{2\pi e^4}{m_e v^2} \sum_k f_k W_k^{n-1} \left\{ \ln \left(\frac{1}{1 - \beta^2} \right) - \beta^2 - \delta_F \right\} \Theta(W_{\text{max}} - W_k), \quad (3.97)$$

⁴The term “stopping power” is somewhat misleading; in fact, S_{in} has the dimensions of force.

respectively. Notice that for distant interactions $W_{\max} = E$, for both electrons and positrons.

The integrated cross sections for close collisions are

$$\sigma_{\text{clo}}^{(n)} = \frac{2\pi e^4}{m_e v^2} \sum_k f_k \int_{W_k}^{W_{\max}} W^{n-2} F^{(\pm)}(E, W) dW. \quad (3.98)$$

In the case of electrons, the integrals in this formula are of the form

$$J_n^{(-)} = \int W^{n-2} \left[1 + \left(\frac{W}{E-W} \right)^2 - \frac{(1-a)W}{E-W} + \frac{aW^2}{E^2} \right] dW \quad (3.99)$$

and can be calculated analytically. For the orders 0, 1 and 2 we have

$$J_0^{(-)} = -\frac{1}{W} + \frac{1}{E-W} + \frac{1-a}{E} \ln \left(\frac{E-W}{W} \right) + \frac{aW}{E^2}, \quad (3.100)$$

$$J_1^{(-)} = \ln W + \frac{E}{E-W} + (2-a) \ln(E-W) + \frac{aW^2}{2E^2} \quad (3.101)$$

and

$$J_2^{(-)} = (2-a)W + \frac{2E^2 - W^2}{E-W} + (3-a)E \ln(E-W) + \frac{aW^3}{3E^2}. \quad (3.102)$$

For positrons, the integrals in (3.98),

$$J_n^{(+)} \equiv \int W^{n-2} \left[1 - b_1 \frac{W}{E} + b_2 \left(\frac{W}{E} \right)^2 - b_3 \left(\frac{W}{E} \right)^3 + b_4 \left(\frac{W}{E} \right)^4 \right] dW, \quad (3.103)$$

can also be evaluated analytically as

$$J_0^{(+)} = -\frac{1}{W} - b_1 \frac{\ln W}{E} + b_2 \frac{W}{E^2} - b_3 \frac{W^2}{2E^3} + b_4 \frac{W^3}{3E^4}, \quad (3.104)$$

$$J_1^{(+)} = \ln W - b_1 \frac{W}{E} + b_2 \frac{W^2}{2E^2} - b_3 \frac{W^3}{3E^3} + b_4 \frac{W^4}{4E^4} \quad (3.105)$$

and

$$J_2^{(+)} = W - b_1 \frac{W^2}{2E} + b_2 \frac{W^3}{3E^2} - b_3 \frac{W^4}{4E^3} + b_4 \frac{W^5}{5E^4}. \quad (3.106)$$

Figure 3.9 displays total inelastic cross sections for electrons in aluminium and gold, as well as contributions from various groups of shells, as functions of the kinetic energy of the projectile. The curves labelled “K” and “L1+...” represent cross sections for ionisation in these shells. The cross section for ionisation in a bound shell decreases rapidly with the shell ionisation energy U_i (since energy transfers less than U_i , which would promote the target electron to occupied states, are forbidden). As a consequence, collisions occur preferentially with electrons in the conduction band and in outer bound shells. Inner-shell ionisation by electron/positron impact is a relatively unlikely process.

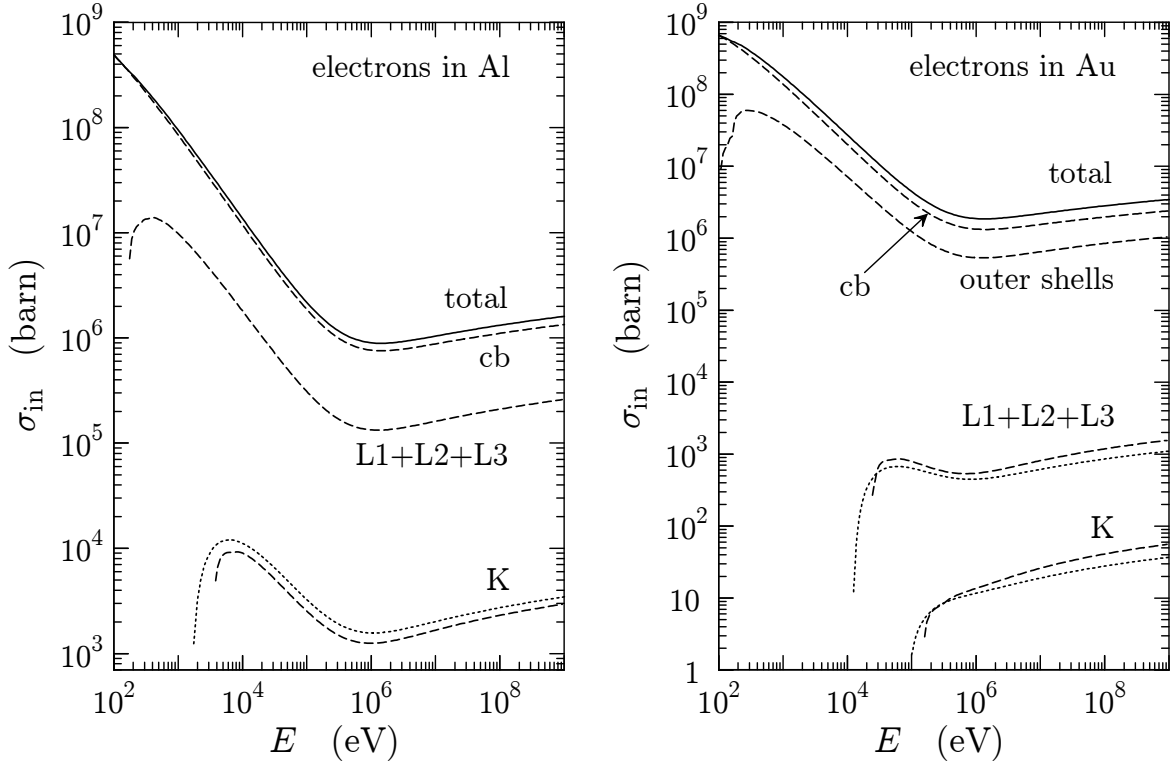


Figure 3.9: Total inelastic cross sections for electrons in aluminium and gold and contributions from the K shell, L shell, conduction band (cb) and outer shells, calculated from our model GOS ignoring density effect corrections (*i.e.*, with $\delta_F = 0$). The dotted curves represent K- and L-shell ionisation cross sections calculated from the optical-data model described in Section 3.2.6, which yields results in close agreement with experimental data. Note: 1 barn= 10^{-24} cm².

It should be noted that our GOS model is too crude to provide an accurate description of inner-shell ionisation. To illustrate this limitation, Fig. 3.9 includes K- and L-shell ionisation cross sections obtained from the optical-data model described in Section 3.2.6, which are known to agree reasonably well with experimental data (Mayol and Salvat, 1990). We see that there are significant differences between the cross sections from the optical-data model and the predictions of our simple GOS model, which is designed to yield accurate stopping powers only. To get a realistic picture of inner-shell ionisation, we have to rely on much more elaborate physics models. In fact, even the Born approximation ceases to be appropriate for projectiles with kinetic energies near the ionisation threshold.

Collision stopping powers for electrons in aluminium, silver and gold obtained from the present analytical model are compared with sample values from the ICRU37 (1984) stopping power tables [given also in Berger and Seltzer (1982)] for $E \geq 10$ keV in Fig. 3.10. Our results practically coincide with the values in the tables of reference, because

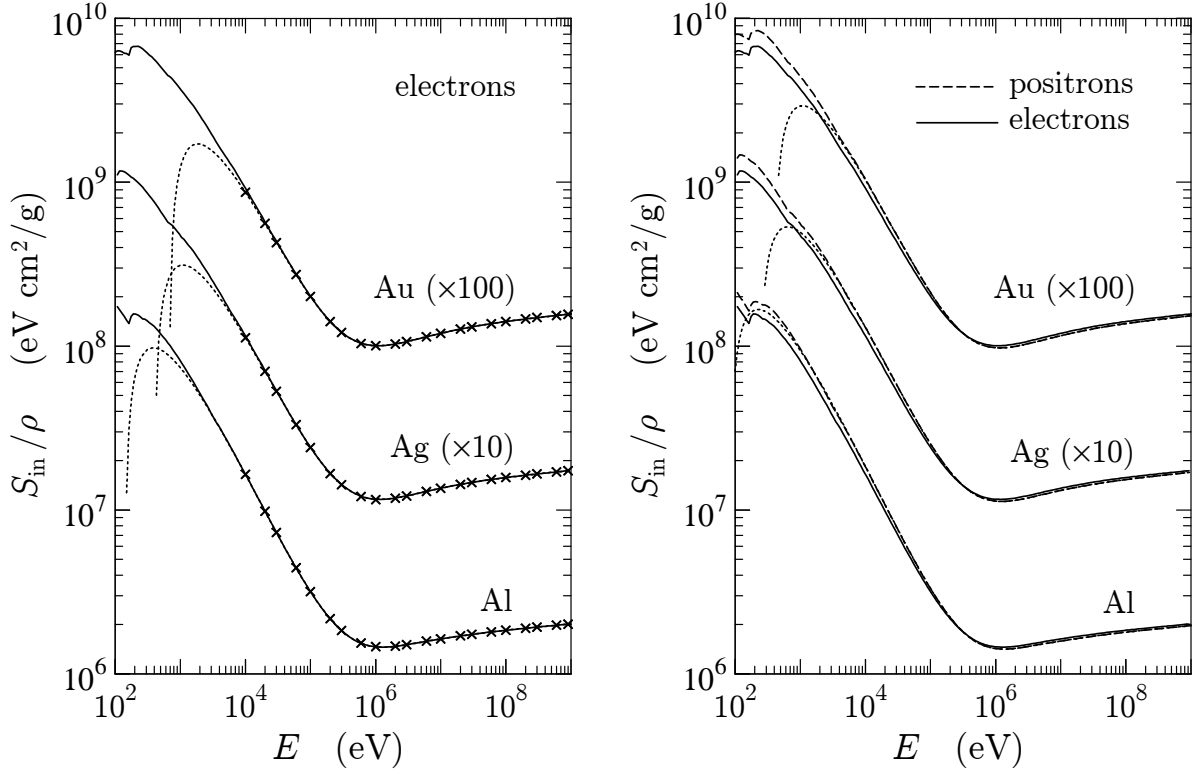


Figure 3.10: Collision stopping power S_{in}/ρ for electrons and positrons in aluminium, silver ($\times 10$) and gold ($\times 100$) as a function of the kinetic energy. Continuous and dashed curves are results from the present model. Crosses are data from the ICRU37 tables (1984) [also, Berger and Seltzer, 1982)]. The dotted curves are predictions from the Bethe formula (3.112), for electrons and positrons.

we are using the same values of the mean excitation energy I . In Fig. 3.11, inelastic mean free paths and stopping powers for low-energy electrons ($E = 100$ eV to 100 keV) in aluminium and gold obtained from the present model are compared with experimental data from several authors. We see that the theory predicts the energy variation of total integrated cross sections down to relatively low energies. It should be noted that the adopted value of W_{cb} , the resonance energy of conduction band electrons, has a strong effect on the calculated mean free paths. In the case of free-electron-like materials such as aluminium, W_{cb} can be identified with the energy of plasmon excitations (which is the dominant energy-loss mechanism). For other solids, the outermost electrons have a broad energy loss spectrum and there is no simple way of predicting this parameter. Fortunately, the stopping power (and, hence, the global stopping process) is practically independent of the adopted value of W_{cb} . To generate the data for aluminium, Fig. 3.11, we have set $W_{\text{cb}} = 15$ eV, which is the measured energy of volume plasmons in the metal [Eq. (3.62) with $f_{\text{cb}} = 3$ conduction electrons per atom gives $W_{\text{cb}} = 15.8$ eV]; in this case, the calculated mean free paths are seen to agree fairly well with measured data. In the case of gold, Eq. (3.62) with $f_{\text{cb}} = 11$ conduction electrons per atom gives $W_{\text{cb}} = 30$ eV.

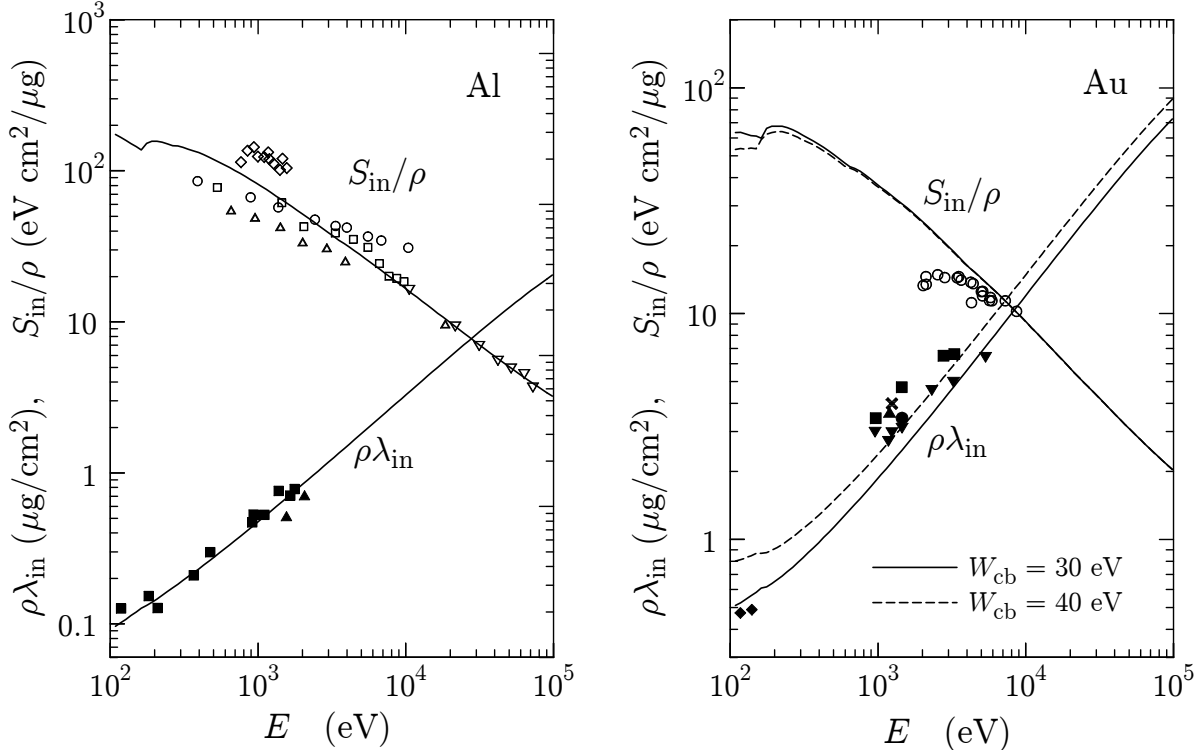


Figure 3.11: Collision mean free path and stopping power for low-energy electrons in aluminium and gold. The plotted quantities are $\rho\lambda_{\text{in}}$ and S_{in}/ρ . Special symbols are experimental data from different sources (see Fernández-Varea *et al.*, 1993a); closed symbols for mean free paths and open symbols for stopping powers.

Figure 3.11 shows stopping powers and mean free paths for electrons in gold obtained with $W_{\text{cb}} = 30$ and 40 eV. We see that, as indicated above, the mean free path varies strongly with this parameter, but the stopping power is practically insensitive to it.

3.2.4 Stopping power of high-energy electrons and positrons

It is of interest to evaluate explicitly the stopping power for projectiles with high energies ($E \gg U_k$). We shall assume that $U_k \ll 2m_e c^2$ (for the most unfavourable case of the K shell of heavy elements, U_k is of the order of $2m_e c^2/10$). Under these circumstances, $Q_- \ll 2m_e c^2$ and we can use the approximation [see Eq. (A.35)]

$$Q_- \simeq W_k^2 / (2m_e c^2 \beta^2). \quad (3.107)$$

The contribution from distant (longitudinal and transverse) interactions to the stopping cross section is then [see Eqs. (3.86) and (3.87)]

$$\sigma_{\text{dis}}^{(1)} \simeq \frac{2\pi e^4}{m_e v^2} \sum_k f_k \left\{ \ln \left(\frac{2m_e c^2}{W_k} \right) + \ln \left(\frac{1}{1 - \beta^2} \right) - \beta^2 - \delta_F \right\}. \quad (3.108)$$

The contribution of close interactions is given by

$$\sigma_{\text{clo}}^{(1)} = \frac{2\pi e^4}{m_e v^2} \sum_k f_k \int_{W_k}^{W_{\text{max}}} W^{-1} F^{(\pm)}(E, W) dW. \quad (3.109)$$

Recalling that $E \gg U_k$, we have

$$\begin{aligned} \sigma_{\text{clo}}^{(1)} \simeq \frac{2\pi e^4}{m_e v^2} \sum_k f_k \left\{ \ln \left(\frac{E}{W_k} \right) + 1 - \left[1 + \beta^2 + 2\sqrt{1 - \beta^2} \right] \ln 2 \right. \\ \left. + \frac{1}{8} \left(1 - \sqrt{1 - \beta^2} \right)^2 \right\} \end{aligned} \quad (3.110)$$

for electrons and

$$\sigma_{\text{clo}}^{(1)} \simeq \frac{2\pi e^4}{m_e v^2} \sum_k f_k \left\{ \ln \left(\frac{E}{W_k} \right) - b_1 + \frac{b_2}{2} - \frac{b_3}{3} + \frac{b_4}{4} \right\} \quad (3.111)$$

for positrons. Adding the distant and close stopping cross sections, and using the relation (3.61), we arrive at the familiar Bethe formula for the stopping power,

$$S_{\text{in}} \equiv \mathcal{N} \left(\sigma_{\text{dis}}^{(1)} + \sigma_{\text{clo}}^{(1)} \right) = \mathcal{N} \frac{2\pi e^4}{m_e v^2} Z \left\{ \ln \left(\frac{E^2}{I^2} \frac{\gamma + 1}{2} \right) + f^{(\pm)}(\gamma) - \delta_F \right\}, \quad (3.112)$$

where

$$f^{(-)}(\gamma) = 1 - \beta^2 - \frac{2\gamma - 1}{\gamma^2} \ln 2 + \frac{1}{8} \left(\frac{\gamma - 1}{\gamma} \right)^2 \quad (3.113)$$

and

$$f^{(+)}(\gamma) = 2 \ln 2 - \frac{\beta^2}{12} \left[23 + \frac{14}{\gamma + 1} + \frac{10}{(\gamma + 1)^2} + \frac{4}{(\gamma + 1)^3} \right] \quad (3.114)$$

for electrons and positrons, respectively. This formula can be derived from very general arguments that do not require knowing the fine details of the GOS; the only information needed is contained in the Bethe sum rule (3.54) and in the definition (3.55) of the mean excitation energy (see, *e.g.*, Fano, 1963). Since our approximate analytical GOS model is physically motivated, it satisfies the sum rule and reproduces the adopted value of the mean ionisation energy, it yields (at high energies) the exact Bethe formula.

It is striking that the “asymptotic” Bethe formula is in fact valid down to fairly small energies, of the order of 10 keV for high- Z materials (see Fig. 3.10). It also accounts for the differences between the stopping powers of electrons and positrons (to the same degree as our GOS model approximation).

For ultrarelativistic projectiles, for which the approximation (3.73) holds, the Bethe formula simplifies to

$$S_{\text{in}} \simeq \mathcal{N} \frac{2\pi e^4}{m_e v^2} Z \left\{ \ln \left(\frac{E^2}{\Omega_p^2} \frac{\gamma + 1}{2\gamma^2} \right) + f^{(\pm)}(\gamma) + 1 \right\}. \quad (3.115)$$

The mean excitation energy I has disappeared from this formula, showing that at very high energies the stopping power depends only on the electron density $\mathcal{N}Z$ of the medium.

3.2.5 Simulation of hard inelastic collisions

The DCSs given by expressions (3.85)–(3.88) permit the random sampling of the energy loss W and the angular deflection θ by using purely analytical methods. In the following we consider the case of mixed (class II) simulation, in which only hard collisions, with energy loss larger than a specified cutoff value W_{cc} , are simulated (see Chapter 4). As the value of the cutoff energy loss can be selected arbitrarily, the sampling algorithm can also be used in detailed (interaction-by-interaction) simulations ($W_{cc} = 0$).

The first stage of the simulation is the selection of the active oscillator, for which we need to know the restricted total cross section,

$$\begin{aligned}\sigma(W_{cc}) &\equiv \int_{W_{cc}}^{W_{\max}} \frac{d\sigma_{\text{in}}}{dW} dW = \sigma_{\text{dis,l}}(W_{cc}) + \sigma_{\text{dis,t}}(W_{cc}) + \sigma_{\text{clo}}(W_{cc}) \\ &= \sum_k \sigma_k(W_{cc}),\end{aligned}\tag{3.116}$$

as well as the contribution of each oscillator, $\sigma_k(W_{cc})$. The active oscillator is sampled from the point probabilities $p_k = \sigma_k(W_{cc})/\sigma(W_{cc})$. In the present version of PENELOPE, these probabilities are calculated at initialisation time and stored in memory. Thus, with a moderate increase in memory storage, the simulation speed is made fairly independent of the number of oscillators [in previous versions of the code, the total cross sections $\sigma_k(W_{cc})$ of all the oscillators were calculated in each interaction]. In mixed simulations, the sampling algorithm can be sped up by using a larger cutoff energy loss W_{cc} , which eliminates all the oscillators with $W_k < W_{cc}$ from the sum.

After selecting the active oscillator, the oscillator branch (distant or close) is determined and, finally, the variables W and Q (or $\cos\theta$) are sampled from the associated DCS. For close collisions, $Q = W$ and, therefore, the scattering angle is obtained directly from the energy loss.

3.2.5.1 Hard distant interactions

In distant interactions with the k -th oscillator, $W = W_k$. The contributions of transverse and longitudinal interactions to the restricted cross section define the relative probabilities of these interaction modes. If the interaction is (distant) transverse, the angular deflection of the projectile is neglected, *i.e.*, $\cos\theta = 1$. For distant longitudinal collisions, the (unnormalised) PDF of Q is given by [see Eq. (3.67)]

$$P_{dk}(Q) = \begin{cases} \frac{1}{Q [1 + Q/(2m_e c^2)]} & \text{if } Q_- < Q < W_k, \\ 0 & \text{otherwise,} \end{cases}\tag{3.117}$$

where Q_- is the minimum recoil energy, Eq. (A.31). Random sampling from this PDF can be performed by the inverse-transform method, which gives the sampling formula

$$Q = Q_S \left\{ \left[\frac{Q_S}{W_k} \left(1 + \frac{W_k}{2m_e c^2} \right) \right]^\xi - \frac{Q_S}{2m_e c^2} \right\}^{-1}, \quad (3.118)$$

where

$$Q_S \equiv \frac{Q_-}{1 + Q_-/(2m_e c^2)}. \quad (3.119)$$

Once the energy loss and the recoil energy have been sampled, the polar scattering angle θ is determined from Eq. (A.40),

$$\cos \theta = \frac{E(E + 2m_e c^2) + (E - W)(E - W + 2m_e c^2) - Q(Q + 2m_e c^2)}{2\sqrt{E(E + 2m_e c^2)(E - W)(E - W + 2m_e c^2)}}. \quad (3.120)$$

The azimuthal scattering angle ϕ is sampled uniformly in the interval $(0, 2\pi)$.

3.2.5.2 Hard close collisions of electrons

For the formulation of the sampling algorithm, it is convenient to introduce the reduced energy loss $\kappa \equiv W/E$. The PDF of κ in close collisions of electrons with the k -th oscillator is given by [see Eqs. (3.78) and (3.79)]

$$P_k^{(-)}(\kappa) \equiv \kappa^{-2} F^{(-)}(E, W) \Theta(\kappa - \kappa_c) \Theta\left(\frac{1}{2} - \kappa\right) = \left[\frac{1}{\kappa^2} + \frac{1}{(1 - \kappa)^2} - \frac{1}{\kappa(1 - \kappa)} + a \left(1 + \frac{1}{\kappa(1 - \kappa)} \right) \right] \Theta(\kappa - \kappa_c) \Theta\left(\frac{1}{2} - \kappa\right), \quad (3.121)$$

with $\kappa_c \equiv \max(W_k, W_{cc})/E$. Notice that the maximum allowed value of κ is $1/2$. Here, normalisation is irrelevant.

We introduce the distribution

$$\Phi^{(-)}(\kappa) \equiv (\kappa^{-2} + 5a) \Theta(\kappa - \kappa_c) \Theta\left(\frac{1}{2} - \kappa\right), \quad a \equiv \left(\frac{\gamma - 1}{\gamma} \right)^2. \quad (3.122)$$

It may be shown that $\Phi^{(-)} > P_k^{(-)}$ in the interval $(\kappa_c, \frac{1}{2})$. Therefore, we can sample the reduced energy loss κ from the PDF (3.121) by using the rejection method (see Section 1.2.5) with trial values sampled from the distribution (3.122) and acceptance probability $P_k^{(-)}/\Phi^{(-)}$.

Random sampling from the PDF (3.122), can be performed by using the composition method (Section 1.2.6). We consider the following decomposition of the (normalised) PDF given by Eq. (3.122):

$$\Phi_{\text{norm}}^{(-)}(\kappa) = \frac{1}{1 + 5a\kappa_c/2} [p_1(\kappa) + (5a\kappa_c/2)p_2(\kappa)], \quad (3.123)$$

where

$$p_1(\kappa) = \frac{\kappa_c}{1 - 2\kappa_c} \kappa^{-2}, \quad p_2(\kappa) = \frac{2}{1 - 2\kappa_c} \quad (3.124)$$

are normalised PDFs in the interval $(\kappa_c, \frac{1}{2})$. Random values of κ from the PDF (3.122) can be generated by using the following algorithm:

- (i) Generate ξ .
- (ii) Set $\zeta = (1 + 5a\kappa_c/2)\xi$.
- (iii) If $\zeta < 1$, deliver the value $\kappa = \kappa_c/[1 - \zeta(1 - 2\kappa_c)]$.
- (iv) If $\zeta > 1$, deliver the value $\kappa = \kappa_c + (\zeta - 1)(1 - 2\kappa_c)/(5a\kappa_c)$.

The rejection algorithm for random sampling of κ from the PDF (3.121) proceeds as follows:

- (i) Sample κ from the distribution given by Eq. (3.122).
- (ii) Generate a random number ξ .
- (iii) If $\xi(1 + 5a\kappa^2) < \kappa^2 P_k^{(-)}(\kappa)$, deliver κ .
- (iv) Go to step (i).

Notice that in the third step we accept the κ value with probability $P_k^{(-)}/\Phi^{(-)}$, which approaches unity when κ is small.

The efficiency of this sampling method depends on the values of the energy E and the cutoff reduced energy loss κ_c , as shown in Table 3.1. For a given energy and for W_{cc} values which are not too large, the efficiency increases when W_{cc} decreases.

Table 3.1: Efficiency (%) of the random sampling algorithm of the energy loss in close collisions of electrons and positrons for different values of the energy E and the cutoff energy loss κ_c .

E (eV)	κ_c				
	0.001	0.01	0.1	0.25	0.4
10^3	99.9	99.9	99.8	99.7	99.6
10^5	99.7	98	87	77	70
10^7	99	93	70	59	59
10^9	99	93	71	62	63

After sampling the energy loss $W = \kappa E$, the polar scattering angle θ is obtained from Eq. (A.40) with $Q = W$. This yields

$$\cos^2 \theta = \frac{E - W}{E} \frac{E + 2m_e c^2}{E - W + 2m_e c^2}, \quad (3.125)$$

which agrees with Eq. (A.17). The azimuthal scattering angle ϕ is sampled uniformly in the interval $(0, 2\pi)$.

3.2.5.3 Hard close collisions of positrons

The PDF of the reduced energy loss $\kappa \equiv W/E$ in positron close collisions with the k -th oscillator is given by [see Eqs. (3.83) and (3.84)]

$$\begin{aligned} P_k^{(+)}(\kappa) &= \kappa^{-2} F_k^{(+)}(E, W) \Theta(\kappa - \kappa_c) \Theta(1 - \kappa) \\ &= \left[\frac{1}{\kappa^2} - \frac{b_1}{\kappa} + b_2 - b_3 \kappa + b_4 \kappa^2 \right] \Theta(\kappa - \kappa_c) \Theta(1 - \kappa) \end{aligned} \quad (3.126)$$

with $\kappa_c \equiv \max(W_k, W_{cc})/E$. The maximum allowed reduced energy loss is 1. Again, normalisation is not important.

Consider the distribution

$$\Phi^{(+)}(\kappa) \equiv \kappa^{-2} \Theta(\kappa - \kappa_c) \Theta(1 - \kappa). \quad (3.127)$$

It is easy to see that $\Phi^{(+)} > P_k^{(+)}$ in the interval $(\kappa_c, 1)$. Therefore, we can generate κ from the PDF, Eq. (3.126), by using the rejection method with trial values sampled from the distribution of Eq. (3.127) and acceptance probability $P_k^{(+)}/\Phi^{(+)}$. Sampling from the PDF $\Phi^{(+)}$ can easily be performed with the inverse-transform method.

The algorithm for random sampling from the PDF (3.126), is:

- (i) Sample κ from the PDF (3.127), as $\kappa = \kappa_c/[1 - \xi(1 - \kappa_c)]$.
- (ii) Generate a new random number ξ .
- (iii) If $\xi < \kappa^2 P_k^{(+)}(\kappa)$, deliver κ .
- (iv) Go to step (i).

The efficiency of this algorithm, for given values of the kinetic energy and the cutoff reduced energy loss κ_c , practically coincides with that of the algorithm for electron collisions described above (see Table 3.1).

3.2.5.4 Secondary electron emission

According to our GOS model, each oscillator W_k corresponds to a shell with f_k electrons and ionisation energy U_k . After a hard collision with an inner-shell electron, the primary electron/positron has kinetic energy $E - W$, the “secondary” electron (delta ray) is ejected with kinetic energy $E_s = W - U_i$, and the residual ion is left in an excited state, with a vacancy in shell i , which corresponds to an excitation energy equal to U_i . This energy is eventually released by emission of energetic x rays and Auger electrons. However, in PENELOPE the relaxation of ions produced in hard collisions is not followed. The production of vacancies in inner shells and their relaxation is simulated by an independent, more accurate, scheme (see Section 3.2.6) that is free from the crude approximations involved in our GOS model. To avoid double counting, the excitation energy U_i of the residual ion is deposited locally. On the other hand, when the impact ionisation occurs in an outer shell or in the conduction band, the initial energy of the secondary electron is set equal to W and no fluorescent radiation from the ionised atom is followed by the simulation program. This is equivalent to assuming that the secondary electron carries away the excitation energy of the target atom.

To set the initial direction of the delta ray, we assume that the target electron was initially at rest, *i.e.*, the delta ray is emitted in the direction of the momentum transfer $\mathbf{q}$. This implies that the polar emission angle θ_s (see Fig. 3.1) coincides with the recoil angle θ_r [which is given by Eq. (A.42)],

$$\cos^2 \theta_s = \frac{W^2/\beta^2}{Q(Q + 2m_e c^2)} \left(1 + \frac{Q(Q + 2m_e c^2) - W^2}{2W(E + m_e c^2)} \right)^2. \quad (3.128)$$

In the case of close collisions ($Q = W$), this expression simplifies to

$$\cos \theta_s (Q = W) = \left(\frac{W}{E} \frac{E + 2m_e c^2}{W + 2m_e c^2} \right)^{1/2}, \quad (3.129)$$

which agrees with the result for binary collisions with free electrons at rest, see Eq. (A.18). Since the momentum transfer lies on the scattering plane (*i.e.*, on the plane formed by the initial and final momenta of the projectile), the azimuthal emission angle is $\phi_s = \pi + \phi$.

In reality, the target electrons are not at rest and, therefore, the angular distribution of emitted delta rays is broad. Since the average momentum of bound electrons is zero, the average direction of delta rays coincides with the direction of $\mathbf{q}$. Thus, our simple emission model correctly predicts the average initial direction of delta rays, but disregards the “Doppler broadening” of the angular distribution. This is not a serious drawback, because secondary electrons are usually emitted with initial kinetic energies that are much smaller than the initial energy of the projectile. This means that the direction of motion of the delta ray is randomised, by elastic and inelastic collisions, after a relatively short path length (much shorter than the transport mean free path of the projectile).

3.2.6 Ionisation of inner shells

As indicated above, the theory presented in Sections 3.2.1 and 3.2.2 does not give realistic values of the cross sections for ionisation of inner shells. Hence, it is not appropriate to simulate inner-shell ionisation by electron and positron impact and the subsequent emission of fluorescent radiation, *i.e.*, Auger electrons and characteristic x rays. Nevertheless, the GOS model does provide an appropriate description of the average (stopping and scattering) effect of inelastic collisions on the projectile.

A consistent model for the simulation of inner-shell ionisation and relaxation must account for the following features of the process: 1) space distribution of inner-shell ionisations along the projectile's track, 2) relative probabilities of ionising various atomic electron shells and 3) energies and emission probabilities of the electrons and x rays released through the de-excitation cascade of the ionised atom. The correlation between energy loss/scattering of the projectile and ionisation events is of minor importance and may be neglected (it is observable only in single-scattering experiments where the inelastically scattered electrons and the emitted x rays or Auger electrons are observed in coincidence). Consequently, we shall consider inner-shell ionisation as an independent interaction process that has no effect on the state of the projectile. Accordingly, in the simulation of inelastic collisions the projectile is assumed to cause only the ejection of knock-on electrons (delta rays); in these collisions the target atom is considered to remain unaltered to avoid double counting of ionisations. Thus, to determine the location of ionising events and the atomic shell that is ionised we only need to consider cross sections for ionisation of individual inner shells, which can be obtained from elaborate theoretical models. The relaxation of the vacancies produced by inner-shell ionisations is simulated as described in Section 2.6. This kind of simulation scheme is trivial to implement, but it may cause artifacts (in the form of small negative doses) in space regions where the simulated dose distributions have large relative statistical uncertainties. The reason is that simulated Auger electrons and x rays remove energy from their site (volume bin) of birth, in quantities that may exceed the actual energy deposited by the projectile.

To simulate the ionisation of K, L and M shells [with ionisation energies larger than $\min(\text{EABS}(1), \text{EBAS}(2))$] by electron and positron impact, PENELOPE uses total ionisation cross sections obtained from an optical-data model of the GOS (Mayol and Salvat, 1990). This model assumes the following relationship between the optical oscillator strength (OOS) of the i -th inner shell, $df_i(W)/dW$, and the photoelectric cross section $\sigma_{\text{ph},i}(Z, W)$ for absorption of a photon with energy W ,

$$\frac{df_i(W)}{dW} = \frac{m_e c}{2\pi^2 e^2 \hbar} \sigma_{\text{ph},i}(Z, W). \quad (3.130)$$

This equality holds when the dipole approximation is applicable, *i.e.*, when the wavelength of the photon is much larger than the “size” of the active electron shell. In the calculations we use the photoelectric cross sections from the PENELOPE database, which were extracted from the EPDL (Cullen *et al.*, 1997). The GOS is modelled as a

continuous superposition of δ -oscillators weighted by the OOS,

$$\frac{df_i(Q, W)}{dW} = \int_{U_i}^{\infty} \frac{df_i(W')}{dW'} F(W'; Q, W) dW' + Z_r \delta(W - Q) \Theta(W - U_i) \quad (3.131)$$

with

$$Z_r = Z_i - \int_{U_i}^{\infty} \frac{df_i(W')}{dW'} dW'. \quad (3.132)$$

This GOS model satisfies the Bethe sum rule

$$\int_{U_i}^{\infty} \frac{df_i(Q, W)}{dW} dW = Z_i \quad \text{if } Q > U_i. \quad (3.133)$$

For $Q < U_i$ the integral of the GOS over W is less than the number Z_i of electrons in the active shell (because there is a transfer of oscillator strength to outer shells). As shown by Mayol and Salvat (1990), this model is formally equivalent to the Weizsäcker-Williams method of virtual quanta (see, *e.g.*, Jackson, 1975).

To compute the inner-shell ionisation (“si”) cross section for electrons or positrons of energy E we first consider the energy-loss DCS, which is given by

$$\begin{aligned} \frac{d\sigma_{\text{si},i}^{(\pm)}(E)}{dW} = & \frac{2\pi e^4}{m_e v^2} \left\{ \frac{df_i(W)}{dW} \frac{1}{W} \left[\ln \left(\frac{W}{Q_-} \frac{Q_- + 2m_e c^2}{W + 2m_e c^2} \right) + \ln \left(\frac{1}{1 - \beta^2} \right) - \beta^2 \right] \right. \\ & \left. + \left(Z_r + \int_{U_i}^W \frac{df_i(W')}{dW'} dW' \right) \frac{1}{W^2} F^{(\pm)}(E, W) \right\} \Theta(W - U_i) \Theta(W_{\text{max}} - U_i), \end{aligned} \quad (3.134)$$

where Q_- is the minimum recoil energy for the energy loss W [see Eq. (A.31)] and $F^{(\pm)}(E, W)$ are the Møller (−) and Bhabha (+) factors, Eqs. (3.79) and (3.83).

It is well known that the Born approximation overestimates the ionisation cross sections for incident electrons with kinetic energies near the ionisation threshold. This is mainly due to the distortion of the projectile wave function by the electrostatic field of the target atom. This field produces an increase in the effective kinetic energy of the projectile, which is expected to be important in close collisions. To account for this effect we assume that the incident electron gains a kinetic energy⁵ $2U_i$ before it interacts with a target electron, which is bound with binding energy U_i . The maximum energy loss is assumed to be $W_{\text{max}} = (E + U_i)/2$, because the final energies of the projectile ($E - W$) and the knock-on secondary electron ($W - U_i$) are equal when W equals that value. With this “Coulomb” correction, the ionisation cross section for electrons is

$$\sigma_{\text{si},i}^{(-)}(E) = \int_{U_i}^{(E+U_i)/2} \frac{d\sigma_{\text{si},i}^{(-)}(E + 2U_i)}{dW} dW. \quad (3.135a)$$

⁵For one-electron atoms and ions, the virial theorem implies that the average potential energy of the bound electron is equal to $2U_i$. In close collisions, for which the projectile reaches the position of the target electron, the gain in kinetic energy should be of the order of $2U_i$.

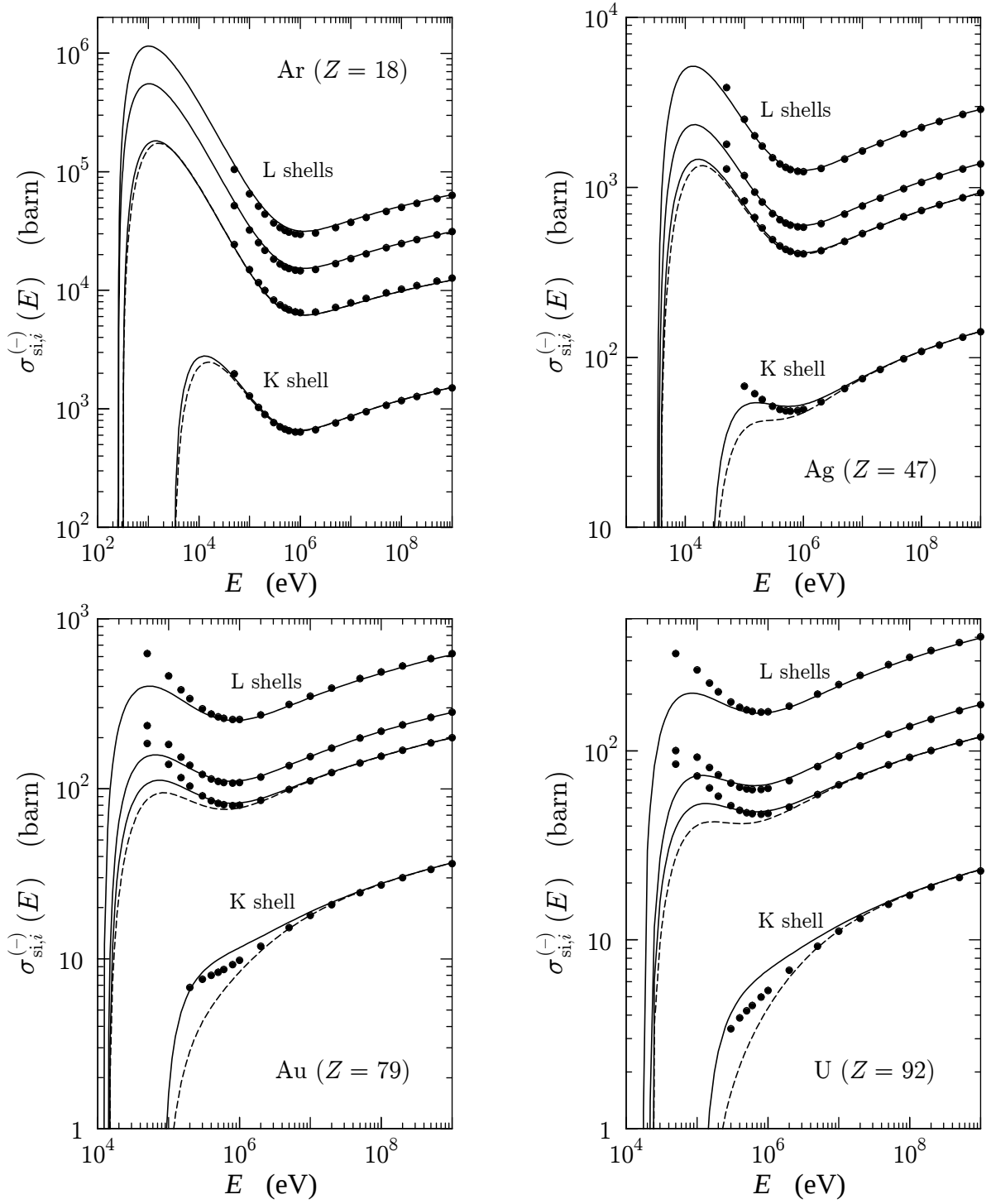


Figure 3.12: Cross sections for ionisation of the K shell and the L subshells of argon, silver, gold and uranium atoms by electron impact as functions of the kinetic energy of the projectiles. Solid curves are results from the present optical data model. Circles represent cross-section values calculated by Scofield (1978) using the relativistic plane-wave Born approximation. The dashed curves are cross sections for ionisation by impact of positrons calculated from Eq. (3.135b).

The Coulomb correction reduces the ionisation cross section near the threshold and yields values in better agreement with experimental data. For positrons the effect of the Coulomb distortion is introduced empirically by simply multiplying the ionisation cross section by a global factor $(1 + U_i/E)^{-3}$. That is,

$$\sigma_{\text{si},i}^{(+)}(E) = \left(\frac{E}{E + U_i} \right)^3 \int_{U_i}^E \frac{d\sigma_{\text{si},i}^{(+)}(E)}{dW} dW. \quad (3.135b)$$

This correction gives positron ionisation cross sections that are smaller than those of electrons near the ionisation threshold, in qualitative agreement with available experimental data (see, *e.g.*, Hippler, 1990; Schneider *et al.*, 1993). Fig. 3.12 displays cross sections for ionisation of K shells and L subshells of atoms of the elements argon, silver, gold and uranium by impact of electrons obtained from the present optical-data model, Eq. (3.126a), together with results from the relativistic plane-wave first Born approximation (Scofield, 1978). The differences at relatively low energies are mostly due to exchange and Coulomb corrections, which were not included in Scofield's calculations. The dashed curves in Fig. 3.12 represent cross sections for ionisation by positron impact, Eq. (3.126b). The relative differences between the cross sections for electrons and positrons are seen to increase with the binding energy of the active shell. It is worth mentioning that the present optical-data model disregards the influence of the polarisability of the medium (density effect) on inner-shell ionisation. This effect causes a reduction of the ionisation cross sections for projectiles with very high energies, which decreases with the binding energy of the knock-on electron.

The molecular cross section for ionisation of inner shells is evaluated as (additivity approximation)

$$\sigma_{\text{si},\text{mol}}^{(\pm)}(E) = \sum_i \sigma_{\text{si},i}^{(\pm)}(E), \quad (3.136)$$

where the summation extends over all inner shells of the atoms in the molecule.

3.3 Bremsstrahlung emission

As a result of the acceleration caused by the electrostatic field of atoms, swift electrons (or positrons) emit bremsstrahlung (braking radiation). In each bremsstrahlung event, an electron with kinetic energy E generates a photon of energy W , which takes values in the interval from 0 to E . The process is described by an atomic DCS, differential in the energy loss W , the final direction of the projectile and the direction of the emitted photon (Koch and Motz, 1959; Tsai, 1974). The habitual practice in Monte Carlo simulation is to sample the energy loss from the single-variable distribution obtained by integrating the DCS over the other variables. This permits the generation of W easily, but information on the angular distributions is completely lost and has to be regained from suitable approximations. Angular deflections of the projectile are considered to be accounted for by the elastic scattering DCS and, consequently, the direction of movement of the projectile is kept unaltered in the simulation of radiative events.

3.3.1 The energy-loss scaled DCS

A simple description of the bremsstrahlung DCS is provided by the Bethe-Heitler formula with screening, which is derived within the Born approximation (Bethe and Heitler, 1934; Tsai, 1974). Although this formula is valid only when the kinetic energy of the electron before and after photon emission is much larger than its rest energy $m_e c^2$, it accounts for the most relevant features of the emission process. Within the Born approximation, bremsstrahlung emission is closely related to electron-positron pair production. In particular, the Bethe-Heitler DCS formulae for pair production and bremsstrahlung emission involve the same screening functions. Considering the exponential screening model (2.74), the Bethe-Heitler DCS for bremsstrahlung emission by electrons in the field of an atom of atomic number Z and screening radius R can be expressed as (Salvat and Fernández-Varea, 1992)

$$\frac{d\sigma_{\text{br}}^{(\text{BH})}}{dW} = r_e^2 \alpha Z(Z + \eta) \frac{1}{W} \left[\epsilon^2 \varphi_1(b) + \frac{4}{3}(1 - \epsilon) \varphi_2(b) \right], \quad (3.137)$$

where α is the fine-structure constant, r_e is the classical electron radius,

$$\epsilon = \frac{W}{E + m_e c^2} = \frac{W}{\gamma m_e c^2}, \quad b = \frac{R m_e c}{\hbar} \frac{1}{2\gamma} \frac{\epsilon}{1 - \epsilon}, \quad (3.138)$$

and

$$\begin{aligned} \varphi_1(b) &= 4 \ln(R m_e c / \hbar) + 2 - 2 \ln(1 + b^2) - 4b \arctan(b^{-1}), \\ \varphi_2(b) &= 4 \ln(R m_e c / \hbar) + \frac{7}{3} - 2 \ln(1 + b^2) - 6b \arctan(b^{-1}) \\ &\quad - b^2 [4 - 4b \arctan(b^{-1}) - 3 \ln(1 + b^{-2})]. \end{aligned} \quad (3.139)$$

The quantity η in Eq. (3.137) accounts for the production of bremsstrahlung in the field of the atomic electrons (see, *e.g.*, Seltzer and Berger, 1985); in the high-energy limit $\eta \simeq 1.2$.

The Bethe-Heitler formula indicates that, for a given value of Z , the quantity $W d\sigma_{\text{br}}/dW$ varies smoothly with E and W . It is therefore customary to express the DCS for bremsstrahlung emission by electrons in the form

$$\frac{d\sigma_{\text{br}}}{dW} = \frac{Z^2}{\beta^2} \frac{1}{W} \chi(Z, E, \kappa), \quad (3.140)$$

where W is the energy of the emitted photon, κ is the reduced photon energy, defined as

$$\kappa \equiv W/E, \quad (3.141)$$

which takes values between 0 and 1. The quantity

$$\chi(Z, E, \kappa) = (\beta^2/Z^2) W \frac{d\sigma_{\text{br}}}{dW} \quad (3.142)$$

is known as the “scaled” bremsstrahlung DCS; for a given element Z , it varies smoothly with E and κ . Seltzer and Berger (1985, 1986) produced extensive tables of the scaled DCS for all the elements ($Z = 1-92$) and for electron energies from 1 keV to 10 GeV. They tabulated the scaled DCSs for emission in the (screened) field of the nucleus (electron-nucleus bremsstrahlung) and in the field of atomic electrons (electron-electron bremsstrahlung) separately, as well as their sum, the total scaled DCS. The electron-nucleus bremsstrahlung DCS was calculated by combining analytical high-energy theories with results from partial-wave calculations by Pratt *et al.* (1977) for bremsstrahlung emission in screened atomic fields and energies below 2 MeV. The scaled DCS for electron-electron bremsstrahlung was obtained from the theory of Haug (1975) combined with a screening correction that involves Hartree-Fock incoherent scattering functions. Seltzer and Berger’s scaled DCS tables constitute the most reliable theoretical representation of bremsstrahlung energy spectra available at present.

The PENELOPE database of scaled bremsstrahlung DCSs consists of 99 files, one for each element from hydrogen to einsteinium, which were generated from the original database of Seltzer and Berger. The file of the element Z contains the values of $\chi(Z, E_i, \kappa_j)$ for a set of electron kinetic energies E_i , which covers the range from 1 keV to 10 GeV and is suitably spaced to allow accurate natural cubic spline interpolation in $\ln E$. For each energy E_i in this grid, the table contains the values of the scaled DCS for a given set of 32 reduced photon energies κ_j (the same for all elements), which span the interval (0,1), with a higher density at the upper end of this interval to reproduce the structure of the bremsstrahlung “tip” (see Fig. 3.13). The spacing of the κ -grid is dense enough to allow linear interpolation of $\chi(Z, E_i, \kappa_j)$ in κ .

In the case of compounds (or mixtures) we use the additivity rule and compute the molecular DCS as the sum of the DCSs of all the atoms in a molecule. Consider a compound $X_x Y_y$, whose molecules consist of x atoms of the element X and y atoms of the element Y . The molecular DCS is

$$\frac{d\sigma_{\text{br,mol}}}{dW} = x \frac{Z_X^2}{\beta^2} \frac{1}{W} \chi(Z_X, E, \kappa) + y \frac{Z_Y^2}{\beta^2} \frac{1}{W} \chi(Z_Y, E, \kappa). \quad (3.143)$$

To simulate each radiative event in a compound, we should first select the element (X or Y) where the emission occurs and then sample the photon energy and direction from the corresponding atomic DCS. This is a lengthy process and requires storing the scaled DCSs for all the elements present. To simplify the simulation, we shall express the molecular DCS in the same form as the atomic DCS, Eq. (3.140),

$$\frac{d\sigma_{\text{br,mol}}}{dW} = \frac{Z_{\text{eq}}^2}{\beta^2} \frac{1}{W} \chi_{\text{mol}}(Z_{\text{eq}}, E, \kappa), \quad (3.144)$$

where

$$Z_{\text{eq}}^2 \equiv \frac{1}{x+y} (xZ_X^2 + yZ_Y^2) \quad (3.145)$$

is the “equivalent” atomic number Z_{eq} and

$$\chi_{\text{mol}}(Z_{\text{eq}}, E, \kappa) = \frac{xZ_X^2}{Z_{\text{eq}}^2} \chi(Z_X, E, \kappa) + \frac{yZ_Y^2}{Z_{\text{eq}}^2} \chi(Z_Y, E, \kappa) \quad (3.146)$$

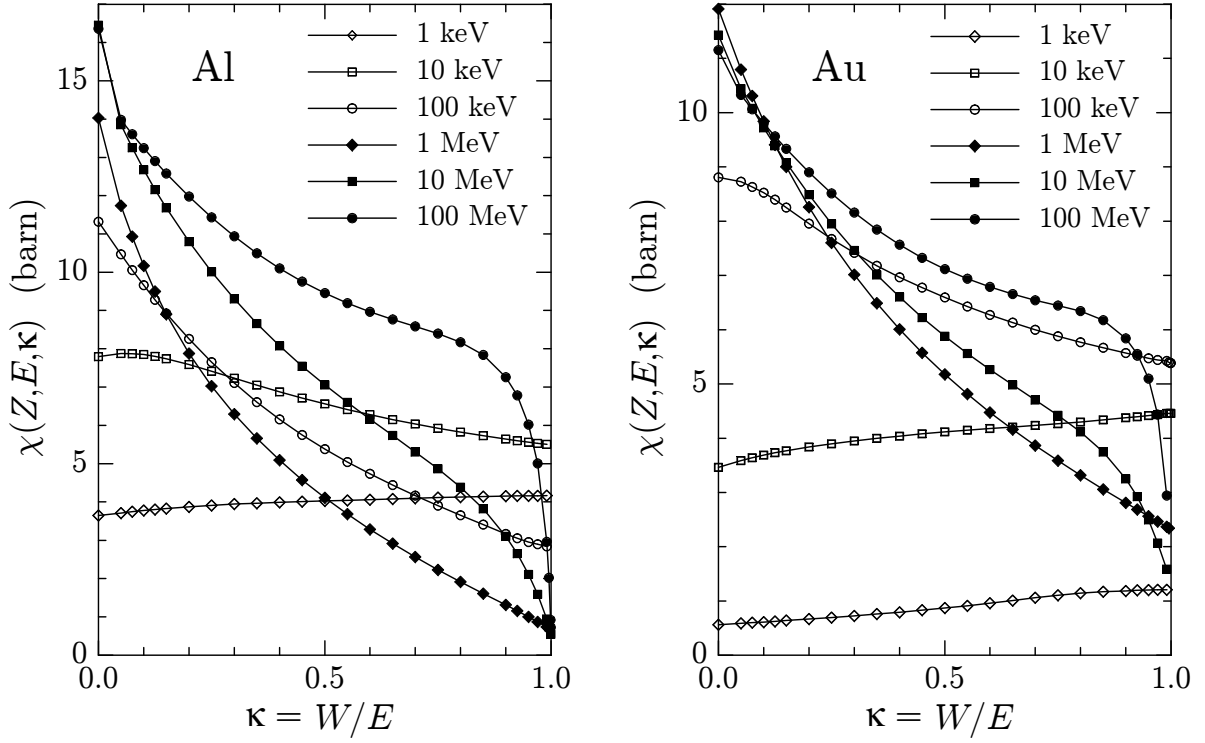


Figure 3.13: Numerical scaled bremsstrahlung energy-loss DCSs of aluminium and gold for electrons with the indicated energies (Seltzer and Berger, 1986).

is the molecular scaled DCS. Radiative events will be sampled directly from the molecular DCS (3.144). This method may introduce slight inconsistencies in the angular distribution of the emitted photons (see below), which usually have a negligible effect on the simulation results.

The radiative DCS for positrons reduces to that of electrons in the high-energy limit but is smaller for intermediate and low energies. Owing to the lack of more accurate calculations, the DCS for positrons is obtained by multiplying the electron DCS by a κ -independent factor, *i.e.*,

$$\frac{d\sigma_{\text{br}}^{(+)}}{dW} = F_p(Z, E) \frac{d\sigma_{\text{br}}^{(-)}}{dW}. \quad (3.147)$$

The factor $F_p(Z, E)$ is set equal to the ratio of the radiative stopping powers for positrons and electrons, which has been calculated by Kim *et al.* (1986) (cf. Berger and Seltzer, 1982). In the calculations we use the following analytical approximation

$$\begin{aligned} F_p(Z, E) = & 1 - \exp(-1.2359 \times 10^{-1} t + 6.1274 \times 10^{-2} t^2 - 3.1516 \times 10^{-2} t^3 \\ & + 7.7446 \times 10^{-3} t^4 - 1.0595 \times 10^{-3} t^5 + 7.0568 \times 10^{-5} t^6 \\ & - 1.8080 \times 10^{-6} t^7), \end{aligned} \quad (3.148)$$

where

$$t = \ln \left(1 + \frac{10^6}{Z^2} \frac{E}{m_e c^2} \right). \quad (3.149)$$

Expression (3.148) reproduces the values of $F_p(Z, E)$ tabulated by Kim *et al.* (1986) to an accuracy of about 0.5%.

3.3.2 Integrated cross sections

The total cross section for bremsstrahlung emission is infinite due to the divergence of the DCS (3.140) for small reduced photon energies. Nevertheless, the cross section for emission of photons with reduced energy larger than a given cutoff value W_{cr} is finite. The corresponding mean free path is

$$\lambda_{\text{br}}^{-1}(E; W_{\text{cr}}) \equiv \mathcal{N} \int_{W_{\text{cr}}}^E \frac{d\sigma_{\text{br}}}{dW} dW = \mathcal{N} \frac{Z^2}{\beta^2} \int_{\kappa_{\text{cr}}}^1 \frac{1}{\kappa} \chi(Z, E, \kappa) d\kappa, \quad (3.150)$$

where $\kappa_{\text{cr}} = W_{\text{cr}}/E$. The radiative stopping power and the radiative energy straggling parameter, defined by

$$S_{\text{br}}(E) \equiv \mathcal{N} \int_0^E W \frac{d\sigma_{\text{br}}}{dW} dW = \mathcal{N} \frac{Z^2}{\beta^2} E \int_0^1 \chi(Z, E, \kappa) d\kappa \quad (3.151)$$

and

$$\Omega_{\text{br}}^2(E) \equiv \mathcal{N} \int_0^E W^2 \frac{d\sigma_{\text{br}}}{dW} dW = \mathcal{N} \frac{Z^2}{\beta^2} E^2 \int_0^1 \kappa \chi(Z, E, \kappa) d\kappa, \quad (3.152)$$

are both finite. For the kinetic energies E_i of the grid, these quantities are easily calculated from the tabulated scaled DCS by using linear interpolation in κ . For positrons, the definitions (3.150)–(3.152) must be multiplied by the factor $F_p(Z, E)$ [Eq. (3.148)].

Radiative stopping powers of aluminium, silver and gold for electrons and positrons are shown as functions of the kinetic energy in Fig. 3.14. The stopping powers computed from the DCS given by Eq. (3.140) practically coincide with ICRU37 (1984) values (also Berger and Seltzer, 1982). To leave room for future improvements, PENELOPE reads the radiative stopping power for electrons from the input material data file, and renormalizes the DCS, Eq. (3.140), (*i.e.*, multiplies it by a κ -independent factor) so as to exactly reproduce the input radiative stopping power.

3.3.2.1 CSDA range

As mentioned above, the stopping power gives the average energy loss per unit path length. Thus, when an electron/positron with kinetic energy E advances a small distance ds within a medium, it loses an (average) energy $dE = -S(E) ds$, where

$$S(E) = S_{\text{in}}(E) + S_{\text{br}}(E) = -\frac{dE}{ds} \quad (3.153)$$

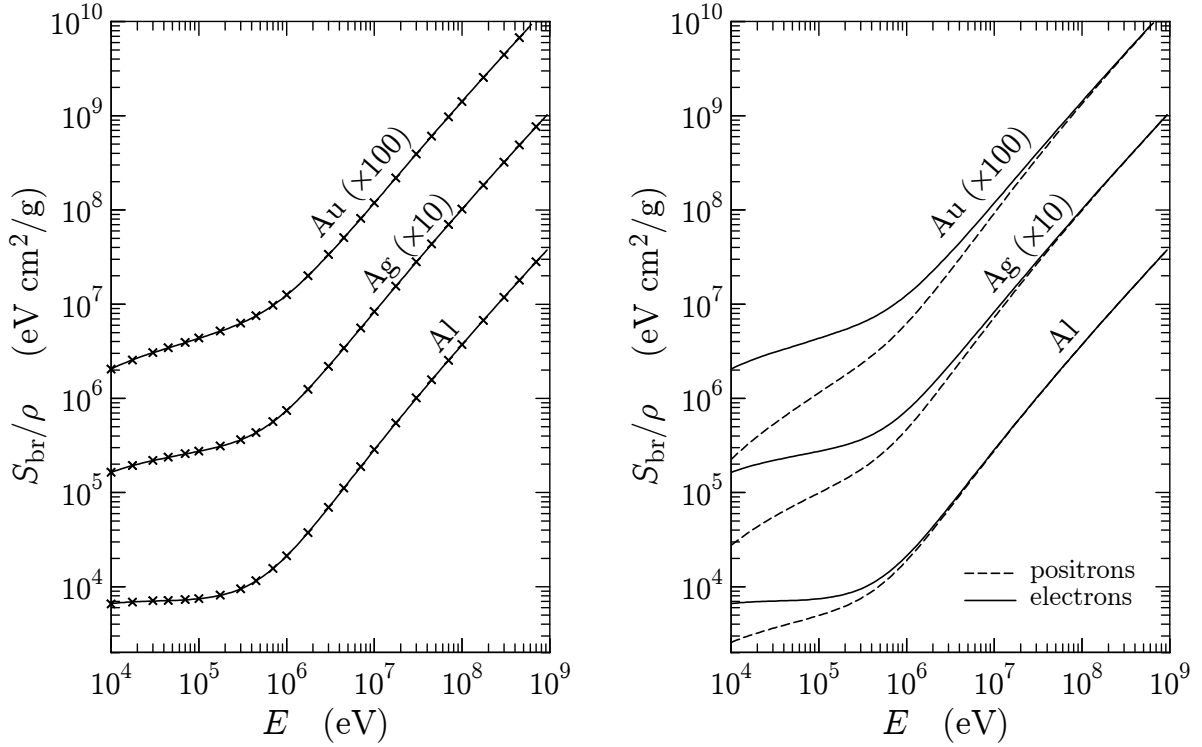


Figure 3.14: Radiative stopping power S_{br}/ρ for electrons and positrons in aluminium, silver ($\times 10$) and gold ($\times 100$) as a function of the kinetic energy. Solid and dashed curves are results from the present model. Crosses are data from the ICRU37 report (1984) (also in Berger and Seltzer, 1982).

is the total (collisional+radiative) stopping power. Many electron transport calculations and old Monte Carlo simulations are based on the so-called continuous slowing down approximation (CSDA), which assumes that particles lose energy in a continuous way and at a rate equal to the stopping power. Evidently, the CSDA disregards energy-loss fluctuations and, therefore, it should be used with caution.

A parameter of practical importance is the so-called CSDA range (or Bethe range), which is defined as the path length travelled by a particle (in an infinite medium) before being absorbed and is given by

$$R(E) = \int_{E_{\text{abs}}}^E \frac{dE'}{S(E')}, \quad (3.154)$$

where we have considered that particles are effectively absorbed when they reach the energy E_{abs} . Notice that the CSDA range gives the *average* path length, actual (or Monte Carlo generated) path lengths fluctuate about the mean $R(E)$; the distribution of ranges has been studied by Lewis (1952). Figure 3.15 displays CSDA ranges for electrons and positrons in aluminium and gold, this information is useful, *e.g.*, in estimating the maximum penetration depth of a beam and for range rejection (a variance-reduction

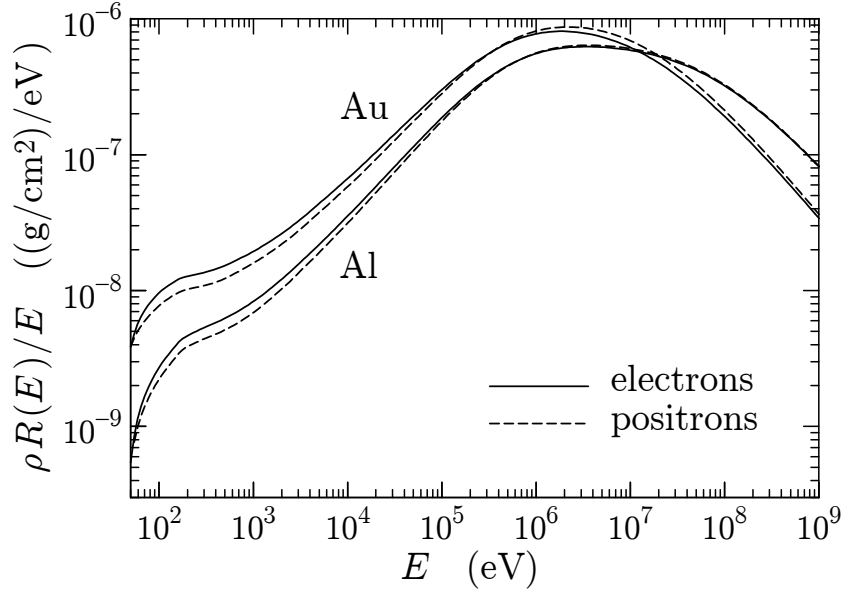


Figure 3.15: CSDA ranges for electrons and positrons in aluminium and gold as functions of the kinetic energy of the particle.

method). Compare Fig. 3.15 with Figs. 3.10 and 3.14 (right plots only) to get a feeling of how differences in stopping power between electrons and positrons are reflected on the CSDA ranges of these particles.

3.3.3 Angular distribution of emitted photons

The direction of the emitted bremsstrahlung photon is defined by the polar angle θ (see Fig. 3.1) and the azimuthal angle ϕ . For isotropic media, with randomly oriented atoms or molecules, the bremsstrahlung DCS is independent of ϕ and can be expressed as

$$\frac{d^2\sigma_{\text{br}}}{dW d(\cos\theta)} = \frac{d\sigma_{\text{br}}}{dW} p_{\text{br}}(Z, E, \kappa; \cos\theta) = \frac{Z^2}{\beta^2} \frac{1}{W} \chi(Z, E, \kappa) p_{\text{br}}(Z, E, \kappa; \cos\theta), \quad (3.155)$$

where $p_{\text{br}}(Z, E, \kappa; \cos\theta)$ is the PDF of $\cos\theta$.

Numerical values of the “shape function” $p_{\text{br}}(Z, E, \kappa; \cos\theta)$, calculated by partial-wave methods, have been published by Kissel *et al.* (1983) for the following benchmark cases: $Z = 2, 8, 13, 47, 79, 92$; $E = 1, 5, 10, 50, 100, 500$ keV and $\kappa = 0, 0.6, 0.8, 0.95$. These authors also gave a parameterisation of the shape function in terms of Legendre polynomials. Unfortunately, their analytical form is not suited for random sampling of the photon direction. In PENELOPE we use a different parameterisation that allows the random sampling of $\cos\theta$ in a simple way. Owing to the lack of numerical data for positrons, it is assumed that the shape function for positrons is the same as for electrons.

In previous simulation studies of x-ray emission from solids bombarded by electron beams (Acosta *et al.*, 1998), the angular distribution of bremsstrahlung photons was

described by means of the semiempirical analytical formulae derived by Kirkpatrick and Wiedmann (1945) [and subsequently modified by Statham (1976)]. These formulae were obtained by fitting the bremsstrahlung DCS derived from Sommerfeld's theory. The shape function obtained from the Kirkpatrick-Wiedmann-Statham fit reads

$$p_{\text{br}}^{(\text{KWS})}(Z, E, \kappa; \cos \theta) = \frac{\sigma_x(1 - \cos^2 \theta) + \sigma_y(1 + \cos^2 \theta)}{(1 - \beta \cos \theta)^2}, \quad (3.156)$$

where the quantities σ_x and σ_y are independent of θ . Although this simple formula predicts the global trends of the partial-wave shape functions of Kissel *et al.* (1983) in certain energy and atomic number ranges, its accuracy is not sufficient for general-purpose simulations. In a preliminary analysis, we tried to improve this formula and determined the parameters σ_x and σ_y by direct fitting to the numerical partial-wave shape functions, but the improvement was not substantial. However, this analysis confirmed that the analytical form (3.156) is flexible enough to approximate the “true” (partial-wave) shape.

The analytical form (3.156) is plausible even for projectiles with relatively high energies, say E larger than 1 MeV, for which the angular distribution of emitted photons is peaked at forward directions. This can be understood by means of the following classical argument (see *e.g.*, Jackson, 1975). Assume that the incident electron is moving in the direction of the z -axis of a reference frame K at rest with respect to the laboratory frame. Let (θ', ϕ') denote the polar and azimuthal angles of the direction of the emitted photon in a reference frame K' that moves with the electron and whose axes are parallel to those of K . In K' , we expect that the angular distribution of the emitted photons will not depart much from the isotropic distribution. To be more specific, we consider the following ansatz (modified dipole distribution) for the shape function in K' ,

$$p_{\text{br,d}}(\cos \theta') = A \frac{3}{8} (1 + \cos^2 \theta') + (1 - A) \frac{3}{4} (1 - \cos^2 \theta'), \quad (0 \leq A \leq 1), \quad (3.157)$$

which is motivated by the relative success of the Kirkpatrick-Wiedmann-Statham formula at low energies (note that the projectile is at rest in K'). The direction of emission (θ, ϕ) in K is obtained by means of the Lorentz transformation

$$\cos \theta = \frac{\cos \theta' + \beta}{1 + \beta \cos \theta'}, \quad \phi = \phi'. \quad (3.158)$$

Thus, the angular distribution in K reads

$$\begin{aligned} p_{\text{br}}(\cos \theta) &= p_{\text{br,d}}(\cos \theta') \frac{d(\cos \theta')}{d(\cos \theta)} \\ &= A \frac{3}{8} \left[1 + \left(\frac{\cos \theta - \beta}{1 - \beta \cos \theta} \right)^2 \right] \frac{1 - \beta^2}{(1 - \beta \cos \theta)^2} \\ &\quad + (1 - A) \frac{3}{4} \left[1 - \left(\frac{\cos \theta - \beta}{1 - \beta \cos \theta} \right)^2 \right] \frac{1 - \beta^2}{(1 - \beta \cos \theta)^2}. \end{aligned} \quad (3.159)$$

Now, it is clear that when β tends to unity, the shape function concentrates at forward directions.

We found that the benchmark partial-wave shape functions of Kissel *et al.* (1983) can be closely approximated by the analytical form (3.159) if one considers A and β as adjustable parameters. Explicitly, we write

$$p_{\text{br,fit}}(\cos \theta) = A \frac{3}{8} \left[1 + \left(\frac{\cos \theta - \beta'}{1 - \beta' \cos \theta} \right)^2 \right] \frac{1 - \beta'^2}{(1 - \beta' \cos \theta)^2} \\ + (1 - A) \frac{3}{4} \left[1 - \left(\frac{\cos \theta - \beta'}{1 - \beta' \cos \theta} \right)^2 \right] \frac{1 - \beta'^2}{(1 - \beta' \cos \theta)^2}, \quad (3.160)$$

with $\beta' = \beta(1 + B)$. The parameters A and B have been determined, by least squares fitting, for the 144 combinations of atomic number, electron energy and reduced photon energy corresponding to the benchmark shape functions tabulated by Kissel *et al.* (1983). Results of this fit are compared with the original partial-wave shape functions in Fig. 3.16. The largest differences between the fits and the data were found for the higher atomic numbers, but even then the fits are very accurate, as shown in Fig. 3.16. The quantities $\ln(AZ\beta)$ and $B\beta$ vary smoothly with Z , β and κ and can be obtained by cubic spline interpolation of their values for the benchmark cases. This permits the fast evaluation of the shape function for any combination of Z , β and κ . Moreover, the random sampling of the photon direction, *i.e.*, of $\cos \theta$, can be performed by means of a simple, fast analytical algorithm (see below). For electrons with kinetic energies larger than 500 keV, the shape function is approximated by the classical dipole distribution, *i.e.*, by the analytical form (3.160) with $A = 1$ and $\beta' = \beta$.

3.3.4 Simulation of hard radiative events

Let us now consider the simulation of hard radiative events ($W > W_{\text{cr}}$) from the DCS defined by Eqs. (3.155) and (3.160). PENELOPE reads the scaled bremsstrahlung DCS from the database files and, by natural cubic spline interpolation/extrapolation in $\ln E$, produces a table for a denser logarithmic grid of 200 energies (and for the “standard” mesh of 32 κ ’s), which is stored in memory. This energy grid spans the full energy range considered in the simulation and allows accurate (and fast) linear interpolation of the scaled DCS in the variable $\ln E$, which is more adequate than E when interpolation over a wide energy interval is required.

Notice that in the Monte Carlo simulation the kinetic energy of the transported electron (or positron) varies in a random way and may take arbitrary values within a certain domain. Hence, we must be able to simulate bremsstrahlung emission by electrons with energies E not included in the grid.

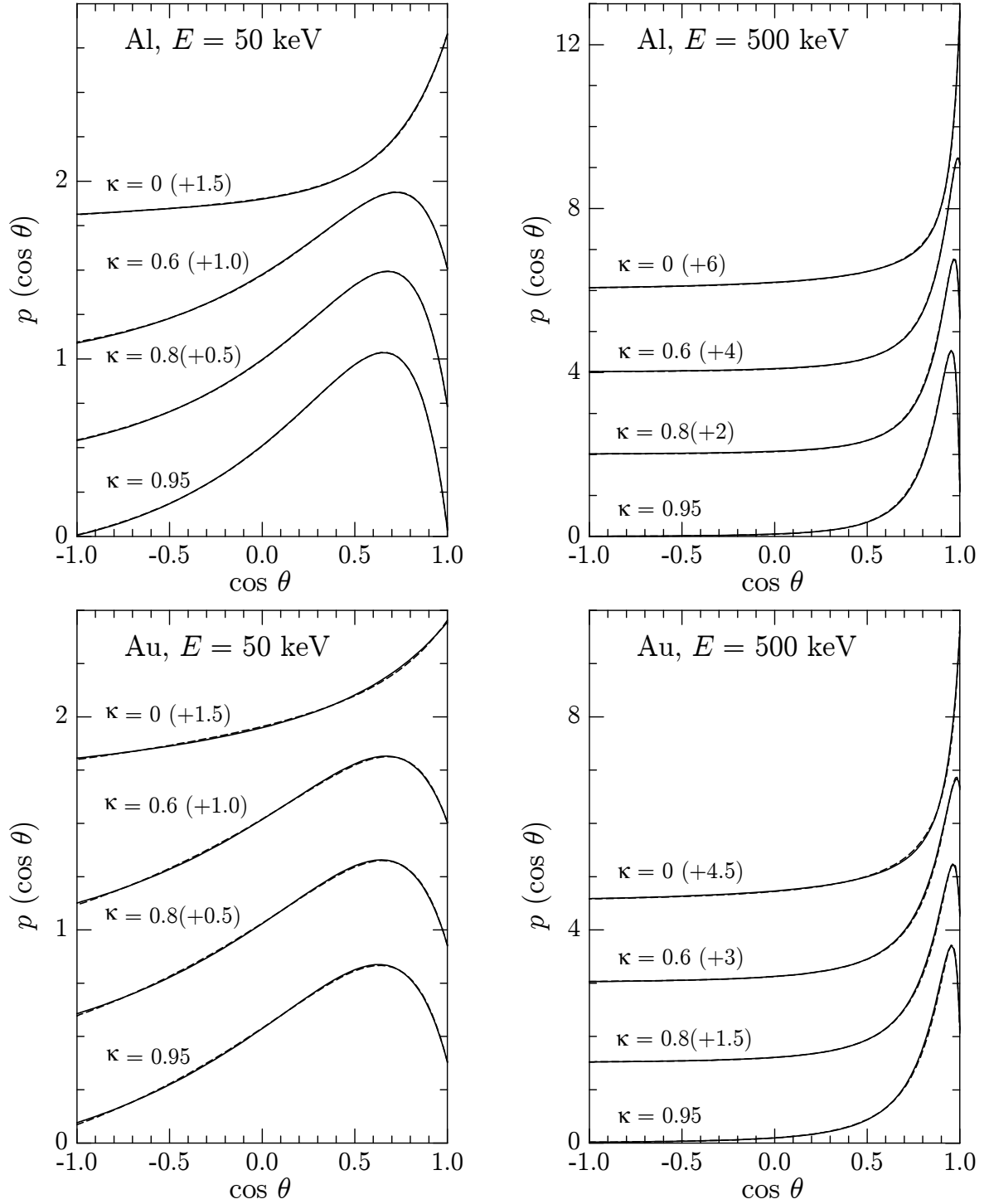


Figure 3.16: Shape functions (angular distributions) for bremsstrahlung emission by electrons of the indicated energies in the fields of aluminium and gold atoms. Dashed curves are partial-wave shape functions of Kissel *et al.* (1983). Continuous curves are the present analytical fits, Eq. (3.160). For visual aid, some curves have been shifted upwards in the amounts indicated in parentheses.

3.3.4.1 Sampling of the photon energy

The PDF for the reduced photon energy, $\kappa = W/E$, is given by [see Eq. (3.140)]

$$p_{\text{br}}(E, \kappa) = \frac{1}{\kappa} \chi(Z, E, \kappa) \Theta(\kappa - \kappa_{\text{cr}}) \Theta(1 - \kappa), \quad (3.161)$$

where $\kappa_{\text{cr}} = W_{\text{cr}}/E$ and $\chi(Z, E, \kappa)$ is calculated by linear interpolation, in both $\ln E$ and κ , in the stored table. That is, $\chi(Z, E, \kappa)$ is considered to be a piecewise linear function of κ . To sample κ from the PDF (3.161) for an energy E_i in the grid, we express the interpolated scaled DCS as

$$\chi(Z, E_i, \kappa) = a_j + b_j \kappa \quad \text{if } \kappa_j \leq \kappa \leq \kappa_{j+1}, \quad (3.162)$$

and introduce the cumulative distribution function,

$$\mathcal{P}_j = \int_{\kappa_{\text{cr}}}^{\kappa_j} p(E_i, \kappa) d\kappa, \quad (3.163)$$

which, for a piecewise linear χ , can be computed exactly. We also define

$$\chi_{\text{max},j} = \max \left\{ \chi(Z, E, \kappa), \kappa \in (\kappa_j, \kappa_{j+1}) \right\} \quad j = 1, \dots, 32. \quad (3.164)$$

With all this we can formulate the following sampling algorithm, which combines a numerical inverse transform and a rejection,

(i) Generate a random number ξ and determine the index j for which $\mathcal{P}_j \leq \xi \mathcal{P}_{32} \leq \mathcal{P}_{j+1}$ using the binary-search method.

(ii) Sample κ from the distribution κ^{-1} in the interval (κ_j, κ_{j+1}) , *i.e.*,

$$\kappa = \kappa_j (\kappa_{j+1}/\kappa_j)^\xi. \quad (3.165)$$

(iii) Generate a new random number ξ . If $\xi \chi_{\text{max},j} < a_j + b_j \kappa$, deliver κ .

(iv) Go to step (i).

This sampling algorithm is exact and very fast [notice that the binary search in step (i) requires at most 5 comparisons], but is only applicable for the energies in the grid where χ is tabulated.

To simulate bremsstrahlung emission by electrons with energies E not included in the grid, we should first obtain the PDF $p_{\text{br}}(E, \kappa)$ by interpolation along the energy axis and then perform the random sampling of κ from this PDF using the algorithm described above. This procedure is too time consuming. A faster method consists of assuming that the grid of energies is dense enough so that linear interpolation in $\ln E$ is sufficiently accurate. If $E_i < E < E_{i+1}$, we can express the interpolated PDF as [cf. Eq. (3.22)]

$$p_{\text{br,int}}(E, \kappa) = \pi_i p_{\text{br}}(E_i, \kappa) + \pi_{i+1} p_{\text{br}}(E_{i+1}, \kappa) \quad (3.166)$$

with the interpolation weights

$$\pi_i = \frac{\ln E_{i+1} - \ln E}{\ln E_{i+1} - \ln E_i}, \quad \pi_{i+1} = \frac{\ln E - \ln E_i}{\ln E_{i+1} - \ln E_i}. \quad (3.167)$$

These weights are positive and add to unity, *i.e.*, they can be interpreted as point probabilities. Therefore, to perform the random sampling of κ from $p_{\text{br,int}}(E, \kappa)$ we can employ the composition method (Section 1.2.6), which leads to the following algorithm:

- (i) Sample the integer variable k , which can take the values i or $i + 1$ with point probabilities π_i and π_{i+1} , respectively.
- (ii) Sample κ from the distribution $p_{\text{br}}(E_k, \kappa)$.

With this interpolation-by-weight method we only need to sample κ from the tabulated PDFs, *i.e.*, for the energies E_i of the grid.

3.3.4.2 Angular distribution of emitted photons

The random sampling of $\cos \theta$ is simplified by noting that the PDF given by Eq. (3.160) results from a Lorentz transformation, with speed β' , of the PDF (3.157). This means that we can sample the photon direction $\cos \theta'$ in the reference frame K' from the PDF (3.157) and then apply the transformation (3.158) (with β' instead of β) to get the direction $\cos \theta$ in the laboratory frame.

To generate random values of $\cos \theta$ from (3.160) we use the following algorithm, which combines the composition and rejection methods,

- (i) Sample a random number ξ_1 .
- (ii) If $\xi_1 < A$, then
 - 1) Sample a random number ξ and set $\cos \theta' = -1 + 2\xi$.
 - 2) Sample a random number ξ .
 - 3) If $2\xi > 1 + \cos^2 \theta'$, go to 1).
- (iii) If $\xi_1 \geq A$, then
 - 4) Sample a random number ξ and set $\cos \theta' = -1 + 2\xi$.
 - 5) Sample a random number ξ .
 - 6) If $\xi > 1 - \cos^2 \theta'$, go to 4).
- (iv) Deliver $\cos \theta = \frac{\cos \theta' + \beta'}{1 + \beta' \cos \theta'}$.

The efficiencies of the rejections in steps (ii) and (iii) are both equal to 0.66. That is, on average, we need 4 random numbers to generate each value of $\cos \theta$.

3.4 Positron annihilation

Following Nelson *et al.* (1985), we consider that positrons penetrating a medium of atomic number Z with kinetic energy E can annihilate with the electrons in the medium by emission of two photons. We assume that the target electrons are free and at rest, thus disregarding electron binding effects, which enable one-photon annihilation (Heitler, 1954). When annihilation occurs in flight, *i.e.*, when the kinetic energy E of the positron is larger than the “absorption” energy, the two photons may have different energies, say E_- and E_+ , which add to $E + 2m_e c^2$. In what follows, quantities referring to the photon with the lowest energy will be denoted by the subscript “-”. Each annihilation event is then completely characterised by the quantity

$$\zeta \equiv \frac{E_-}{E + 2m_e c^2}. \quad (3.168)$$

Assuming that the positron moves initially in the direction of the z -axis, from conservation of energy and momentum it follows that the two photons are emitted in directions with polar angles [see Eqs. (A.21) and (A.22) in Appendix A]

$$\cos \theta_- = (\gamma^2 - 1)^{-1/2} (\gamma + 1 - 1/\zeta) \quad (3.169)$$

and

$$\cos \theta_+ = (\gamma^2 - 1)^{-1/2} [\gamma + 1 - 1/(1 - \zeta)], \quad (3.170)$$

and azimuthal angles ϕ_- and $\phi_+ = \phi_- + \pi$. The quantity $\gamma = 1 + E/(m_e c^2)$ is the total energy of the positron in units of its rest energy.

The maximum value of ζ is $1/2$, its minimum value is found when $\cos \theta_- = -1$ and is given by

$$\zeta_{\min} = \frac{1}{\gamma + 1 + (\gamma^2 - 1)^{1/2}}. \quad (3.171)$$

The DCS (per electron) for two-photon annihilation, as observed in the centre-of-mass system of the positron and the electron, is given by Heitler (1954). Nelson *et al.* (1985) transformed this DCS to the laboratory system (where the electron is at rest), their result can be written as

$$\frac{d\sigma_{\text{an}}}{d\zeta} = \frac{\pi r_e^2}{(\gamma + 1)(\gamma^2 - 1)} [S(\zeta) + S(1 - \zeta)], \quad (3.172)$$

where

$$S(\zeta) = -(\gamma + 1)^2 + (\gamma^2 + 4\gamma + 1) \frac{1}{\zeta} - \frac{1}{\zeta^2}. \quad (3.173)$$

Owing to the axial symmetry of the process, the DCS is independent of the azimuthal angle ϕ_- , which is uniformly distributed on the interval $(0, 2\pi)$. For fast positrons, annihilation photons are emitted preferentially at forward directions. When the kinetic energy of the positron decreases, the angular distribution of the generated photons becomes more isotropical (see Fig. 3.17).

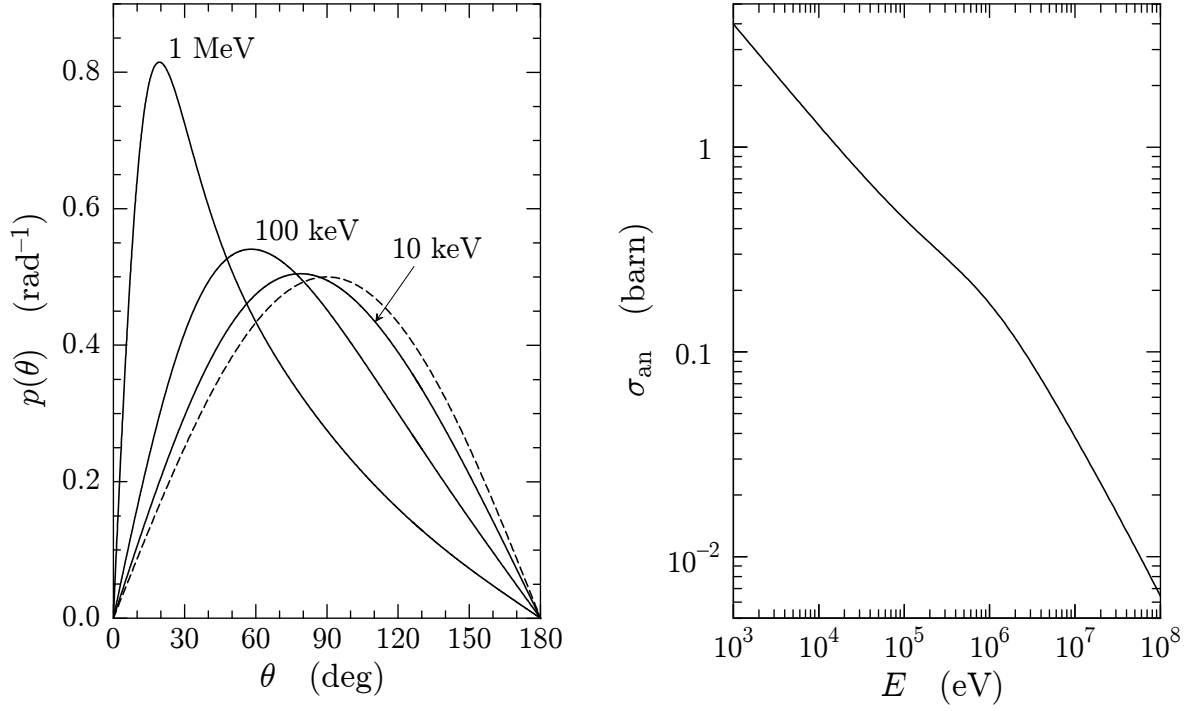


Figure 3.17: Left: angular distributions of photons produced by annihilation in flight of positrons with the indicated kinetic energies. The dashed line represents the isotropic distribution. Right: Annihilation cross section per target electron as a function of the kinetic energy of the positron.

The cross section (per target electron) for two-photon annihilation is

$$\sigma_{\text{an}} = \int_{\zeta_{\min}}^{1/2} \frac{d\sigma_{\text{an}}}{d\zeta} d\zeta = \frac{\pi r_e^2}{(\gamma + 1)(\gamma^2 - 1)} \times \left\{ (\gamma^2 + 4\gamma + 1) \ln \left[\gamma + (\gamma^2 - 1)^{1/2} \right] - (3 + \gamma) (\gamma^2 - 1)^{1/2} \right\}. \quad (3.174)$$

The annihilation mean free path is given by

$$\lambda_{\text{an}}^{-1} = \mathcal{N}Z\sigma_{\text{an}}, \quad (3.175)$$

where $\mathcal{N}Z$ is the density of electrons in the medium. The annihilation cross section is displayed in Fig. 3.17. The cross section decreases with the kinetic energy and, therefore, high-energy positrons can travel path lengths of the order of the CSDA range before annihilating.

3.4.1 Generation of emitted photons

The PDF of ζ is given by (normalisation is irrelevant here)

$$p_{\text{an}}(\zeta) = S(\zeta) + S(1 - \zeta), \quad \zeta_{\min} \leq \zeta \leq 1/2. \quad (3.176)$$

To sample ζ , we may take advantage of the symmetry of this expression under the exchange of the two photons, which corresponds to exchanging ζ and $1 - \zeta$. We first consider the distribution

$$P(v) \equiv S(v), \quad \zeta_{\min} \leq v \leq 1 - \zeta_{\min} \quad (3.177)$$

and write it in the form

$$P(v) = \pi(v)g(v) \quad (3.178)$$

with

$$\pi(v) = \left[\ln \left(\frac{1 - \zeta_{\min}}{\zeta_{\min}} \right) \right]^{-1} \frac{1}{v} \quad (3.179)$$

and

$$g(v) = \left[-(\gamma + 1)^2 v + (\gamma^2 + 4\gamma + 1) - \frac{1}{v} \right]. \quad (3.180)$$

$\pi(v)$ is a proper PDF (*i.e.*, it is definite positive and normalised to unity) and $g(v)$ is a monotonically decreasing function. Random values of v from the distribution $P(v)$ can be generated by using the following algorithm (rejection method):

- (i) Sample a value v from the distribution $\pi(v)$. This is easily done with the inverse-transform method, which yields the following sampling equation

$$v = \zeta_{\min} \left(\frac{1 - \zeta_{\min}}{\zeta_{\min}} \right)^{\xi}. \quad (3.181)$$

- (ii) Generate a new random number ξ .

- (iii) If $\xi g(\zeta_{\min}) > g(v)$, go to step (i).

- (iv) Deliver v .

It is clear that the random value

$$\zeta = \min(v, 1 - v) \quad (3.182)$$

follows the distribution given by Eq. (3.176) when v is sampled from the distribution $P(v)$. The efficiency of this sampling algorithm practically equals 100% for positrons with kinetic energy E less than 10 keV, decreases when E increases to reach a minimum value of $\sim 80\%$ at $E \sim 10$ MeV and increases monotonically for larger energies.

As the result of annihilation, two photons with energies $E_- = \zeta(E + 2m_e c^2)$ and $E_+ = (1 - \zeta)(E + 2m_e c^2)$ are emitted in the directions given by Eqs. (3.169) and (3.170).

Chapter 4

Electron/positron transport mechanics

In principle, the scattering model and sampling techniques described in Chapter 3 allow the detailed Monte Carlo simulation of electron and positron transport in matter. However, detailed simulation is feasible only when the mean number of interactions per track is small (a few hundred at most). This occurs for electrons with low initial kinetic energies or for thin geometries. The number of interactions experienced by an electron or positron before being effectively stopped increases with its initial energy and, therefore, detailed simulation becomes impractical at high energies.

PENELOPE implements a “mixed” simulation scheme (Berger, 1963; Reimer and Krefting, 1976; Andreo and Brahme, 1984), which combines the detailed simulation of hard events (*i.e.*, events with polar angular deflection θ or energy loss W larger than previously selected cutoff values θ_c and W_c) with condensed simulation of soft events, in which $\theta < \theta_c$ or $W < W_c$. Owing to the fact that for high-energy electrons the DCSs for the various interaction processes decrease rapidly with the polar scattering angle and the energy loss, cutoff values can be selected such that the mean number of hard events per electron track is sufficiently small to permit their detailed simulation. In general, this is accomplished by using relatively small cutoff values, so that each soft interaction has only a slight effect on the simulated track. The global effect of the (usually many) soft interactions that take place between each pair of consecutive hard events can then be simulated accurately by using a multiple-scattering approach. Hard events occur much less frequently than soft events, but they have severe effects on the track evolution (*i.e.*, they cause large angular deflections and lateral displacements or considerable energy losses), which can only be properly reproduced by detailed simulation. The computer time needed to simulate each track diminishes rapidly when the cutoff values for the angular deflection and the energy loss are increased. Mixed simulation algorithms are usually very stable under variations of the adopted cutoff values, whenever these are kept below some reasonable limits. Mixed simulation is then preferable to condensed simulation because 1) spatial distributions are simulated more accurately, 2) tracks in the vicinity of interfaces are properly handled, and 3) possible dependence of the results on user-defined parameters is largely reduced.

4.1 Elastic scattering

Let us start by considering electrons (or positrons) with kinetic energy E moving in a hypothetical infinite homogeneous medium, with $\mathcal{N}$ scattering centres per unit volume, in which they experience only pure elastic collisions (*i.e.*, with no energy loss).

4.1.1 Multiple elastic scattering theory

Assume that an electron starts off from a certain position, which we select as the origin of our reference frame, moving in the direction of the z -axis. Let $f(s; \mathbf{r}, \hat{\mathbf{d}})$ denote the probability density of finding the electron at the position $\mathbf{r} = (x, y, z)$, moving in the direction given by the unit vector $\hat{\mathbf{d}}$ after having travelled a path length s . The diffusion equation for this problem is (Lewis, 1950)

$$\frac{\partial f}{\partial s} + \hat{\mathbf{d}} \cdot \nabla f = \mathcal{N} \int \left[f(s; \mathbf{r}, \hat{\mathbf{d}}') - f(s; \mathbf{r}, \hat{\mathbf{d}}) \right] \frac{d\sigma_{\text{el}}(\theta)}{d\Omega} d\Omega, \quad (4.1)$$

where $\theta \equiv \arccos(\hat{\mathbf{d}} \cdot \hat{\mathbf{d}}')$ is the scattering angle corresponding to the angular deflection $\hat{\mathbf{d}}' \rightarrow \hat{\mathbf{d}}$. This equation has to be solved with the boundary condition $f(0; \mathbf{r}, \hat{\mathbf{d}}) = (1/\pi)\delta(\mathbf{r})\delta(1 - \cos \chi)$, where χ is the polar angle of the direction $\hat{\mathbf{d}}$. By expanding $f(s; \mathbf{r}, \hat{\mathbf{d}})$ in spherical harmonics, Lewis (1950) obtained exact expressions for the angular distribution and for the first moments of the spatial distribution after a given path length s . The probability density $F(s; \chi)$ of having a final direction in the solid angle element $d\Omega$ around a direction defined by the polar angle χ is given by

$$F(s; \chi) = \int f(s; \mathbf{r}, \hat{\mathbf{d}}) d\mathbf{r} = \sum_{\ell=0}^{\infty} \frac{2\ell+1}{4\pi} \exp(-s/\lambda_{\text{el},\ell}) P_{\ell}(\cos \chi), \quad (4.2)$$

where $P_{\ell}(\cos \chi)$ are Legendre polynomials and $\lambda_{\text{el},\ell} = 1/(\mathcal{N}\sigma_{\text{el},\ell})$ is the ℓ -th transport mean free path defined by Eq. (3.16). The result given by Eq. (4.2) coincides with the multiple-scattering distribution obtained by Goudsmit and Saunderson (1940a, 1940b). Evidently, the distribution $F(s; \chi)$ is symmetric about the z -axis, *i.e.*, independent of the azimuthal angle of the final direction.

From the orthogonality of the Legendre polynomials, it follows that

$$\langle P_{\ell}(\cos \chi) \rangle \equiv 2\pi \int_{-1}^1 P_{\ell}(\cos \chi) F(s; \chi) d(\cos \chi) = \exp(-s/\lambda_{\text{el},\ell}). \quad (4.3)$$

In particular, we have

$$\langle \cos \chi \rangle = \exp(-s/\lambda_{\text{el},1}) \quad (4.4)$$

and

$$\langle \cos^2 \chi \rangle = \frac{1}{3} [1 + 2 \exp(-s/\lambda_{\text{el},2})]. \quad (4.5)$$

Lewis (1950) also derived analytical formulas for the first moments of the spatial distribution and the correlation function of z and $\cos \chi$. Neglecting energy losses, the results explicitly given in Lewis' paper simplify to

$$\langle z \rangle \equiv 2\pi \int z f(s; \mathbf{r}, \hat{\mathbf{d}}) d(\cos \chi) d\mathbf{r} = \lambda_{\text{el},1} [1 - \exp(-s/\lambda_{\text{el},1})], \quad (4.6)$$

$$\begin{aligned} \langle x^2 + y^2 \rangle &\equiv 2\pi \int (x^2 + y^2) f(s; \mathbf{r}, \hat{\mathbf{d}}) d(\cos \chi) d\mathbf{r} \\ &= \frac{4}{3} \int_0^s dt \exp(-t/\lambda_{\text{el},1}) \int_0^t [1 - \exp(-u/\lambda_{\text{el},2})] \exp(u/\lambda_{\text{el},1}) du, \end{aligned} \quad (4.7)$$

$$\begin{aligned} \langle z \cos \chi \rangle &\equiv 2\pi \int z \cos \chi f(s; \mathbf{r}, \hat{\mathbf{d}}) d(\cos \chi) d\mathbf{r} \\ &= \exp(-s/\lambda_{\text{el},1}) \int_0^s [1 + 2 \exp(-t/\lambda_{\text{el},2})] \exp(t/\lambda_{\text{el},1}) dt. \end{aligned} \quad (4.8)$$

It is worth observing that the quantities (4.4)–(4.8) are completely determined by the values of the transport mean free paths $\lambda_{\text{el},1}$ and $\lambda_{\text{el},2}$; they are independent of the elastic mean free path λ_{el} .

4.1.2 Mixed simulation of elastic scattering

At high energies, where detailed simulation becomes impractical, $\lambda_{\text{el},1} \gg \lambda_{\text{el}}$ (see Fig. 3.3) so that the average angular deflection in each collision is small. In other words, the great majority of elastic collisions of fast electrons are soft collisions with very small deflections. We shall consider mixed simulation procedures (see Fernández-Varea *et al.*, 1993b; Baró *et al.*, 1994b) in which hard collisions, with scattering angle θ larger than a certain value θ_c , are individually simulated and soft collisions (with $\theta < \theta_c$) are described by means of a multiple-scattering approach.

In practice, the mixed algorithm will be defined by specifying the mean free path $\lambda_{\text{el}}^{(\text{h})}$ between hard elastic events, defined by [see Eq. (3.25)]

$$\frac{1}{\lambda_{\text{el}}^{(\text{h})}} = \mathcal{N} 2\pi \int_{\theta_c}^{\pi} \frac{d\sigma_{\text{el}}(\theta)}{d\Omega} \sin \theta d\theta. \quad (4.9)$$

This equation determines the cutoff angle θ_c as a function of $\lambda_{\text{el}}^{(\text{h})}$. A convenient recipe to set the mean free path $\lambda_{\text{el}}^{(\text{h})}$ is

$$\lambda_{\text{el}}^{(\text{h})}(E) = \max \{ \lambda_{\text{el}}(E), C_1 \lambda_{\text{el},1}(E) \}, \quad (4.10)$$

where C_1 is a pre-selected small constant (say, less than ~ 0.1). For increasing energies, λ_{el} attains a constant value and $\lambda_{\text{el},1}$ increases steadily (see Fig. 3.3) so that the formula (4.10) gives a mean free path for hard collisions that increases with energy, *i.e.*, hard

collisions are less frequent when the scattering effect is weaker. The recipe (4.10) also ensures that $\lambda_{\text{el}}^{(\text{h})}$ will reduce to the actual mean free path λ_{el} for low energies. In this case, soft collisions cease to occur ($\theta_c = 0$) and mixed simulation becomes purely detailed. It is worth noticing that, when mixed simulation is effective (*i.e.*, when $\lambda_{\text{el}}^{(\text{h})} > \lambda_{\text{el}}$), the mean angular deflection in a path length $\lambda_{\text{el}}^{(\text{h})}$ is [see Eq. (4.4)]

$$1 - \langle \cos \chi \rangle = 1 - \exp(-\lambda_{\text{el}}^{(\text{h})}/\lambda_{\text{el},1}) \simeq C_1. \quad (4.11)$$

Hence, when using the prescription (4.10), the average angular deflection due to all elastic collisions occurring along a path length $\lambda_{\text{el}}^{(\text{h})}$ equals C_1 .

The PDF of the step length s between two successive hard collisions is

$$p(s) = \frac{1}{\lambda_{\text{el}}^{(\text{h})}} \exp(-s/\lambda_{\text{el}}^{(\text{h})}), \quad (4.12)$$

and random values of s can be generated by means of the sampling formula, Eq. (1.36)

$$s = -\lambda_{\text{el}}^{(\text{h})} \ln \xi. \quad (4.13)$$

The (unnormalized) PDF of the polar deflection θ in single hard collisions is

$$p^{(\text{h})}(\theta) = \frac{d\sigma_{\text{el}}(\theta)}{d\Omega} \sin \theta \Theta(\theta - \theta_c), \quad (4.14)$$

where $\Theta(x)$ stands for the step function.

The inverse transport mean free paths $\lambda_{\text{el},\ell}^{-1}$, see Eqs. (3.14) and (3.16), for the actual scattering process can be split into contributions from soft and hard collisions, that is

$$\frac{1}{\lambda_{\text{el},\ell}} = \frac{1}{\lambda_{\text{el},\ell}^{(\text{s})}} + \frac{1}{\lambda_{\text{el},\ell}^{(\text{h})}}, \quad (4.15)$$

where

$$\frac{1}{\lambda_{\text{el},\ell}^{(\text{s})}} = \mathcal{N} 2\pi \int_0^{\theta_c} [1 - P_\ell(\cos \theta)] \frac{d\sigma_{\text{el}}(\theta)}{d\Omega} \sin \theta d\theta \quad (4.16a)$$

and

$$\frac{1}{\lambda_{\text{el},\ell}^{(\text{h})}} = \mathcal{N} 2\pi \int_{\theta_c}^{\pi} [1 - P_\ell(\cos \theta)] \frac{d\sigma_{\text{el}}(\theta)}{d\Omega} \sin \theta d\theta. \quad (4.16b)$$

Let us assume that an electron starts off from the origin of coordinates moving in the direction of the z -axis and undergoes the first hard collision after travelling a path length s . The exact angular distribution produced by the soft collisions along this step is

$$F^{(\text{s})}(s; \chi) = \sum_{\ell=0}^{\infty} \frac{2\ell+1}{4\pi} \exp(-s/\lambda_{\text{el},\ell}^{(\text{s})}) P_\ell(\cos \chi). \quad (4.17)$$

The exact average longitudinal and transverse displacements at the end of the step are given by [see Eqs. (4.6) and (4.7)]

$$\langle z \rangle^{(s)} = \lambda_{\text{el},1}^{(s)} \left[1 - \exp(-s/\lambda_{\text{el},1}^{(s)}) \right] = s \left[1 - \frac{1}{2} \left(\frac{s}{\lambda_{\text{el},1}^{(s)}} \right) + \frac{1}{6} \left(\frac{s}{\lambda_{\text{el},1}^{(s)}} \right)^2 - \dots \right], \quad (4.18)$$

$$\langle x^2 + y^2 \rangle^{(s)} = \frac{2}{9} \frac{s^3}{\lambda_{\text{el},2}^{(s)}} \left[1 - \frac{1}{4} \left(1 + \frac{\lambda_{\text{el},1}^{(s)}}{\lambda_{\text{el},2}^{(s)}} \right) \left(\frac{s}{\lambda_{\text{el},1}^{(s)}} \right) + \dots \right], \quad (4.19)$$

where $\lambda_{\text{el},1}^{(s)}$, the first transport mean free path for soft collisions, is larger than $\lambda_{\text{el},1}$. As the mean free path between hard collisions is normally much less than $\lambda_{\text{el},1}^{(s)}$ (depending on the value of C_1), the value $s/\lambda_{\text{el},1}^{(s)}$ is, on average, much less than unity (note that $\langle s \rangle = \lambda_{\text{el}}^{(h)}$). Therefore, the global effect of the soft collisions in the step, *i.e.*, the change in direction of movement *and* the lateral displacement, is very small (part of the deflection is caused by the hard interaction at the end of the step).

In PENELOPE, 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 method¹ (Fernández-Varea *et al.*, 1993b). The associated algorithm can be formulated as follows (see Fig. 4.1),

- (i) The electron first moves a random distance τ , which is sampled uniformly in the interval $(0, s)$, in the initial direction.
- (ii) Then a single artificial soft scattering event (a hinge) takes place, in which the electron changes its direction of movement according to the multiple-scattering distribution $F^{(s)}(s; \chi)$.
- (iii) Finally, the electron moves a distance $s - \tau$ in the new direction.

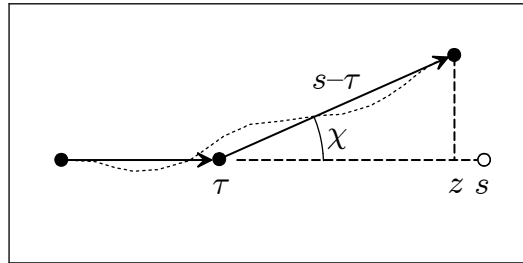


Figure 4.1: Simulation of the global effect of soft collisions between two consecutive hard collisions by the random-hinge method.

¹The name was coined by Ron Kensek.

Obviously, this algorithm leads to the exact angular distribution at the end of the step. The average longitudinal displacement at the end of the simulated step is

$$\langle z \rangle_{\text{sim}}^{(s)} = \frac{s}{2} + \frac{s}{2} \langle \cos \chi \rangle^{(s)} = s \left[1 - \frac{1}{2} \left(\frac{s}{\lambda_{\text{el},1}^{(s)}} \right) + \frac{1}{4} \left(\frac{s}{\lambda_{\text{el},1}^{(s)}} \right)^2 - \dots \right], \quad (4.20)$$

which agrees closely with the exact result given by Eq. (4.18). Moreover, the average simulated transverse displacement is

$$\begin{aligned} \langle x^2 + y^2 \rangle_{\text{sim}}^{(s)} &= \langle (s - \tau)^2 \sin^2 \chi \rangle_{\text{sim}}^{(s)} = \frac{1}{3} s^2 (1 - \langle \cos^2 \chi \rangle^{(s)}) \\ &= \frac{2}{9} \frac{s^3}{\lambda_{\text{el},2}^{(s)}} \left[1 - \frac{1}{2} \frac{\lambda_{\text{el},1}^{(s)}}{\lambda_{\text{el},2}^{(s)}} \left(\frac{s}{\lambda_{\text{el},1}^{(s)}} \right) + \dots \right], \end{aligned} \quad (4.21)$$

which does not differ much from the exact value given by Eq. (4.19). From these facts, we may conclude that the random-hinge method provides a faithful description of the transport when the step length s is much shorter than the first transport mean free path $\lambda_{\text{el},1}$, so that the global angular deflection and lateral displacement are small. Surprisingly, it does work well also in condensed (class I) simulations, where this requirement is not met. In spite of its simplicity, the random-hinge method competes in accuracy and speed with other, much more sophisticated transport algorithms (see Bielajew and Salvat, 2001, and references therein). It seems that the randomness of the hinge position τ leads to correlations between the angular deflection and the displacement that are close to the actual correlations.

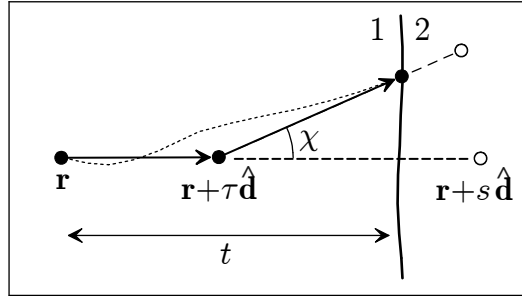


Figure 4.2: Simulation of a track near the crossing of an interface.

The random-hinge algorithm can be readily adapted to simulate multiple-scattering processes in limited material structures, which may consist of several regions of different compositions separated by well-defined surfaces (interfaces). In these geometries, when the track crosses an interface, we simply stop it at the crossing point, and resume the simulation in the new material. In spite of its simplicity, this recipe gives a fairly accurate description of interface crossing. To see this, consider that a hard collision has occurred at the position $\mathbf{r}$ in region “1” and assume that the following hard collision

occurs in region “2”. The step length s between these two hard collisions is larger than the distance t from $\mathbf{r}$ to the interface (see Fig. 4.2). If the artificial soft elastic collision occurs in region “1”, the angular deflection in this collision is sampled from the distribution $F^{(s)}(s; \chi)$. Otherwise, the electron reaches the interface without changing its direction of movement. Assuming $s \ll \lambda_{\text{el},1}^{(s)}$, the mean angular deflection due to soft collisions is

$$1 - \langle \cos \chi \rangle^{(s)} = 1 - \exp(-s/\lambda_{\text{el},1}^{(s)}) \simeq \frac{s}{\lambda_{\text{el},1}^{(s)}}. \quad (4.22)$$

Moreover, when this assumption is valid, lateral displacements due to soft collisions are small and can be neglected to a first approximation. As the probability for the soft collision to occur within region “1” equals t/s , the average angular deflection of the simulated electron track when it reaches the interface is

$$1 - \langle \cos \chi \rangle = \frac{t}{s} (1 - \langle \cos \chi \rangle^{(s)}) \simeq \frac{t}{\lambda_{\text{el},1}^{(s)}}, \quad (4.23)$$

which practically coincides with the exact mean deviation after the path length t within region “1”, as required. Thus, by sampling the position of the soft collision uniformly in the segment $(0, s)$ we make sure that the electron reaches the interface with the correct average direction of movement.

4.1.2.1 Angular deflections in soft scattering events

In the random-hinge method, the global effect of the soft collisions experienced by the particle along a path segment of length s between two consecutive hard events is simulated as a single artificial soft scattering event. The angular deflection follows the multiple-scattering distribution $F^{(s)}(s; \chi)$. Unfortunately, the exact Legendre expansion, Eq. (4.17), is not appropriate for Monte Carlo simulation, since this expansion converges very slowly (because the associated single-scattering DCS is not continuous) and the sum varies rapidly with the path length s .

Whenever the cutoff angle θ_c is small, the distribution $F^{(s)}(s; \chi)$ may be calculated by using the small-angle approximation (see, *e.g.*, Lewis, 1950). Notice that θ_c can be made as small as desired by selecting a small enough value of C_1 , see Eqs. (4.9) and (4.10). Introducing the limiting form of the Legendre polynomials

$$P_\ell(\cos \theta) \simeq 1 - \frac{1}{4} \ell(\ell + 1) \theta^2 \quad (4.24)$$

into Eq. (4.16a) we get

$$\frac{1}{\lambda_{\text{el},\ell}^{(s)}} = \mathcal{N} 2\pi \frac{\ell(\ell + 1)}{4} \int_0^{\theta_c} \theta^2 \frac{d\sigma_{\text{el}}(\theta)}{d\Omega} \sin \theta d\theta = \frac{\ell(\ell + 1)}{2} \frac{1}{\lambda_{\text{el},1}^{(s)}}, \quad (4.25)$$

i.e., the transport mean free paths $\lambda_{\text{el},\ell}^{(s)}$ are completely determined by the single value

$\lambda_{\text{el},1}^{(s)}$. The angular distribution $F^{(s)}$ then simplifies to

$$F^{(s)}(s; \chi) = \sum_{\ell=0}^{\infty} \frac{2\ell+1}{4\pi} \exp \left[-\frac{\ell(\ell+1)}{2} \frac{s}{\lambda_{\text{el},1}^{(s)}} \right] P_{\ell}(\cos \chi). \quad (4.26)$$

This expression can be evaluated by using the Molière (1948) approximation for the Legendre polynomials, we obtain (see Fernández-Varea *et al.*, 1993b)

$$F^{(s)}(s; \chi) = \frac{1}{2\pi} \left(\frac{\chi}{\sin \chi} \right)^{1/2} \frac{\lambda_{\text{el},1}^{(s)}}{s} \exp \left[\frac{s}{8\lambda_{\text{el},1}^{(s)}} - \frac{\lambda_{\text{el},1}^{(s)}}{2s} \chi^2 \right], \quad (4.27)$$

which does not differ significantly from the Gaussian distribution with variance $s/\lambda_{\text{el},1}^{(s)}$. This result is accurate whenever $s \ll \lambda_{\text{el},1}^{(s)}$ and $\theta_c \ll 1$. It offers a possible method of generating the angular deflection in artificial soft events. When the result given by Eq. (4.27) is applicable, the single parameter $\lambda_{\text{el},1}^{(s)}$ completely determines the multiple-scattering distribution due to soft collisions, *i.e.*, other details of the DCS for scattering angles less than θ_c are irrelevant. However, in actual Monte Carlo simulations, the small-angle approximation is seldom applicable.

In most practical cases the number of hard collisions per electron track can be made relatively large by simply using a small value of the parameter C_1 [see Eq. (4.10)]. When the number of steps is large enough, say larger than ~ 10 , it is not necessary to use the exact distribution $F^{(s)}(s; \chi)$ to sample the angular deflection in artificial soft collisions. Instead, we may use a simpler distribution, $F_a(s; \chi)$, with the same mean and variance, without appreciably distorting the simulation results. This is so because the details of the adopted distribution are washed out after a sufficiently large number of steps and will not be seen in the simulated distributions. Notice that, within the small-angle approximation, it is necessary to keep only the proper value of the first moment to get the correct final distributions. However, if the cutoff angle θ_c is not small enough, the angular distribution $F^{(s)}(s; \chi)$ may become sensitive to higher-order moments of the soft single-scattering distribution. Thus, by also keeping the proper value of the variance, the range of validity of the simulation algorithm is extended, *i.e.*, we can speed up the simulation by using larger values of C_1 (or of $\lambda_{\text{el}}^{(h)}$) and still obtain the correct distributions.

4.1.3 Simulation of soft events

We now return to the notation of Section 3.1, and use the variable $\mu \equiv (1 - \cos \chi)/2$ to describe angular deflections resulting from multiple soft scattering events. The exact first and second moments of the multiple-scattering distribution $F^{(s)}(s; \mu)$ are

$$\langle \mu \rangle^{(s)} \equiv \int_0^1 \mu F_a(s; \mu) d\mu = \frac{1}{2} \left[1 - \exp(-s/\lambda_{\text{el},1}^{(s)}) \right] \quad (4.28)$$

and

$$\langle \mu^2 \rangle^{(s)} \equiv \int_0^1 \mu^2 F_a(s; \mu) d\mu = \langle \mu \rangle^{(s)} - \frac{1}{6} \left[1 - \exp(-s/\lambda_{\text{el},2}^{(s)}) \right]. \quad (4.29)$$

The angular deflection in soft scattering events will be generated from a distribution $F_a(s; \mu)$, which is required to satisfy Eqs. (4.28) and (4.29), but is otherwise arbitrary. PENELOPE uses the following,

$$F_a(s; \mu) = aU_{0,b}(\mu) + (1-a)U_{b,1}(\mu), \quad (4.30)$$

where $U_{u,v}(x)$ denotes the normalised uniform distribution in the interval (u, v) ,

$$U_{u,v}(x) = \begin{cases} 1/(v-u) & \text{if } u \leq x \leq v, \\ 0 & \text{otherwise.} \end{cases} \quad (4.31)$$

The parameters a and b , obtained from the conditions (4.28) and (4.29), are

$$b = \frac{2\langle \mu \rangle^{(s)} - 3\langle \mu^2 \rangle^{(s)}}{1 - 2\langle \mu \rangle^{(s)}}, \quad a = 1 - 2\langle \mu \rangle^{(s)} + b. \quad (4.32)$$

The simple distribution (4.30) is flexible enough to reproduce the combinations of first and second moments encountered in the simulations [notice that $\langle \mu \rangle^{(s)}$, Eq. (4.28), is always less than $1/2$] and allows fast random sampling of μ .

PENELOPE simulates elastic scattering by using either numerical DCSs from the ELSEPA database or the MW DCS model (see Section 3.1). The polar deflection μ in hard events is sampled by using the algorithms described in Section 3.1. These algorithms implement the inverse-transform method (Section 1.2.2), *i.e.*, random values of μ are obtained from the sampling equation [see Eq. (3.24)]

$$\mu = \mathcal{P}_{\text{el}}^{-1}(\xi), \quad (4.33)$$

where

$$\mathcal{P}_{\text{el}}(\mu) = \int_0^\mu p_{\text{el}}(E; \mu) d\mu \quad (4.34)$$

is the cumulative distribution function of μ . To get sampled μ -values greater than the cutoff deflection $\mu_c = (1 - \cos \theta_c)/2$, we only need to use random numbers ξ restricted to the interval $(\xi_c, 1)$ [see Eq. (3.27)].

The angular distribution of soft events $F_a(s; \mu)$, Eq. (4.30), is determined by the first and second transport mean free paths for soft collisions. Evidently, these quantities depend on the cutoff deflection. The mean free path $\lambda_{\text{el}}^{(h)}$ between hard elastic events and the cutoff deflection μ_c are related through [see Eqs. (3.26) and (4.9)]

$$\frac{1}{\lambda_{\text{el}}^{(h)}} = \frac{1}{\lambda_{\text{el}}} \int_{\mu_c}^1 p_{\text{el}}(\mu) d\mu. \quad (4.35)$$

This equation can be inverted to give

$$\mu_c = \mathcal{P}_{\text{el}}^{-1}(\xi_c), \quad (4.36)$$

where

$$\xi_c \equiv 1 - \frac{\lambda_{\text{el}}}{\lambda_{\text{el}}^{(\text{h})}} \quad (4.37)$$

and $\mathcal{P}_{\text{el}}^{-1}$ is the inverse of the single-scattering cumulative distribution function, as defined in Eq. (3.24). Note that Eq. (4.36) has the same form as the sampling equation for μ , Eq. (4.33), and, therefore, the value of μ_c can be obtained by simply invoking the elastic-scattering sampling routine.

Now, we can calculate the first and second transport mean free paths for soft collisions, which are given by

$$\left(\lambda_{\text{el},1}^{(\text{s})}\right)^{-1} = \frac{2}{\lambda_{\text{el}}} T_1(\mu_c) \quad \text{and} \quad \left(\lambda_{\text{el},2}^{(\text{s})}\right)^{-1} = \frac{6}{\lambda_{\text{el}}} [T_1(\mu_c) - T_2(\mu_c)] \quad (4.38)$$

with

$$T_1(\mu_c) = \int_0^{\mu_c} \mu p_{\text{el}}(\mu) d\mu \quad \text{and} \quad T_2(\mu_c) = \int_0^{\mu_c} \mu^2 p_{\text{el}}(\mu) d\mu. \quad (4.39)$$

When numerical DCSs from the ELSEPA database are used, these integrals have to be calculated numerically. With the MW DCS, they can be evaluated analytically. For the case-I MW model, Eq. (3.34), we have

$$\begin{aligned} T_1(\mu_c) &= \int_0^{\mu_c} \mu p_{\text{MW-I}}(\mu) d\mu \\ &= \begin{cases} (1-B)I_1(\mu_c) & \text{if } 0 \leq \xi_c < \xi_0 \\ (1-B)I_1(\mu_0) + (\xi_c - \xi_0)\mu_0 & \text{if } \xi_0 \leq \xi_c < \xi_0 + B \\ (1-B)I_1(\mu_c) + B\mu_0 & \text{if } \xi_0 + B \leq \xi_c \leq 1 \end{cases} \end{aligned} \quad (4.40)$$

and

$$\begin{aligned} T_2(\mu_c) &= \int_0^{\mu_c} \mu^2 p_{\text{MW-I}}(\mu) d\mu \\ &= \begin{cases} (1-B)I_2(\mu_c) & \text{if } 0 \leq \xi_c < \xi_0 \\ (1-B)I_2(\mu_0) + (\xi_c - \xi_0)\mu_0^2 & \text{if } \xi_0 \leq \xi_c < \xi_0 + B \\ (1-B)I_2(\mu_c) + B\mu_0^2 & \text{if } \xi_0 + B \leq \xi_c \leq 1 \end{cases} \end{aligned} \quad (4.41)$$

with

$$I_1(\mu) \equiv A \left[(1+A) \ln \left(\frac{A+\mu}{A} \right) - \frac{(1+A)\mu}{A+\mu} \right] \quad (4.42)$$

and

$$I_2(\mu) \equiv A \left[\frac{(1+A)\mu^2}{A+\mu} - 2I_1(\mu) \right]. \quad (4.43)$$

The quantities ξ_0 and ξ_c are defined by Eqs. (3.43) and (4.37), respectively. The corresponding formulas for the case-II MW model can be derived in a similar way.

4.2 Soft energy losses

The high-energy codes currently available implement different approximate methods to simulate inelastic collisions. Thus, ETRAN and ITS3 make use of the multiple-scattering theories of Landau (1944) and Blunck and Leisegang (1950) to obtain the energy loss distribution due to inelastic collisions after a given path length; the production of secondary electrons is simulated by means of the Møller (1932) and Bhabha (1936) DCSs, which neglect binding effects. This approach accounts for the whole energy straggling, within the accuracy of the multiple-scattering theory, but disregards the correlation between delta ray emission and energy loss in each track segment. As a consequence, energetic delta rays can be generated in a track segment where the energy lost by the primary particle is smaller than the energy of the emitted delta rays. EGS4 uses a mixed procedure to simulate collision energy losses: hard inelastic collisions are simulated from the Møller and Bhabha DCSs, thus neglecting binding effects, and soft inelastic collisions are described by means of the continuous slowing down approximation (CSDA), *i.e.*, energy straggling due to soft inelastic collisions is ignored. As regards bremsstrahlung emission, EGS4 implements a mixed procedure in which hard radiative events are simulated in detail and use is made of the CSDA to simulate the effect of soft photon emission; ETRAN uses strictly detailed simulation.

To make the arguments more precise, we introduce the cutoff values W_{cc} and W_{cr} , and consider inelastic collisions with energy loss $W < W_{cc}$ and emission of bremsstrahlung photons with $W < W_{cr}$ as soft stopping interactions. The use of the CSDA to describe soft interactions is well justified when the energy straggling due to these interactions is negligible, as happens when the cutoff energies W_{cc} and W_{cr} are both small, so that the fraction of the stopping power due to soft interactions is also small. To improve the description of energy straggling one should reduce the cutoff energies, but this enlarges the number of hard inelastic and radiative events to be simulated along each track and hence the simulation time. Our purpose is to go beyond the CSDA by introducing energy straggling in the description of soft stopping interactions. It is clear that, by proceeding in this way, we will be able to use larger values of the cutoff energies W_{cc} and W_{cr} , and hence speed up the simulation, without distorting the energy distributions.

In previous versions of PENELOPE, soft energy losses were simulated by using the mixed simulation algorithm described by Baró *et al.* (1995). The quantities that define the algorithm are the mean free paths $\lambda_{in}^{(h)}$ and $\lambda_{br}^{(h)}$ between hard collisions and hard radiative events, the stopping power S_s and the energy straggling parameter Ω_s^2 associated with soft interactions. These quantities are given by

$$\lambda_{in}^{(h)}(E) = \left(\mathcal{N} \int_{W_{cc}}^E \frac{d\sigma_{in}}{dW} dW \right)^{-1}, \quad (4.44)$$

$$\lambda_{br}^{(h)}(E) = \left(\mathcal{N} \int_{W_{cr}}^E \frac{d\sigma_{br}}{dW} dW \right)^{-1}, \quad (4.45)$$

$$S_s(E) = \mathcal{N} \int_0^{W_{cc}} W \frac{d\sigma_{in}}{dW} dW + \mathcal{N} \int_0^{W_{cr}} W \frac{d\sigma_{br}}{dW} dW \quad (4.46)$$

and

$$\Omega_s^2(E) = \mathcal{N} \int_0^{W_{cc}} W^2 \frac{d\sigma_{in}}{dW} dW + \mathcal{N} \int_0^{W_{cr}} W^2 \frac{d\sigma_{br}}{dW} dW. \quad (4.47)$$

To prevent $\lambda_{br}^{(h)}(E)$ from vanishing (infrared divergence), in PENELOPE the radiative cutoff energy W_{cr} is required to be larger than or equal to 10 eV.

Let us consider that a particle, electron or positron, travels a step of length s between two consecutive hard events of any kind (*i.e.*, hard elastic or inelastic collisions, hard bremsstrahlung emissions, and annihilation in the case of positrons). Along this step, the particle is assumed to interact only through soft inelastic collisions and soft bremsstrahlung emission. We consider that the average energy loss in this path length, $S_s(E)s$, is much less than the initial energy E so that the DCSs can be assumed to stay essentially constant along the step. Let $G(s; \omega)$ denote the PDF of the energy loss ω along the path length s ; this distribution satisfies the transport equation (Landau, 1944)

$$\frac{\partial G(s; \omega)}{\partial s} = \mathcal{N} \int_0^\infty [G(s; \omega - W) - G(s; \omega)] \sigma_s(E; W) dW \quad (4.48)$$

with the initial value $G(0; \omega) = \delta(\omega)$. Here, $\sigma_s(E; W)$ stands for the DCS for soft stopping interactions, *i.e.*,

$$\sigma_s(E; W) \equiv \frac{d\sigma_s}{dW} = \frac{d\sigma_{in}}{dW} \Theta(W_{cc} - W) + \frac{d\sigma_{br}}{dW} \Theta(W_{cr} - W), \quad (4.49)$$

where $\Theta(x)$ is the step function. A closed formal solution of the integral equation (4.48) may be obtained by considering its Fourier, or Laplace, transform with respect to ω (see, *e.g.*, Landau, 1944, Blunck and Leisegang, 1950). For our purposes it is only necessary to know the first moments of the energy loss distribution after the path length s ,

$$\langle \omega^n \rangle \equiv \int_0^\infty \omega^n G(s; \omega) d\omega. \quad (4.50)$$

From Eq. (4.48) it follows that

$$\begin{aligned} \frac{d}{ds} \langle \omega^n \rangle &= \mathcal{N} \int_0^\infty d\omega \int_0^\infty dW \omega^n [G(s; \omega - W) - G(s; \omega)] \sigma_s(E; W) \\ &= \mathcal{N} \left(\int_0^\infty d\omega' \int_0^\infty dW (\omega' + W)^n G(s; \omega') \sigma_s(E; W) - \langle \omega^n \rangle \int_0^\infty \sigma_s(E; W) dW \right) \\ &= \sum_{k=1}^n \frac{n!}{k!(n-k)!} \langle \omega^{n-k} \rangle \mathcal{N} \int_0^\infty W^k \sigma_s(E; W) dW, \end{aligned} \quad (4.51)$$

where use has been made of the fact that $\sigma_s(E; W)$ vanishes when $W < 0$. In particular, we have

$$\frac{d}{ds} \langle \omega \rangle = \mathcal{N} \int_0^\infty W \sigma_s(E; W) dW = S_s, \quad (4.52)$$

$$\begin{aligned}\frac{d}{ds}\langle\omega^2\rangle &= 2\langle\omega\rangle\mathcal{N}\int_0^\infty W\sigma_s(E;W)dW + \mathcal{N}\int_0^\infty W^2\sigma_s(E;W)dW \\ &= 2\langle\omega\rangle S_s + \Omega_s^2\end{aligned}\quad (4.53)$$

and, hence,

$$\langle\omega\rangle = S_s s, \quad (4.54)$$

$$\langle\omega^2\rangle = (S_s s)^2 + \Omega_s^2 s. \quad (4.55)$$

The variance of the energy loss distribution is

$$\text{var}(\omega) = \langle\omega^2\rangle - \langle\omega\rangle^2 = \Omega_s^2 s, \quad (4.56)$$

i.e., the energy straggling parameter Ω_s^2 equals the variance increase per unit path length.

The key point in our argument is that soft interactions involve only comparatively small energy losses. If the number of soft interactions along the path length s is statistically sufficient, it follows from the central limit theorem that the energy loss distribution is Gaussian with mean $S_s s$ and variance $\Omega_s^2 s$,

$$G(s; \omega) \simeq \frac{1}{(2\pi\Omega_s^2(E)s)^{1/2}} \exp\left[-\frac{(\omega - S_s(E)s)^2}{2\Omega_s^2(E)s}\right]. \quad (4.57)$$

This result is accurate only if 1) the average energy loss $S_s(E)s$ is much smaller than E (so that the DCS $d\sigma_s/dW$ is nearly constant along the step) and 2) its standard deviation $[\Omega_s^2(E)s]^{1/2}$ is much smaller than its mean $S_s(E)s$ (otherwise there would be a finite probability of negative energy losses), *i.e.*,

$$[\Omega_s^2(E)s]^{1/2} \ll S_s(E)s \ll E. \quad (4.58)$$

Requirement 1) implies that the cutoff energies W_{cc} and W_{cr} for delta ray production and photon emission have to be relatively small. The second requirement holds for path lengths larger than $s_{\text{crit}} = \Omega_s^2/S_s^2$.

Now, we address ourselves to the problem of simulating the energy losses due to soft stopping interactions between two consecutive hard events. The distribution (4.57) gives the desired result when conditions (4.58) are satisfied. In fact, the use of a Gaussian distribution to simulate the effect of soft stopping interactions was previously proposed by Andreo and Brahme (1984). Unfortunately, the step lengths found in our simulations are frequently too short for conditions (4.58) to hold (*i.e.*, s is usually less than s_{crit}). To get over this problem, we replace the actual energy loss distribution $G(s; \omega)$ by a simpler “equivalent” distribution $G_a(s; \omega)$ with the same mean and variance, given by Eqs. (4.54) and (4.56). Other details of the adopted distribution have no effect on the simulation results, provided that the number of steps along each track is statistically sufficient (say, larger than ~ 20). PENELOPE generates ω from the following distributions

- Case I. If $\langle\omega\rangle^2 > 9 \text{var}(\omega)$, we use a truncated Gaussian distribution (normalisation is irrelevant here),

$$G_{\text{a,I}}(s; \omega) = \begin{cases} \exp \left[-\frac{(\omega - \langle\omega\rangle)^2}{2(1.015387\sigma)^2} \right] & \text{if } |\omega - \langle\omega\rangle| < 3\sigma, \\ 0 & \text{otherwise,} \end{cases} \quad (4.59)$$

where $\sigma = [\text{var}(\omega)]^{1/2}$ is the standard deviation and the numerical factor 1.015387 corrects for the effect of the truncation. Notice that the shape of this distribution is very similar to that of the “true” energy-loss distribution, Eq. (4.57). Random sampling from the distribution (4.59) is performed by means of the RITA method (Section 1.2.4) with Walker’s aliasing.

- Case II. When $3 \text{var}(\omega) < \langle\omega\rangle^2 < 9 \text{var}(\omega)$, the energy loss is sampled from the uniform distribution

$$G_{\text{a,II}}(s; \omega) = U_{\omega_1, \omega_2}(\omega) \quad (4.60)$$

with

$$\omega_1 = \langle\omega\rangle - \sqrt{3}\sigma, \quad \omega_2 = \langle\omega\rangle + \sqrt{3}\sigma. \quad (4.61)$$

- Case III. Finally, when $\langle\omega\rangle^2 < 3 \text{var}(\omega)$, the adopted distribution is an admixture of a delta and a uniform distribution,

$$G_{\text{a,III}}(s; \omega) = a\delta(\omega) + (1 - a)U_{0, \omega_0}(\omega) \quad (4.62)$$

with

$$a = \frac{3\text{var}(\omega) - \langle\omega\rangle^2}{3\text{var}(\omega) + 3\langle\omega\rangle^2} \quad \text{and} \quad \omega_0 = \frac{3\text{var}(\omega) + 3\langle\omega\rangle^2}{2\langle\omega\rangle}. \quad (4.63)$$

It can be easily verified that these distributions have the required mean and variance. It is also worth noticing that they yield ω values that are less than

$$\omega_{\text{max}} = \begin{cases} \langle\omega\rangle + 3\sigma & \text{in case I,} \\ \omega_2 & \text{in case II,} \\ \omega_0 & \text{in case III.} \end{cases} \quad (4.64)$$

ω_{max} is normally much less than the kinetic energy E of the transported particle. Energy losses larger than E might be generated only when the step length s has a value of the order of the Bethe range, but this never happens in practical simulation (see below). It is worth noticing that, after a moderately large number of steps, this simple simulation scheme effectively yields an energy loss distribution that has the correct first and second moments and is similar in shape to the “true” distribution. Further improvements of the distribution of soft energy losses would mean considering higher order moments of the single scattering inelastic DCS given by Eq. (4.49).

In spatial-dose calculations, the energy loss ω due to soft stopping interactions can be considered to be locally deposited at a random position uniformly distributed along the step. This procedure yields dose distributions identical to those obtained by assuming

that the energy loss is deposited at a constant rate along the step, but is computationally simpler. According to this, PENELOPE simulates the combined effect of all soft elastic collisions and soft stopping interactions that occur between a pair of successive hard events, separated a distance s , as a single event (a hinge) in which the particle changes its direction of movement according to the distribution $F_a(s; \mu)$, Eqs. (4.30)–(4.32), and loses energy ω that is generated from the distribution $G_a(s; \omega)$, Eqs. (4.59)–(4.63). The position of the hinge is sampled uniformly along the step, as in the case of purely elastic scattering (Section 4.1.2). When the step crosses an interface (see Fig. 4.2), the artificial event is simulated only when its position lies in the initial material; otherwise the track is stopped at the interface and restarted in the new material. It can be easily verified that the particle reaches the interface not only with the correct average direction of movement, but also with the correct average energy, $E - S_s t$.

4.2.1 Energy dependence of the soft DCS

The simulation model for soft energy losses described above is based on the assumption that the associated energy-loss DCS does not vary with the energy of the transported particle. To account for the energy dependence of the DCS in a rigorous way, we have to start from the transport equation [cf. Eq. (4.48)]

$$\begin{aligned} \frac{\partial G(s; \omega)}{\partial s} = & \mathcal{N} \int_0^\infty G(s; \omega - W) \sigma_s(E_0 - \omega + W; W) dW \\ & - \mathcal{N} \int_0^\infty G(s; \omega) \sigma_s(E_0 - \omega; W) dW, \end{aligned} \quad (4.65)$$

where E_0 denotes the kinetic energy of the particle at the beginning of the step. We desire to obtain expressions for the first and second moments, $\langle \omega \rangle$ and $\langle \omega^2 \rangle$, of the multiple-scattering energy-loss distribution, which define the artificial distribution $G_a(s; \omega)$ as described above. Unfortunately, for a realistic DCS, these moments can only be obtained after arduous numerical calculations and we have to rely on simple approximations that can be easily implemented in the simulation code.

Let us consider that, at least for relatively small fractional energy losses, the DCS varies linearly with the kinetic energy of the particle,

$$\sigma_s(E_0 - \omega; W) \simeq \sigma_s(E_0; W) - \left[\frac{\partial \sigma_s(E; W)}{\partial E} \right]_{E=E_0} \omega. \quad (4.66)$$

We recall that we are considering only soft energy-loss interactions (inelastic collisions and bremsstrahlung emission) for which the cutoff energies, W_{cc} and W_{cr} , do not vary with E . Therefore, the upper limit of the integrals in the right hand side of Eq. (4.65) is finite and independent of the energy of the particle. The stopping power $S_s(E_0 - \omega)$ can then be approximated as

$$S_s(E_0 - \omega) \equiv \mathcal{N} \int W \sigma_s(E_0 - \omega; W) dW \simeq S_s(E_0) - S'_s(E_0) \omega, \quad (4.67)$$

where the prime denotes the derivative with respect to E . Similarly, for the straggling parameter $\Omega_s^2(E)$ we have

$$\Omega_s^2(E_0 - \omega) \equiv \mathcal{N} \int W^2 \sigma_s(E_0 - \omega; W) dW \simeq \Omega_s^2(E_0) - \Omega_s^{2'}(E_0) \omega. \quad (4.68)$$

From Eq. (4.65) it follows that the moments of the multiple-scattering distribution,

$$\langle \omega^n \rangle = \int \omega^n G(s; \omega) d\omega,$$

satisfy the equations

$$\begin{aligned} \frac{d}{ds} \langle \omega^n \rangle &= \mathcal{N} \int d\omega \int dW [(\omega + W)^n G(s; \omega) \sigma_s(E_0 - \omega; W)] \\ &\quad - \mathcal{N} \int d\omega \int dW \omega^n G(s; \omega) \sigma_s(E_0 - \omega; W) \\ &= \mathcal{N} \sum_{k=1}^n \frac{n!}{k!(n-k)!} \int d\omega \int dW \omega^{n-k} W^k G(s; \omega) \sigma_s(E_0 - \omega; W). \end{aligned} \quad (4.69)$$

By inserting the approximation (4.66), we obtain

$$\frac{d}{ds} \langle \omega^n \rangle = \sum_{k=1}^n \frac{n!}{k!(n-k)!} (\langle \omega^{n-k} \rangle M_k - \langle \omega^{n-k+1} \rangle M'_k), \quad (4.70)$$

where

$$M_k \equiv \mathcal{N} \int W^k \sigma_s(E_0; W) dW \quad (4.71)$$

and

$$M'_k \equiv \mathcal{N} \int W^k \left[\frac{\partial \sigma_s(E; W)}{\partial E} \right]_{E=E_0} dW = \left[\frac{dM_k}{dE} \right]_{E=E_0}. \quad (4.72)$$

The equations (4.70) with the boundary conditions $\langle \omega^n \rangle_{s=0} = 0$ can now be solved sequentially to any order. For $n = 1$ we have

$$\frac{d}{ds} \langle \omega \rangle = S_s(E_0) - S'_s(E_0) \langle \omega \rangle, \quad (4.73)$$

which yields

$$\langle \omega \rangle = \frac{S_s(E_0)}{S'_s(E_0)} \left\{ 1 - \exp[-S'_s(E_0)s] \right\}. \quad (4.74)$$

The equation for $n = 2$ reads,

$$\frac{d}{ds} \langle \omega^2 \rangle = \Omega_s^2(E_0) + \left[2S_s(E_0) - \Omega_s^{2'}(E_0) \right] \langle \omega \rangle - 2S'_s(E_0) \langle \omega^2 \rangle, \quad (4.75)$$

and its solution is

$$\begin{aligned} \langle \omega^2 \rangle &= \Omega_s^2(E_0) \frac{1 - \exp[-2S'_s(E_0)s]}{2S'_s(E_0)} \\ &+ s \left[2S_s(E_0) - \Omega_s^{2'}(E_0) \right] S_s(E_0) \left[\frac{1 - \exp[-S'_s(E_0)s]}{2S'_s(E_0)} \right]^2. \end{aligned} \quad (4.76)$$

Hence,

$$\begin{aligned} \text{var}(\omega) &= \langle \omega^2 \rangle - \langle \omega \rangle^2 \\ &= \Omega_s^2(E_0) \frac{1 - \exp[-2S'_s(E_0)s]}{2S'_s(E_0)} - 2\Omega_s^{2'}(E_0)S_s(E_0) \left[\frac{1 - \exp[-S'_s(E_0)s]}{2S'_s(E_0)} \right]^2. \end{aligned} \quad (4.77)$$

Since these expressions are derived from the linear approximation, Eq. (4.66), it is consistent to evaluate $\langle \omega \rangle$ and $\text{var}(\omega)$ from their Taylor expansions to second order,

$$\begin{aligned} \langle \omega \rangle &= S_s(E_0) s \left[1 - \frac{1}{2} S'_s(E_0) s + \mathcal{O}(s^2) \right] \\ &\simeq S_s(E_0) s \left\{ 1 - \frac{1}{2} \left[\frac{d \ln S_s(E)}{dE} \right]_{E=E_0} S_s(E_0) s \right\} \end{aligned} \quad (4.78)$$

and

$$\begin{aligned} \text{var}(\omega) &= \Omega_s^2(E_0) s - \left[\frac{1}{2} \Omega_s^{2'}(E_0) S_s(E_0) + \Omega_s^2(E_0) S'_s(E_0) \right] s^2 + \mathcal{O}(s^3) \\ &\simeq \Omega_s^2(0) s \left\{ 1 - \left[\frac{1}{2} \frac{d \ln \Omega_s^2(E)}{dE} + \frac{d \ln S_s(E)}{dE} \right]_{E=E_0} S_s(E_0) s \right\}, \end{aligned} \quad (4.79)$$

where the logarithmic derivatives have been introduced for numerical convenience. The factors in curly brackets account for the global effect of the energy dependence of the soft energy-loss DCS (within the linear approximation). To simulate soft energy losses, we sample ω from the artificial distribution $G_a(\omega; s)$, Eqs. (4.59) to (4.63), with the “correct” first moment and variance, given by expressions (4.78) and (4.79). In PENELOPE, we use step lengths s such that the fractional energy loss along each step is relatively small (see below) and, consequently, the energy-dependence correction is also small (*i.e.*, the correcting factors are close to unity).

4.3 Combined scattering and energy loss

Up to this point, soft scattering and energy loss have been regarded as essentially independent processes, while in reality they coexist. In this Section, we consider their interplay and set the basis of an algorithm that simulates their combined effect.

Ours is a mixed algorithm, where hard interactions are described individually from the associated DCSs (see Chapter 3). These interactions are 1) hard elastic collisions, “el”, 2) hard inelastic collisions, “in”, 3) hard bremsstrahlung photon emission, “br”, 4) ionisation of inner shells, “si”, and, in the case of positrons, 5) positron annihilation, “an”. The mean free path between consecutive hard events, $\lambda_T^{(h)}$, is given by

$$\left[\lambda_T^{(h)}\right]^{-1} = \mathcal{N}\sigma_T^{(h)} = \mathcal{N}\left[\sigma_{\text{el}}^{(h)} + \sigma_{\text{in}}^{(h)} + \sigma_{\text{br}}^{(h)} + \sigma_{\text{si}} (+\sigma_{\text{an}})\right] \equiv \Sigma_h, \quad (4.80)$$

where $\sigma_T^{(h)}$ is the total atomic cross section for hard interactions. We recall that the inverse mean free path, Σ_h , gives the interaction probability per unit path length. In the absence of soft energy-loss events, the PDF of the step length s between two successive hard events (or from a given point in the track to the next hard event) is

$$p(s) = \Sigma_h \exp(-\Sigma_h s). \quad (4.81)$$

In each hard event, one and only one interaction (i=“el”, “in”, “br”, “si” or “an”) occurs with probability

$$p_i = \sigma_i^{(h)} / \sigma_T^{(h)}. \quad (4.82)$$

When soft energy-losses are considered, the PDF of the distance s travelled by the particle to the following hard interaction is not given by Eq. (4.81), because the mean free path $\lambda_T^{(h)}$ varies with energy and may change appreciably along a single step. The simplest way to cope with this problem is to limit the length of the step to make sure that the *average* energy loss is much smaller than the kinetic energy E at the beginning of the step, and consider that $\lambda_T^{(h)}(E)$ remains essentially constant along the step. Then, the mean energy loss in a step is given by

$$\langle \Delta E \rangle = \lambda_T^{(h)} S(E), \quad (4.83)$$

where

$$S(E) = S_{\text{in}}(E) + S_{\text{br}}(E) \quad (4.84)$$

is the total stopping power. Since the mean free path between consecutive hard events of any kind is shorter than the mean free path between hard elastic events, the energy loss per step can be limited by re-defining the hard mean free path. If we wish to tolerate average fractional energy losses $\Delta E/E$ along a step of the order of C_2 (a small value, say, 0.05), we simply take

$$\lambda_{\text{el}}^{(h)}(E) = \max \left\{ \lambda_{\text{el}}(E), \min \left[C_1 \lambda_{\text{el},1}(E), C_2 \frac{E}{S(E)} \right] \right\}. \quad (4.85)$$

This effectively limits the average energy loss per step at the expense of increasing the frequency of hard elastic events. The parameters C_1 and C_2 in Eq. (4.85), to be selected by the user, determine the computer time needed to simulate each track. Ideally, they should not have any influence on the *accuracy* of the simulation results. This happens only when their values are sufficiently small (see below).

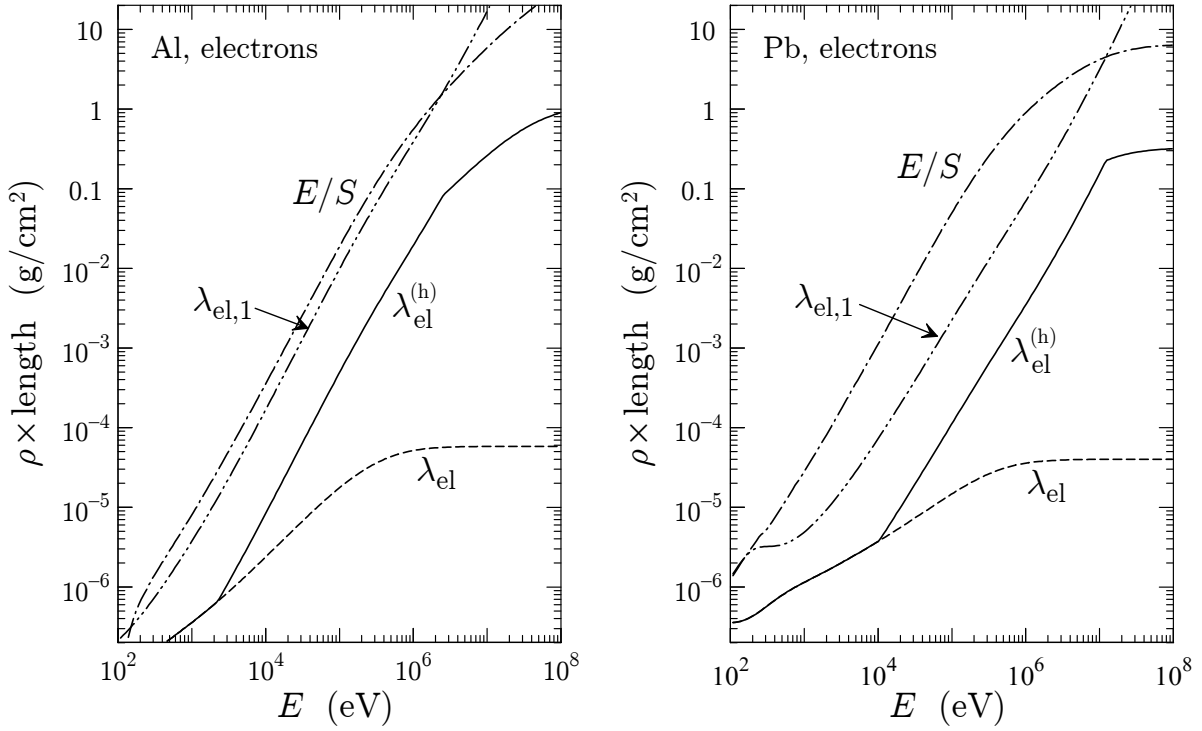


Figure 4.3: Elastic mean free path λ_{el} , first transport mean free path $\lambda_{\text{el},1}$ and $E/S(E)$ for electrons in aluminium and lead. The solid line represents the mean free path between hard elastic events $\lambda_{\text{el}}^{(\text{h})}$ obtained from Eq. (4.85) with $C_1 = C_2 = 0.05$.

It should be noted that C_1 and C_2 act on different energy domains. This is illustrated in Fig. 4.3, where the lengths λ_{el} , $\lambda_{\text{el},1}$ and E/S for electrons in aluminium and lead are represented as functions of the kinetic energy. The mean free path $\lambda_{\text{el}}^{(\text{h})}$ for hard elastic events, determined from the prescription (4.85) with $C_1 = C_2 = 0.05$ is also plotted. For low energies, $\lambda_{\text{el}}^{(\text{h})} = \lambda_{\text{el}}$ and the simulation is purely detailed ($\mu_c = 0$). For intermediate energies, $\lambda_{\text{el}}^{(\text{h})} = C_1 \lambda_{\text{el},1}$, whereas $\lambda_{\text{el}}^{(\text{h})} = C_2 E/S(E)$ in the high-energy domain. From Fig. 4.3 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.

4.3.1 Variation of $\lambda_{\text{T}}^{(\text{h})}$ with energy

With the definition (4.85) of the hard elastic mean free path, we only set a limit on the *average* step length. However, since s is sampled from the exponential distribution, its realisations fluctuate amply about the average value. On the other hand, the soft energy loss ω along a step of given length s also fluctuates about the mean value $\langle \omega \rangle$ given by Eq. (4.78). This means that the inverse mean free path $\Sigma_{\text{h}}(E)$ varies along the step in an essentially unpredictable way.

Let us consider for a moment that the CSDA is applicable (*i.e.*, that the effect of soft energy straggling is negligible). In this case, there is a one-by-one correspondence between the kinetic energy E of the electron and the travelled path length s ,

$$s = \int_E^{E_0} \frac{dE'}{S_s(E')}, \quad (4.86)$$

where E_0 is the initial energy (at $s = 0$) and $S_s(E)$ is the soft stopping power, Eq. (4.46) [we consider that no hard interactions occur along the step]. Equivalently,

$$\frac{ds}{dE} = -\frac{1}{S_s(E)}. \quad (4.87)$$

Thus, the inverse mean free path Σ_h can be formally considered as a function of the path length s . The probability $p(s) ds$ of having the first hard interaction when the particle has travelled a length in the interval $(s, s + ds)$ is determined by the equation [cf. Eq. (1.109)]

$$p(s) = \Sigma_h(s) \int_s^\infty p(s') ds', \quad (4.88)$$

with the normalisation condition,

$$\int_0^\infty p(s) ds = 1. \quad (4.89)$$

Instead of the path length s , it is convenient to consider the dimensionless variable

$$q \equiv \int_E^{E_0} \frac{\Sigma_h(E')}{S_s(E')} dE' = \int_0^s \Sigma_h(s') ds', \quad (4.90)$$

which varies with energy and

$$\frac{dq}{dE} = -\frac{\Sigma_h(E)}{S_s(E)}. \quad (4.91)$$

The PDF of q is

$$\pi(q) = p(s) \frac{ds}{dq} = p(s) \frac{ds}{dE} \frac{dE}{dq} = p(s) \frac{1}{\Sigma_h(s)}. \quad (4.92)$$

From Eq. (4.88) it follows that $\pi(q)$ satisfies the equation

$$\pi(q) = \int_q^\infty \pi(q') dq'. \quad (4.93)$$

Therefore, q is distributed exponentially,

$$\pi(q) = \exp(-q). \quad (4.94)$$

The PDF of the step length s is obtained by inverting the transformation (4.90),

$$p(s) = \Sigma_h(s) \exp\left(-\int_0^s \Sigma_h(s') ds'\right). \quad (4.95)$$

It is not practical to sample s from this complicated PDF. It is much more convenient to sample q [as $-\ln \xi$, cf. Eq. (1.36)] and then determine s from (4.90), which can be inverted numerically (for practical details, see Berger, 1998). Although this sampling method effectively accounts for the energy dependence of $\Sigma_s(E)$, it is applicable only to simulations in the CSDA.

A more versatile algorithm for sampling the position of hard events, still within the CSDA, is the following. We let the electron move in steps of maximum length $s_{\max}$, a value specified by the user. This determines the maximum energy loss along the step,

$$\omega_{\max} = \int_0^{s_{\max}} S_s(s) ds. \quad (4.96)$$

Let $\Sigma_{h,\max}$ denote an upper bound for the inverse mean free path of hard events in the swept energy interval, *i.e.*,

$$\Sigma_{h,\max} > \max \{ \Sigma_h(E), E \in (E_0 - \omega_{\max}, E_0) \} \quad (4.97)$$

We now assume that the electron may undergo fictitious events in which the energy and direction remain unaltered (delta interactions). The inverse mean free path of these interactions is defined as

$$\Sigma_\delta(E) = \Sigma_{h,\max} - \Sigma_h(E), \quad (4.98)$$

so that the inverse mean free path of the combined process (delta interactions + hard events) equals $\Sigma_{h,\max}$, a constant. Owing to the Markovian character of the processes, the introduction of delta interactions does not influence the path-length distribution between hard events. Therefore, the occurrence of hard events can be sampled by means of the following simple algorithm,

- (i) Sample a distance s from the exponential distribution with inverse mean free path $\Sigma_{h,\max}$, *i.e.*, $s = (-\ln \xi)/\Sigma_{h,\max}$.
- (ii) If $s > s_{\max}$, move the electron a path length $s_{\max}$ and determine the soft energy loss ω along this path length. Modify the electron energy², $E \leftarrow E - \omega$, and assume that a delta interaction occurs at the end of the step.
- (iii) If $s < s_{\max}$, move the electron a step of length s . Determine the energy loss ω and update the energy, $E \leftarrow E - \omega$. Sample a random number ξ .
 - (1) If $\xi \Sigma_{h,\max} < \Sigma_h(E)$, simulate a hard interaction
 - (2) Otherwise, assume that the particle undergoes a delta interaction.
- (i) Return to (i).

It is clear that the path-length s to the first hard interaction generated with this algorithm follows the PDF (4.95). The interesting peculiarity of this algorithm is that it makes no explicit reference to the CSDA. Therefore, it can be adopted in mixed simulations with soft-energy-loss straggling, provided only that an upper bound exists for the energy ω lost along the path length $s_{\max}$.

²In the description of the algorithms we use the symbol $\leftarrow$ in expressions such as " $a \leftarrow b$ " to indicate that the value b replaces the value of a .

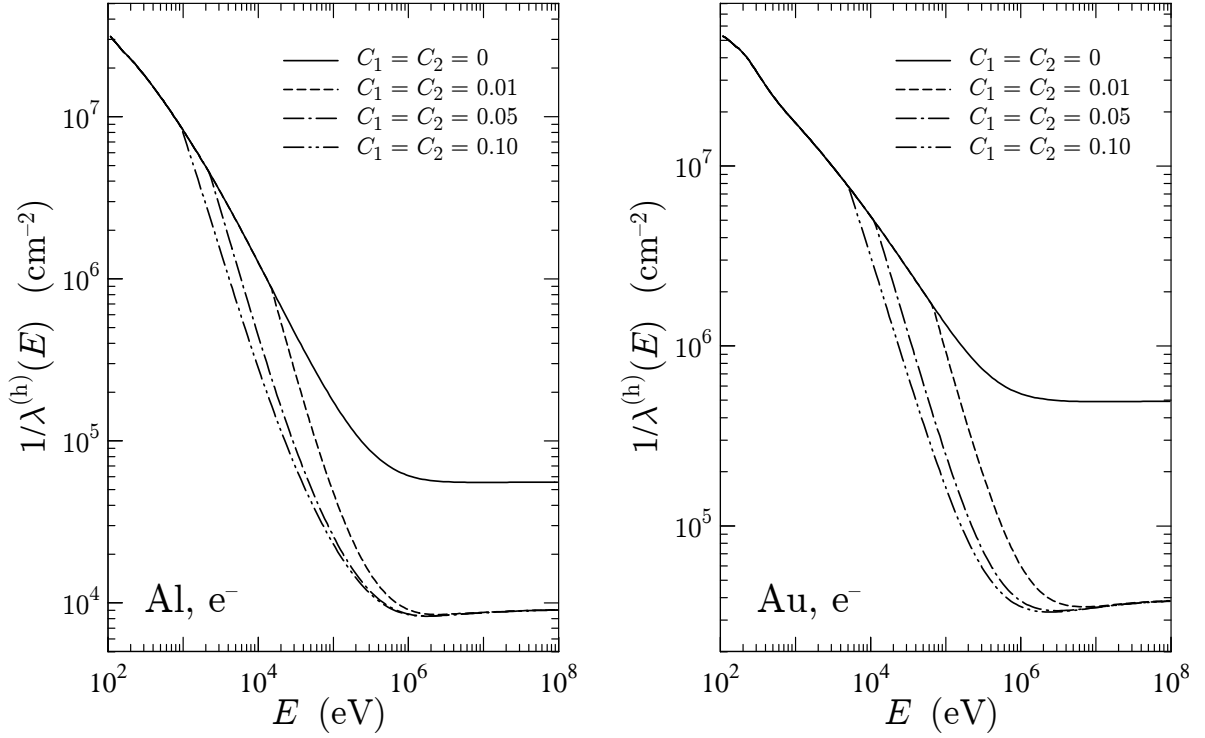


Figure 4.4: Inverse mean free path (interaction probability per unit path length) for hard interactions of electrons in aluminium and gold for the indicated values of the simulation parameters. The plotted curves were calculated with $W_{cc} = W_{cr} = 100$ eV. Note that, by varying the values of C_1 and C_2 , the inverse mean free path cannot be made smaller than the contributions from hard inelastic and radiative events.

Fortunately, the energy loss generated from the artificial distribution $G_a(\omega; s)$, Eqs. (4.59)–(4.63), is always less than $\omega_{\max}$, Eq. (4.64). Indeed, in case I we use the truncated Gaussian distribution (4.59) just to enforce this property. In our mixed simulation we shall select a maximum step length $s_{\max}$, which serves to set an upper bound for the energy that the transported electrons may lose along each step. Since the hard inverse mean free path $\Sigma_h(E)$ has a broad minimum (and no local maxima) in the whole energy interval of interest (see Fig. 4.4), the maximum value of Σ_h within a certain energy interval (E_1, E_2) occurs at one of the end points. This makes the practical implementation of the above algorithm very easy.

4.3.2 Scattering by atomic electrons

Most of the existing high-energy simulation codes have difficulties in accounting for the angular deflections of the projectile due to inelastic collisions (see, *e.g.*, Jenkins *et al.*, 1988). The inelastic cross section differential in the scattering angle can be calculated approximately in terms of the incoherent scattering function (see, *e.g.*, Mott and Massey,

1965). This was the approach followed by Fano (1954) in order to introduce electron scattering effects in the Molière (1948) multiple-scattering theory. However, the DCS calculated in this way accounts for all excitations and, hence, it is not adequate for mixed simulations, where the part of electron scattering due to hard collisions is explicitly simulated. Moreover, the calculation of the DCS from the incoherent scattering function involves an average over excitation energies that cannot be performed exactly; instead an effective “minimum momentum transfer” is introduced, which must be estimated empirically. This may cause inconsistencies for low-energy projectiles. A more consistent approach (Baró *et al.*, 1995) is obtained by simply computing the restricted angular DCS, for soft collisions with $W < W_{cc}$, from our inelastic scattering model (see Section 3.2), as follows.

We recall that the recoil energy Q is given by (see Appendix B)

$$Q(Q + 2m_e c^2) = c^2(p^2 + p'^2 - 2pp' \cos \theta), \quad (4.99)$$

where p and p' are the magnitudes of the momentum of the projectile before and after the collision,

$$(cp)^2 = E(E + 2m_e c^2) \quad \text{and} \quad (cp')^2 = (E - W)(E - W + 2m_e c^2). \quad (4.100)$$

In soft distant interactions, the angular deflection $\mu = (1 - \cos \theta)/2$ and the recoil energy Q are related through

$$Q(Q + 2m_e c^2) = 4cp \, cp_k \mu + (cp - cp_k)^2, \quad (4.101)$$

where p_k is the momentum of the projectile after the collision,

$$(cp_k)^2 = (E - W_k)(E - W_k + 2m_e c^2). \quad (4.102)$$

The cross section for soft distant interactions³, Eq. (3.67), can then be expressed in terms of the variable μ as

$$\frac{d\sigma_{\text{dis},l}}{d\mu} = \frac{4\pi e^4}{m_e v^2} \sum_k f_k \frac{1}{W_k} \frac{m_e c^2}{4cp \, cp_k \mu + (cp - cp_k)^2} \frac{4cp \, cp_k}{2(Q + m_e c^2)}. \quad (4.103)$$

Considering that $Q \ll m_e c^2$ for the majority of soft distant collisions, we have

$$\frac{d\sigma_{\text{dis},l}}{d\mu} = \frac{2\pi e^4}{m_e v^2} \sum_k f_k \frac{1}{W_k} \frac{1}{R_k + \mu}, \quad 0 < \mu < \mu_k, \quad (4.104)$$

where

$$R_k = \frac{(cp - cp_k)^2}{4cp \, cp_k} \quad (4.105)$$

and

$$\mu_k = \mu(Q = W_k) = \frac{W_k(W_k + 2m_e c^2) - (cp - cp_k)^2}{4cp \, cp_k}. \quad (4.106)$$

³Distant transverse interactions do not cause scattering.

On the other hand, the DCS per unit oscillator strength for soft ($W < W_{cc}$) close collisions with the i -th oscillator is given by [see Eqs. (3.78) and (3.84)]

$$\frac{d\sigma_{clo}^{(\pm)}}{dW} = \frac{2\pi e^4}{m_e v^2} \sum_k f_k \frac{1}{W^2} F^{(\pm)}(E, W). \quad (4.107)$$

The angular deflection and the energy loss are related by (3.125), which implies that

$$W = \frac{E(E + 2m_e c^2)2(\mu - \mu^2)}{2E(\mu - \mu^2) + m_e c^2} \quad (4.108)$$

and

$$\frac{dW}{d\mu} = \frac{E(E + 2m_e c^2)m_e c^2 2(1 - 2\mu)}{[2E(\mu - \mu^2) + m_e c^2]^2}. \quad (4.109)$$

Therefore,

$$\frac{d\sigma_{clo}^{(\pm)}}{d\mu} = \frac{2\pi e^4}{m_e v^2} \sum_k f_k \frac{1}{W^2} F^{(\pm)}(E, W) \frac{dW}{d\mu}, \quad \mu_k < \mu < \mu_{cc}, \quad (4.110)$$

where

$$\mu_{cc} = \mu(Q = W_{cc}) = \frac{W_{cc}(W_{cc} + 2m_e c^2) - (cp - cp_{cc})^2}{4cp \, cp_{cc}} \quad (4.111)$$

with

$$(cp_{cc})^2 = (E - W_{cc})(E - W_{cc} + 2m_e c^2). \quad (4.112)$$

The angular DCS for soft inelastic interactions is then given by

$$\begin{aligned} \frac{d\sigma_s}{d\mu} &= \frac{d\sigma_{dis,l}}{d\mu} + \frac{d\sigma_{clo}^{(\pm)}}{d\mu} \\ &= \frac{2\pi e^4}{m_e v^2} \sum_k f_k \left\{ \frac{1}{W_k} \frac{1}{R_k + \mu} + \frac{1}{W^2} F^{(\pm)}(E, W) \frac{dW}{d\mu} \right\}, \end{aligned} \quad (4.113)$$

where the summations extend over the oscillators with resonance energy less than W_{cc} and greater than W_{max} , and each term contributes only for the μ -intervals indicated above. The mean free path and the first and second transport mean free paths for soft inelastic scattering are

$$\left[\lambda_{in}^{(s)} \right]^{-1} = \mathcal{N} \int_0^{\mu_2} \frac{d\sigma_{in}^{(s)}}{d\mu} d\mu, \quad (4.114)$$

$$\left[\lambda_{in,1}^{(s)} \right]^{-1} = \mathcal{N} \int_0^{\mu_2} 2\mu \frac{d\sigma_{in}^{(s)}}{d\mu} d\mu \quad (4.115)$$

and

$$\left[\lambda_{in,2}^{(s)} \right]^{-1} = \mathcal{N} \int_0^{\mu_2} 6(\mu - \mu^2) \frac{d\sigma_{in}^{(s)}}{d\mu} d\mu. \quad (4.116)$$

In PENELOPE, soft electronic scattering is simulated together with soft elastic scattering, by means of the artificial distribution (4.30). The combined process is described by the transport mean free paths

$$\left[\lambda_{\text{comb},1}^{(s)}\right]^{-1} = \left[\lambda_{\text{el},1}^{(s)}\right]^{-1} + \left[\lambda_{\text{in},1}^{(s)}\right]^{-1} \quad (4.117)$$

and

$$\left[\lambda_{\text{comb},2}^{(s)}\right]^{-1} = \left[\lambda_{\text{el},2}^{(s)}\right]^{-1} + \left[\lambda_{\text{in},2}^{(s)}\right]^{-1}. \quad (4.118)$$

Thus, to account for soft electronic scattering we only have to replace the soft elastic transport mean free paths by those of the combined process.

4.3.3 Bielajew's alternate random hinge

Angular deflections due to soft interactions along a step of length s are generated from the artificial distribution (4.30) with first and second moments given by Eqs. (4.28) and (4.29), which are determined by the transport mean free paths $\lambda_{\text{comb},1}^{(s)}$ and $\lambda_{\text{comb},2}^{(s)}$. To account (at least partially) for the energy dependence of these quantities we use a trick due to Alex Bielajew. The soft energy loss and angular deflection (which occur at the hinge) are considered as independent processes and are simulated in random order. That is, the soft angular deflection is evaluated for the energy at either the beginning or the end of the step, with equal probabilities. This is equivalent to assuming that the transport mean free paths $\lambda_{\text{comb},1}^{(s)}(E)$ and $\lambda_{\text{comb},2}^{(s)}(E)$ vary linearly with energy. The method is fairly accurate and computationally inexpensive provided only that the fractional energy loss along each step (which is of the order of C_2) is sufficiently small.

4.4 Generation of random tracks

Each simulated electron or positron history consists of a chronological succession of events. These can be either hard events, artificial soft events (hinges) or other relevant stages of the particle history (such as its initial state, the crossing of an interface or the effective absorption after slowing down). The trajectory of the particle between a pair of successive events is straight and will be referred to as a “segment”. We keep the term “step” to designate the portion of a track between two hard events, which consists of two segments and a hinge (when mixed simulation is effective).

Simulation with PENELOPE is controlled by the constants C_1 and C_2 [see Eq. (4.85)] and the cutoff energies W_{cc} and W_{cr} . Hereafter, these four quantities will be referred to as simulation parameters. The parameter C_1 , which determines the mean free path $\lambda_{\text{el}}^{(h)}$ between hard elastic events, should be small enough to ensure reliable simulation results. PENELOPE admits values of C_1 from 0 (detailed simulation) up to 0.2, which corresponds to a mean angular deflection $\langle\theta\rangle \sim 37$ deg after a step-length $\lambda_{\text{el}}^{(h)}$. The simulation parameter C_2 gives the maximum average fractional energy loss in a single

step and it is effective only at high energies. From the discussion in Section 4.3, it is clear that C_2 should also be small. PENELOPE allows values of C_2 between zero and 0.2. The cutoff energies W_{cc} and W_{cr} mainly influence the simulated energy distributions. The simulation speeds up by using larger cutoff energies, but if these are too large the simulated distributions may be somewhat distorted. In practice, simulated energy distributions are found to be quite insensitive to the adopted values of W_{cc} and W_{cr} when these are less than the bin width used to tally the energy distributions. Thus, the desired energy resolution determines the maximum allowed cutoff energies.

The combined effect of all soft elastic and stopping interactions in a step is simulated as a single artificial event or hinge, in which the particle changes its direction of movement and loses energy. When W_{cc} is less than the lowest oscillator resonance energy, the simulation of inelastic collisions becomes purely detailed, *i.e.*, inelastic collisions do not contribute to the soft stopping power. On the other hand, the simulation of bremsstrahlung emission is only possible by means of a mixed scheme, because of the divergence of the DCS at $W = 0$ [see Eq. (3.140)]. To test the accuracy of mixed algorithms, and also in studies of low-energy electron and positron transport (with, say, $E < 100$ keV), it may be convenient to perform strictly detailed simulations (see below). For this purpose, PENELOPE allows the user to switch off the emission of soft bremsstrahlung photons with energy less than 10 eV. This option is activated when the W_{cr} value selected by the user is negative, in which case the program sets $W_{cr} = 10$ eV, disregards soft bremsstrahlung events and simulates hard events (with $W > 10$ eV) in a detailed way. The generation of the angular deflection in artificial events is discontinued when the simulation of elastic and inelastic scattering becomes detailed (*i.e.*, when $\lambda_{el}^{(h)} = \lambda_{el}$, $W_{cc} = 0$).

As indicated above, the length of the steps generated by PENELOPE is always less than s_{max} , an upper bound selected by the user. The simulation code limits the step length by placing delta interactions along the particle track. These are fictitious interactions that do not alter the state of the particle. Their only effect is to interrupt the sequence of simulation operations, which requires altering the values of inner control variables to permit resuming the simulation in a consistent way. The use of bounded step lengths is necessary to account for the energy dependence of the DCSs for soft interactions. However, this is not the only reason for limiting the step length. Since energy losses and deflections at the hinges are sampled from artificial distributions, the number of hinges per primary track must be “statistically sufficient”, *i.e.*, larger than ~ 10 , to smear off the unphysical details of the adopted artificial distributions. Therefore, 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. In PENELOPE, the parameter s_{max} can be varied freely during the course of the simulation of a single track. To ensure internal consistency, s_{max} is required to be less than $4\lambda_T^{(h)}$. When the user-selected value is larger, the code sets $s_{max} = 4\lambda_T^{(h)}$; in this case, about 2 per cent of the sampled steps have lengths that exceed s_{max} and are terminated by a delta interaction. This slows down the simulation a little ($\sim 5\%$), but ensures that the energy dependence of $\lambda_T^{(h)}$ is correctly accounted for. In order to eliminate an artifact in the depth-dose

distribution from parallel electron/positron beams near the entrance interface, instead of the $s_{\max}$ value set by the user, PENELOPE uses a random maximum step length [from a uniform distribution in the interval $(0.5, 1.0) \times s_{\max}$] that averages to 0.75 times the user's value. Incidentally, limiting the step length is also necessary to perform simulation of electron/positron transport in external static electromagnetic fields (see Appendix C).

The state of the particle immediately after an event is defined by its position coordinates $\mathbf{r}$, energy E and direction cosines of its direction of movement $\hat{\mathbf{d}}$, as seen from the laboratory reference frame. It is assumed that particles are locally absorbed when their energy becomes smaller than a preselected value E_{abs} ; positrons are considered to annihilate after absorption. The practical generation of random electron and positron tracks in arbitrary material structures, which may consist of several homogeneous regions of different compositions separated by well-defined surfaces (interfaces), proceeds as follows:

- (i) Set the initial position $\mathbf{r}$, kinetic energy E and direction of movement $\hat{\mathbf{d}}$ of the primary particle.
- (ii) Determine the maximum allowed soft energy loss $\omega_{\max}$ along a step and set the value of inverse mean free path for hard events (see Section 4.3). The results depend on the adopted $s_{\max}$, which can vary along the simulated track.
- (iii) Sample the distance s to be travelled to the following hard event (or delta interaction) as

$$s = -\ln \xi / \Sigma_{\text{h,max}}. \quad (4.119)$$

If $s > s_{\max}$, truncate the step by setting $s = s_{\max}$.

- (iv) Generate the length $\tau = s\xi$ of the step to the next hinge. Let the particle advance this distance in the direction $\hat{\mathbf{d}}$: $\mathbf{r} \leftarrow \mathbf{r} + \tau\hat{\mathbf{d}}$.
- (v) If the track has crossed an interface:

Stop it at the crossing point (*i.e.*, redefine $\mathbf{r}$ as equal to the position of this point and set τ equal to the travelled distance).

Go to (ii) to continue the simulation in the new material, or go to (xi) if the new material is the outer vacuum.
- (vi) Simulate the energy loss and deflection at the hinge. This step consists of two actions:
 - a) Sample the polar angular deflection $\mu = (1 - \cos \theta)/2$ from the distribution $F_a(s; \mu)$, Eq. (4.30), corresponding to the current energy E . Sample the azimuthal scattering angle as $\phi = 2\pi\xi$. Perform a rotation $\mathcal{R}(\theta, \phi)$ of the vector $\hat{\mathbf{d}}$ according to the sampled polar and azimuthal angular deflections (as described in Section 1.4.4) to obtain the new direction: $\hat{\mathbf{d}} \leftarrow \mathcal{R}(\theta, \phi)\hat{\mathbf{d}}$.
 - b) Sample the energy loss ω due to soft stopping interactions along the step s from the distribution $G_a(s; \omega)$, Eqs. (4.59)–(4.63), and reduce the kinetic energy: $E \leftarrow E - \omega$.

These two actions are performed in random order to account for the energy dependence of the soft transport mean free paths (see Section 4.3.3).

Go to (xi) if $E < E_{\text{abs}}$.

- (vii) Let the particle advance the distance $s - \tau$ in the direction $\hat{\mathbf{d}}$: $\mathbf{r} \leftarrow \mathbf{r} + (s - \tau)\hat{\mathbf{d}}$.
- (viii) Do as in (v).
- (ix) If in step (iii) the step length was truncated, *i.e.*, $s = s_{\max}$, simulate a delta interaction.
Go to (ii).
- (x) Simulate the hard event:
Sample the kind of interaction according to the point probabilities,

$$p_{\text{el}} = \frac{\mathcal{N}\sigma_{\text{el}}^{(\text{h})}}{\Sigma_{\text{h,max}}}, \quad p_{\text{in}} = \frac{\mathcal{N}\sigma_{\text{in}}^{(\text{h})}}{\Sigma_{\text{h,max}}}, \quad p_{\text{br}} = \frac{\mathcal{N}\sigma_{\text{br}}^{(\text{h})}}{\Sigma_{\text{h,max}}}, \quad p_{\text{si}} = \frac{\mathcal{N}\sigma_{\text{si}}}{\Sigma_{\text{h,max}}},$$

$$p_{\delta} = \frac{\Sigma_{\delta}}{\Sigma_{\text{h,max}}}, \quad \text{and} \quad p_{\text{an}} = \frac{\mathcal{N}\sigma_{\text{an}}}{\Sigma_{\text{h,max}}} \quad \text{in the case of positrons.} \quad (4.120)$$

If the event is a delta interaction, return to (ii).

If the event is an inner-shell ionisation, sample the active shell, simulate the relaxation cascade of the residual ion and return to (ii). Notice that in this case the state of the projectile remains unaltered.

Sample the polar scattering angle θ and the energy loss W from the corresponding DCS. Generate the azimuthal scattering angle as $\phi = 2\pi\xi$. Perform a rotation $\mathcal{R}(\theta, \phi)$ of the vector $\hat{\mathbf{d}}$ to obtain the new direction: $\hat{\mathbf{d}} \leftarrow \mathcal{R}(\theta, \phi)\hat{\mathbf{d}}$.

Reduce the kinetic energy of the particle: $E \leftarrow E - W$.

If, as a result of the interaction, a secondary particle is emitted in a direction $\hat{\mathbf{d}}_{\text{s}}$, with energy $E_{\text{s}} > E_{\text{abs}}$, store its initial state $(\mathbf{r}, E_{\text{s}}, \hat{\mathbf{d}}_{\text{s}})$.

Go to (ii) if $E > E_{\text{abs}}$.

- (xi) Simulate the tracks of the secondary electrons and photons produced by the primary particle (or by other secondaries previously followed) before starting a new primary track.

4.4.1 Stability of the simulation algorithm

The present simulation scheme for electrons/positrons is relatively stable under variations of the simulation parameters, due mostly to the effectiveness of the energy-loss corrections. This implies that the simulation parameters can be varied amply without practically altering the accuracy of the results. For the important case of low-energy electrons/positrons (with energies of the order of 500 keV or less), the relevant parameters are E_{abs} , C_1 , W_{cc} and $s_{\max}$, because C_2 is not effective (see Fig. 4.3) and radiative emission is unimportant (hard bremsstrahlung events occur very seldom and, therefore, W_{cr} has no influence). The value of the parameter $s_{\max}$ is important to ensure the reliability of the results; a safe recipe is to set $s_{\max}$ equal to one tenth of the “expected track length” or less. Since the values of E_{abs} and W_{cc} are dictated by the characteristics of the considered experiment, it follows that the only “critical” parameter, with a direct influence on the speed of the simulation, is C_1 . As mentioned above, PENELOPE accepts values of C_1 ranging from 0 (detailed simulation of elastic scattering) to 0.2.

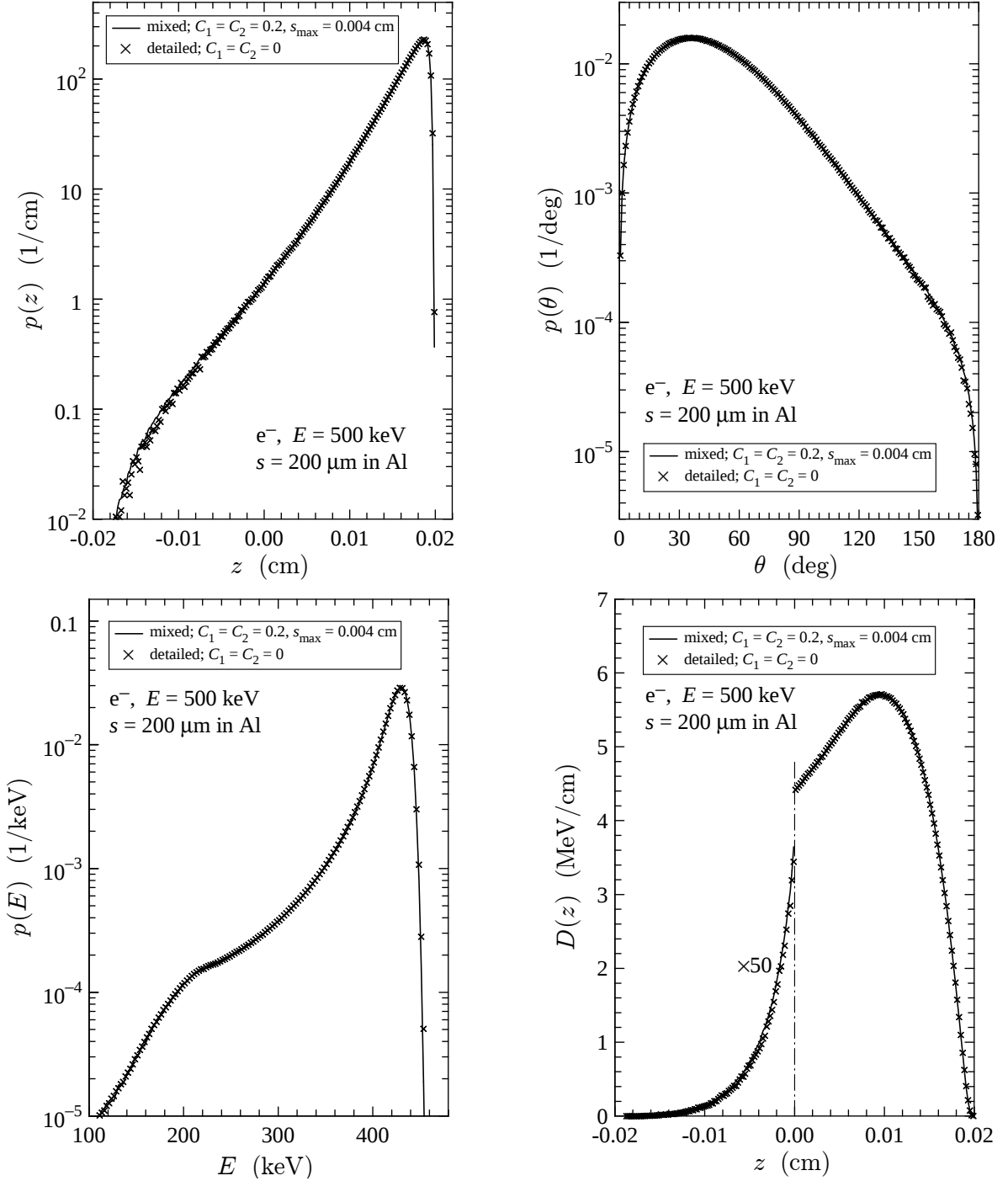


Figure 4.5: Results from the simulations of 500 keV electrons in aluminium described in the text. Crosses, detailed simulation; continuous curves, mixed simulation. $p(z)$ is the PDF of the z -coordinate of the final electron position, after travelling the prescribed 200 μm . $p(\theta)$ and $p(E)$ are the PDFs of the direction of motion (specified by the polar angle θ) and the kinetic energy E of the electrons at the end of the simulated tracks. The function $D(z)$ represents the “depth-dose” function, *i.e.*, the average energy deposited in the material per unit length along the z -direction (the residual energy at the end of the track is not included in the dose).

In practice, the value of C_1 does not influence the *accuracy* of the simulation results when the other parameters are given “safe” values. This is illustrated in Fig. 4.5, which displays results from simulations of 500 keV electrons in aluminium (infinite medium). Electrons started off from the origin of coordinates moving in the direction of the z axis. During the generation of each electron track, we scored the energy deposited at different “depths” (z -coordinate) to get the “depth-dose” distribution. The simulation of a track was discontinued when the electron had travelled a path length s equal to $200\ \mu\text{m}$, and the PDFs of the final electron energy and position coordinates were tallied. Notice that no secondary radiation was followed and that the kinetic energy of the electrons at $s = 200\ \mu\text{m}$ was not included in the dose distribution (*i.e.*, the calculated “dose” does not represent the quantity that would be obtained from a measurement).

The results displayed in Fig. 4.5 are from equivalent detailed and mixed simulations with $E_{\text{abs}} = 10\ \text{keV}$ and $s_{\text{max}} = 40\ \mu\text{m}$. The detailed simulation was performed by setting $C_1 = C_2 = 0$, $W_{\text{cc}} = 0$ and $W_{\text{cr}} = -100$. Notice that when the user enters a negative value of the cutoff energy loss for radiative events, PENELOPE sets $W_{\text{cr}} = 10\ \text{eV}$, disregards the emission of soft bremsstrahlung photons with $W < 10\ \text{eV}$ (which represents a negligible fraction of the stopping power) and simulates hard bremsstrahlung events as usually, that is, in a detailed way. The mixed simulation results shown in Fig. 4.5 were generated with $C_1 = C_2 = 0.2$, $W_{\text{cc}} = 1\ \text{keV}$ and $W_{\text{cr}} = -100$ (*i.e.*, radiative events were described as in the detailed simulation).

In the detailed simulation, about 10 million electron tracks were generated by running a modified version of the code `penslab.f` (see Section 6.2.1) on a 2.00 GHz Intel Pentium M computer for about 13 hours, which corresponds to a simulation speed of 211 tracks/s. The average numbers of elastic, inelastic and bremsstrahlung interactions that had to be simulated to produce each detailed track were 1297, 1171 and 0.03, respectively. On the same computer, the mixed simulation generated 100 million tracks in about 1.8 hours, which represents a simulation speed of 15,640 tracks/s, 74 times faster than that of detailed simulation. The reason for this higher speed is that, on average, there were only 2.4 hard elastic collisions, 6.3 hard inelastic collisions, 0.03 hard bremsstrahlung events and 3.2 delta interactions along each track. From Fig. 4.5 we conclude that, in this case, the mixed algorithm is not only stable under variations of the parameter C_1 over the accepted range (0,0.2), but also provides results that are essentially equivalent to those from the detailed simulation. It is worth recalling that detailed simulation is nominally exact, the results are affected only by statistical uncertainties.

In general, our mixed simulation algorithm yields very accurate results (*i.e.*, agreeing with those from detailed simulation) for electron and positron transport in infinite media, but not necessarily for limited geometries. The existence of interfaces poses considerable problems to condensed (class I) simulation, for which a satisfactory solution/approximation is not yet known. The present mixed (class II) algorithm handles interface crossing in a more accurate, but still approximate way. The rule to ensure accuracy for transport in the vicinity of interfaces is to use a small-enough value of s_{max} .

Chapter 5

Constructive quadric geometry

Practical simulations of radiation transport in material systems involve two different kinds of operations, namely physical (determination of the path length to the next interaction, random sampling of the different interactions) and geometrical (space displacements, interface crossings, . . .). In the case of material systems with complex geometries, geometrical operations can take a large fraction of the simulation time. These operations are normally performed by dedicated subroutine packages, whose characteristics depend on the kind of algorithm used to simulate the interactions. The material system is assumed to consist of a number of homogeneous bodies limited by well-defined surfaces. The evolution of particles within each homogeneous body is dictated by the physical simulation routines, which operate as if particles were moving in an infinite medium with a given composition. Normally, the physical routines can handle a number of different media, whose interaction properties have been previously stored in memory. The job of the geometry routines is to steer the simulation of particle histories in the material system. They must determine the active medium, change it when the particle crosses an interface (*i.e.*, a surface that separates two different media) and, for certain simulation algorithms, they must also keep control of the proximity of interfaces.

In this Chapter we describe the Fortran subroutine package PENGEO, which is suitable for detailed simulation algorithms (*i.e.*, algorithms where all single interactions in the history of a particle are simulated in chronological succession). Using these algorithms, the description of interface crossing is very simple: when the particle reaches an interface, its track is stopped just after entering a new material body and restarted again in the new active medium. This method (stopping and restarting a track when it crosses an interface) is applicable even when we have the same medium on both sides of the surface. That is, detailed simulations with a single homogeneous body and with the same body split into two parts by an arbitrary surface yield the same results (apart from statistical uncertainties).

As we have seen, detailed simulation is feasible only for photon transport and low-energy electron transport. For high-energy electrons and positrons, most Monte Carlo codes [*e.g.*, ETRAN (Berger and Seltzer, 1988), ITS3 (Halbleib *et al.*, 1992), EGS4 (Nel-

son *et al.*, 1985), EGSnrc (Kawrakow and Rogers, 2001), GEANT3 (Brun *et al.*, 1986), GEANT4 (Agostinelli *et al.*, 2003; Allison *et al.*, 2006)] have recourse to condensed (class I) or mixed (class II) simulation, where the global effect of multiple interactions along a path segment of a given length is evaluated using available multiple-scattering theories. To avoid large step lengths that could place the particle within a different medium, these condensed procedures require the evaluation of the distance from the current position to the nearest interface, an operation with a high computational cost (see, *e.g.*, Bielajew, 1995). The mixed procedure implemented in PENELOPE is, at least computationally, analogous to detailed simulation (it gives a “jump-and-knock” description of particle tracks). In fact, the structure of PENELOPE’s tracking algorithm was designed to minimise the influence of the geometry on the transport physics. This algorithm operates independently of interface proximity and only requires knowledge of the material at the current position of the particle. As a consequence, the geometry package PENGEOm can be directly linked to PENELOPE. However, since PENGEOm does not evaluate the distance to the closest interface, it cannot be used with condensed simulation codes, such as those mentioned above.

Let us mention in passing that in simulations of high-energy photon transport complex geometries can be handled by means of relatively simple methods, which do not require control of interface crossings (see, *e.g.*, Snyder *et al.*, 1969). Unfortunately, similar techniques are not applicable to electron and positron transport, mainly because these particles have much shorter track lengths and, hence, the transport process is strongly influenced by inhomogeneities of the medium. With the analogue simulation scheme adopted in PENELOPE, it is necessary to determine when a particle track crosses an interface, not only for electrons and positrons but also for photons.

With PENGEOm we can describe any material system consisting of homogeneous bodies limited by quadric surfaces. To speed up the geometry operations, the bodies of the material system can be grouped into modules (connected volumes, limited by quadric surfaces, that contain one or several bodies); modules can, in turn, form part of larger modules, and so on. This hierarchic modular structure allows a reduction of the work of the geometry routines, which becomes more effective when the complexity of the system increases. The present version 2006 of PENGEOm is able to simulate very complex material systems, with up to 5,000 bodies and 9,999 limiting surfaces. These large numbers of elements would be useless if one had to define the geometry manually, element by element, by editing enormous definition files. However, PENGEOm includes the option of cloning modules (and groups of modules), an operation which rapidly increases the number of elements used.

Except for trivial cases, the correctness of the geometry definition is difficult to check and, moreover, 3D structures with interpenetrating bodies are difficult to visualise. A pair of programs, named GVIEW2D and GVIEW3D, have been written to display the geometry on the computer screen. These programs use specific computer graphics software and, therefore, they are not portable. The executable files included in the PENELOPE distribution package run on personal computers under Microsoft Windows; they are simple and effective tools for debugging the geometry definition file.

5.1 Rotations and translations

The definition of various parts of the material system (quadric surfaces in reduced form and modules) involves rotations and translations. To describe these transformations, we shall adopt the active point of view: the reference frame remains fixed and only the space points (vectors) are translated or rotated.

In what follows, and in the computer programs, all lengths are in cm. The position and direction of movement of a particle are referred to the laboratory coordinate system, a Cartesian reference frame which is defined by the position of its origin of coordinates and the unit vectors $\hat{\mathbf{x}} = (1, 0, 0)$, $\hat{\mathbf{y}} = (0, 1, 0)$ and $\hat{\mathbf{z}} = (0, 0, 1)$ along the directions of its axes.

A translation $\mathcal{T}(\mathbf{t})$, defined by the displacement vector $\mathbf{t} = (t_x, t_y, t_z)$, transforms the vector $\mathbf{r} = (x, y, z)$ into

$$\mathcal{T}(\mathbf{t})\mathbf{r} = \mathbf{r} + \mathbf{t} = (x + t_x, y + t_y, z + t_z). \quad (5.1)$$

Evidently, the inverse translation $\mathcal{T}^{-1}(\mathbf{t})$ corresponds to the displacement vector $-\mathbf{t}$, *i.e.*, $\mathcal{T}^{-1}(\mathbf{t}) = \mathcal{T}(-\mathbf{t})$.

A rotation $\mathcal{R}$ is defined through the Euler angles ω , θ and ϕ , which specify a sequence of rotations about the coordinate axes¹: first there is a rotation of angle ω about the z -axis, followed by a rotation of angle θ about the y -axis and, finally, a rotation of angle ϕ about the z -axis. A positive rotation about a given axis would carry a right-handed screw in the positive direction along that axis. Positive (negative) angles define positive (negative) rotations. Although the definition of a rotation by means of the Euler angles may seem awkward, this is the most direct and compact method to specify a rotation. Figure 5.1 displays the effect of a rotation, defined by its Euler angles, on an asymmetric object, an arrow pointing initially along the z axis with a protuberance in the direction of the negative y axis. The rotation angles θ and ϕ are the polar and azimuthal angles of the final direction of the arrow, respectively. The first rotation, of angle ω , determines the final orientation of the protuberance. It is clear that, by selecting appropriate values of the Euler angles, we can set the object in any given orientation.

The rotation $\mathcal{R}(\omega, \theta, \phi)$ transforms the vector $\mathbf{r} = (x, y, z)$ into a vector

$$\mathbf{r}' = \mathcal{R}(\omega, \theta, \phi)\mathbf{r} = (x', y', z'), \quad (5.2)$$

whose coordinates are given by

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \mathcal{R}(\omega, \theta, \phi) \begin{pmatrix} x \\ y \\ z \end{pmatrix}, \quad (5.3)$$

¹This definition of the Euler angles is the one usually adopted in Quantum Mechanics (see, *e.g.*, Edmonds, 1960).

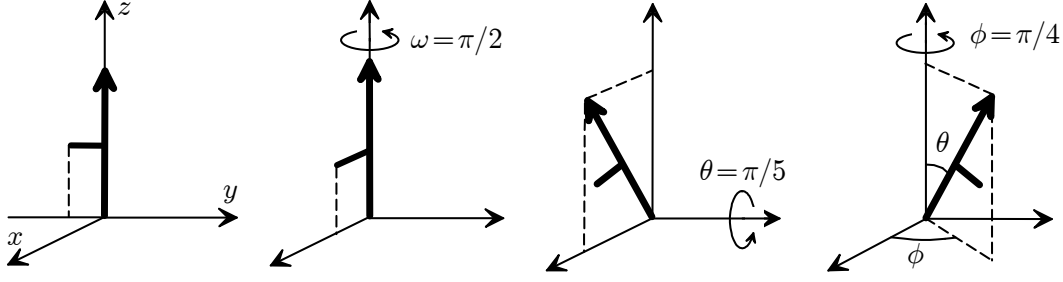


Figure 5.1: Rotation of an asymmetric object, an arrow with a protuberance, as defined by the Euler angles ω , θ and ϕ . The orientation of the axes is the same in all the diagrams.

where

$$\mathcal{R}(\omega, \theta, \phi) = \begin{pmatrix} R_{xx} & R_{xy} & R_{xz} \\ R_{yx} & R_{yy} & R_{yz} \\ R_{zx} & R_{zy} & R_{zz} \end{pmatrix} \quad (5.4)$$

is the rotation matrix. To obtain its explicit form, we recall that the matrices for rotations about the z - and y -axes are

$$\mathcal{R}_z(\phi) = \begin{pmatrix} \cos \phi & -\sin \phi & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{and} \quad \mathcal{R}_y(\theta) = \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix}, \quad (5.5)$$

respectively. Hence,

$$\begin{aligned} \mathcal{R}(\omega, \theta, \phi) &= \mathcal{R}_z(\phi) \mathcal{R}_y(\theta) \mathcal{R}_z(\omega) \\ &= \begin{pmatrix} \cos \phi & -\sin \phi & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix} \begin{pmatrix} \cos \omega & -\sin \omega & 0 \\ \sin \omega & \cos \omega & 0 \\ 0 & 0 & 1 \end{pmatrix} = \\ &= \begin{pmatrix} \cos \phi \cos \theta \cos \omega - \sin \phi \sin \omega & -\cos \phi \cos \theta \sin \omega - \sin \phi \cos \omega & \cos \phi \sin \theta \\ \sin \phi \cos \theta \cos \omega + \cos \phi \sin \omega & -\sin \phi \cos \theta \sin \omega + \cos \phi \cos \omega & \sin \phi \sin \theta \\ -\sin \theta \cos \omega & \sin \theta \sin \omega & \cos \theta \end{pmatrix}. \end{aligned} \quad (5.6)$$

The inverse of the rotation $\mathcal{R}(\omega, \theta, \phi)$ is $\mathcal{R}(-\phi, -\theta, -\omega)$ and its matrix is the transpose of $\mathcal{R}(\omega, \theta, \phi)$, *i.e.*,

$$\mathcal{R}^{-1}(\omega, \theta, \phi) = \mathcal{R}(-\phi, -\theta, -\omega) = \mathcal{R}_z(-\omega) \mathcal{R}_y(-\theta) \mathcal{R}_z(-\phi) = \mathcal{R}^T(\omega, \theta, \phi). \quad (5.7)$$

Let us now consider transformations $\mathcal{C} = \mathcal{T}(\mathbf{t}) \mathcal{R}(\omega, \theta, \phi)$ that are products of a rotation $\mathcal{R}(\omega, \theta, \phi)$ and a translation $\mathcal{T}(\mathbf{t})$. $\mathcal{C}$ transforms a point $\mathbf{r}$ into

$$\mathbf{r}' = \mathcal{C}(\mathbf{r}) = \mathcal{T}(\mathbf{t}) \mathcal{R}(\omega, \theta, \phi) \mathbf{r} \quad (5.8)$$

or, in matrix form,

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \mathcal{R}(\omega, \theta, \phi) \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}. \quad (5.9)$$

Notice that the order of the factors does matter; the product of the same factors in reverse order $\mathcal{D} = \mathcal{R}(\omega, \theta, \phi) \mathcal{T}(\mathbf{t})$ transforms $\mathbf{r}$ into a point $\mathbf{r}' = \mathcal{D}(\mathbf{r})$ with coordinates

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \mathcal{R}(\omega, \theta, \phi) \begin{pmatrix} x + t_x \\ y + t_y \\ z + t_z \end{pmatrix}. \quad (5.10)$$

Given a function $F(\mathbf{r})$, the equation $F(\mathbf{r}) = 0$ defines a surface in implicit form. We can generate a new surface by applying a rotation $\mathcal{R}(\omega, \theta, \phi)$ followed by a translation $\mathcal{T}(\mathbf{t})$ (we shall always adopt this order). The implicit equation of the transformed surface is

$$G(\mathbf{r}) = F[\mathcal{R}^{-1}(\omega, \theta, \phi) \mathcal{T}^{-1}(\mathbf{t}) \mathbf{r}] = 0, \quad (5.11)$$

which simply expresses the fact that $G(\mathbf{r})$ equals the value of the original function at the point $\mathbf{r}' = \mathcal{R}^{-1}(\omega, \theta, \phi) \mathcal{T}^{-1}(\mathbf{t}) \mathbf{r}$ that transforms into $\mathbf{r}$ [*i.e.*, $\mathbf{r} = \mathcal{T}(\mathbf{t})\mathcal{R}(\omega, \theta, \phi)\mathbf{r}'$].

5.2 Quadric surfaces

As already mentioned, the material system consists of a number of homogeneous bodies, defined by their composition (material) and limiting surfaces. For practical reasons, all limiting surfaces are assumed to be quadrics given by the implicit equation

$$\begin{aligned} F(x, y, z) = & A_{xx}x^2 + A_{xy}xy + A_{xz}xz + A_{yy}y^2 + A_{yz}yz + A_{zz}z^2 \\ & + A_x x + A_y y + A_z z + A_0 = 0, \end{aligned} \quad (5.12)$$

which includes planes, pairs of planes, spheres, cylinders, cones, ellipsoids, paraboloids, hyperboloids, etc. In practice, limiting surfaces are frequently known in “graphical” form and it may be very difficult to obtain the corresponding quadric parameters. Try with a simple example: calculate the parameters of a circular cylinder of radius R such that its symmetry axis goes through the origin and is parallel to the vector $(1,1,1)$. To facilitate the definition of the geometry, each quadric surface can be specified either through its implicit equation or by means of its reduced form, which defines the “shape” of the surface (see Fig. 5.2), and a few simple geometrical transformations.

A reduced quadric is defined by the expression

$$F_r(x, y, z) = I_1 x^2 + I_2 y^2 + I_3 z^2 + I_4 z + I_5 = 0, \quad (5.13)$$

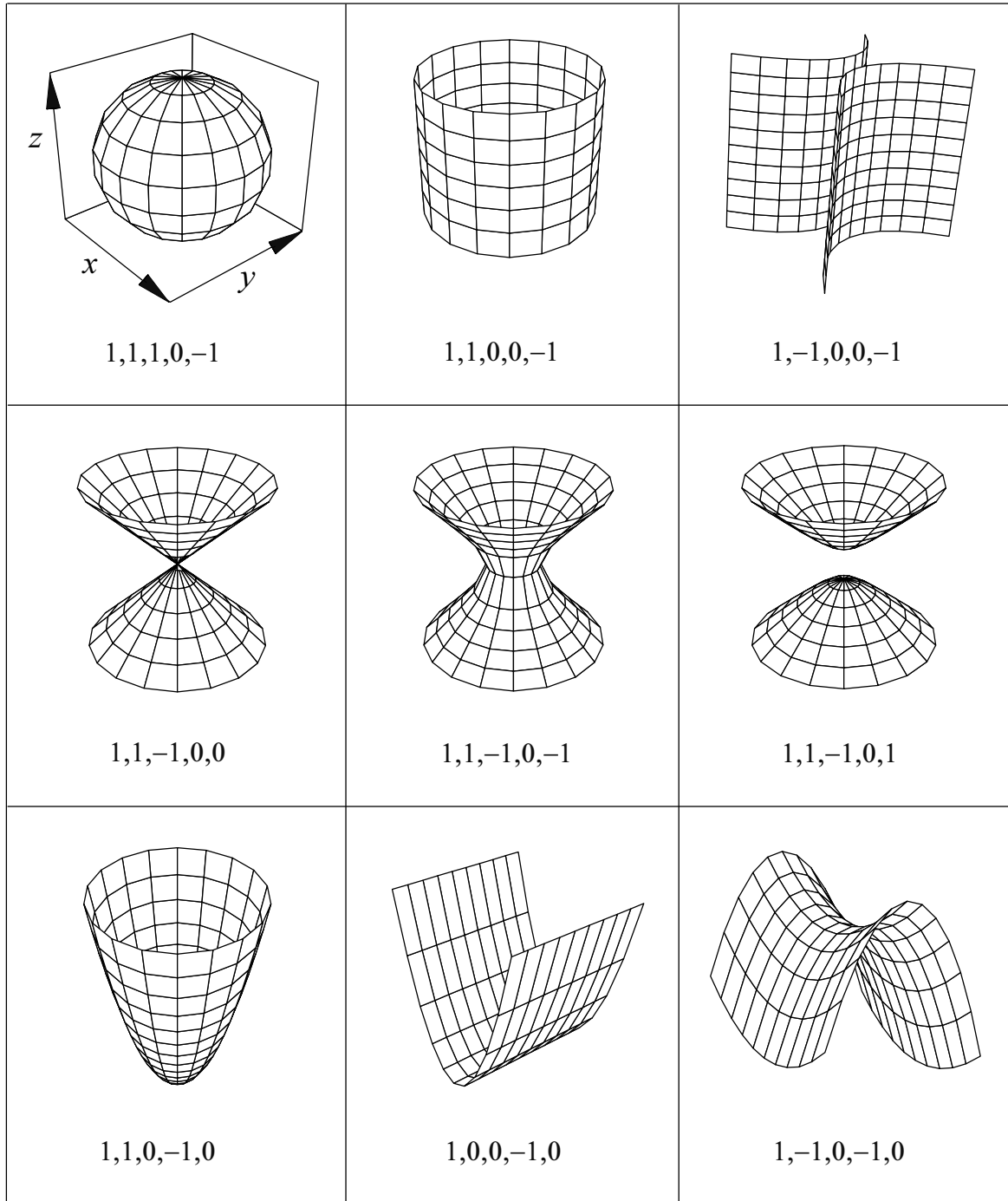


Figure 5.2: Non-planar reduced quadric surfaces and their indices [see Eq. (5.13)]. In all cases, the perspective is the same as for the sphere.

Table 5.1: Reduced quadrics.

Reduced form	Indices					Quadric
$z - 1 = 0$	0	0	0	1	-1	plane
$z^2 - 1 = 0$	0	0	1	0	-1	pair of parallel planes
$x^2 + y^2 + z^2 - 1 = 0$	1	1	1	0	-1	sphere
$x^2 + y^2 - 1 = 0$	1	1	0	0	-1	cylinder
$x^2 - y^2 - 1 = 0$	1	-1	0	0	-1	hyperbolic cylinder
$x^2 + y^2 - z^2 = 0$	1	1	-1	0	0	cone
$x^2 + y^2 - z^2 - 1 = 0$	1	1	-1	0	-1	one sheet hyperboloid
$x^2 + y^2 - z^2 + 1 = 0$	1	1	-1	0	1	two sheet hyperboloid
$x^2 + y^2 - z = 0$	1	1	0	-1	0	paraboloid
$x^2 - z = 0$	1	0	0	-1	0	parabolic cylinder
$x^2 - y^2 - z = 0$	1	-1	0	-1	0	hyperbolic paraboloid
... and permutations of x , y and z that preserve the central symmetry with respect to the z -axis.						

where the coefficients (indices) I_1 to I_5 can only take the values -1 , 0 or 1 . Notice that reduced quadrics have central symmetry about the z -axis, *i.e.*, $F_r(-x, -y, z) = F_r(x, y, z)$. The possible (real) reduced quadrics are given in Table 5.1.

A general quadric is obtained from the corresponding reduced form by applying the following transformations (in the quoted order)².

- (i) An expansion along the directions of the axes, defined by the scaling factors **X-SCALE**= a , **Y-SCALE**= b and **Z-SCALE**= c . The equation of the scaled quadric is

$$F_s(x, y, z) = I_1 \left(\frac{x}{a} \right)^2 + I_2 \left(\frac{y}{b} \right)^2 + I_3 \left(\frac{z}{c} \right)^2 + I_4 \frac{z}{c} + I_5 = 0. \quad (5.14)$$

For instance, this transforms the reduced sphere into an ellipsoid with semiaxes equal to the scaling factors.

- (ii) A rotation, $\mathcal{R}(\omega, \theta, \phi)$, defined through the Euler angles **OMEGA**= ω , **THETA**= θ and **PHI**= ϕ . Notice that the rotation $\mathcal{R}(\omega, \theta, \phi)$ transforms a plane perpendicular to the z -axis into a plane perpendicular to the direction with polar and azimuthal angles **THETA** and **PHI**, respectively. The first Euler angle, ω has no effect when the initial (scaled) quadric is symmetric about the z -axis.

²Keywords used to denote the various parameters in the geometry definition file are written in typewriter font, *e.g.*, **X-SCALE**). See Section 5.4.

- (iii) A translation, defined by the components of the displacement vector $\mathbf{t}$ (X-SHIFT = t_x , Y-SHIFT = t_y , Z-SHIFT = t_z).

A quadric is completely specified by giving the set of indices (I_1, I_2, I_3, I_4, I_5), the scale factors (X-SCALE, Y-SCALE, Z-SCALE), the Euler angles (OMEGA, THETA, PHI) and the displacement vector (X-SHIFT, Y-SHIFT, Z-SHIFT). Any quadric surface can be expressed in this way. The implicit equation of the quadric is obtained as follows. We define the matrix

$$\mathcal{A} = \begin{pmatrix} A_{xx} & \frac{1}{2}A_{xy} & \frac{1}{2}A_{xz} \\ \frac{1}{2}A_{xy} & A_{yy} & \frac{1}{2}A_{yz} \\ \frac{1}{2}A_{xz} & \frac{1}{2}A_{yz} & A_{zz} \end{pmatrix} \quad (5.15)$$

and write the generic quadric equation (5.12) in matrix form

$$\mathbf{r}^T \mathcal{A} \mathbf{r} + \mathbf{A}^T \mathbf{r} + A_0 = 0, \quad (5.16)$$

where $\mathbf{r}$ and $\mathbf{A} \equiv (A_x, A_y, A_z)$ are considered here as one-column matrices. The superscript “T” denotes the transpose matrix; thus, $\mathbf{r}^T$ is a one-row matrix. Notice that the matrix $\mathcal{A}$ is symmetric ($\mathcal{A}^T = \mathcal{A}$). Expressing the scaled quadric (5.14) in the form (5.16), the equation for the rotated and shifted quadric is [see Eq. (5.11)]

$$(\mathbf{r} - \mathbf{t})^T \mathcal{R} \mathcal{A} \mathcal{R}^T (\mathbf{r} - \mathbf{t}) + (\mathcal{R} \mathbf{A})^T (\mathbf{r} - \mathbf{t}) + A_0 = 0, \quad (5.17)$$

which can be written in the generic form (5.16)

$$\mathbf{r}^T \mathcal{A}' \mathbf{r} + \mathbf{A}'^T \mathbf{r} + A'_0 = 0 \quad (5.18)$$

with

$$\mathcal{A}' = \mathcal{R} \mathcal{A} \mathcal{R}^T, \quad \mathbf{A}' = \mathcal{R} \mathbf{A} - 2\mathcal{A}' \mathbf{t}, \quad A'_0 = A_0 + \mathbf{t}^T (\mathcal{A}' \mathbf{t} - \mathcal{R} \mathbf{A}). \quad (5.19)$$

From these relations, the parameters of the implicit equation (5.12) are easily obtained.

A quadric surface $F(x, y, z) = 0$ divides the space into two exclusive regions that are identified by the sign of $F(x, y, z)$, the *surface side pointer*. A point with coordinates (x_0, y_0, z_0) is said to be inside the surface if $F(x_0, y_0, z_0) \leq 0$ (side pointer = -1), and outside it if $F(x_0, y_0, z_0) > 0$ (side pointer = $+1$).

5.3 Constructive quadric geometry

A body is defined as a space volume limited by quadric surfaces and filled with a homogeneous material. To specify a body we have to define its limiting quadric surfaces $F(\mathbf{r}) = 0$, with corresponding side pointers ($+1$ or -1), and its composition (*i.e.*, the integer label used by PENELOPE to identify the material). It is considered that bodies are defined in “ascending”, exclusive order so that previously defined bodies effectively

delimit the new ones. This is convenient, for instance, to describe bodies with inclusions. The work of the geometry routines is much easier when bodies are completely defined by their limiting surfaces, but this is not always possible or convenient for the user. The example in Section 5.7 describes an arrow inside a sphere (Fig. 5.3); the arrow is defined first so that it limits the volume filled by the material inside the sphere. It is impossible to define the hollow sphere (as a single body) by means of only its limiting quadric surfaces.

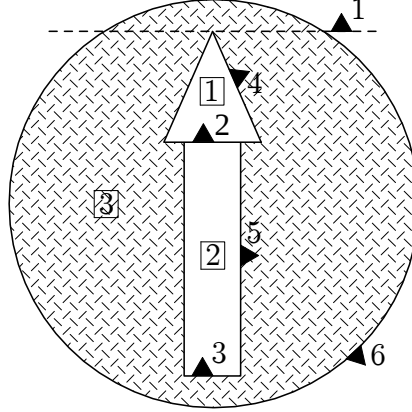


Figure 5.3: Example of simple quadric geometry; an arrow within a sphere (the corresponding definition file is given in Section 5.7). The solid triangles indicate the outside of the surfaces (side pointer = +1). Numbers in squares indicate bodies.

The subroutine package PENGEOm contains a subroutine, named LOCATE, that “locates” a point $\mathbf{r}$, *i.e.*, determines the body that contains it, if any. The obvious method is to compute the side pointers [*i.e.*, the sign of $F(\mathbf{r})$] for *all* surfaces and, then, explore the bodies in ascending order looking for the first one that fits the given side pointers. This brute-force procedure was used in older versions of PENGEOm; it has the advantage of being robust (and easy to program) but becomes too slow for complex systems. A second subroutine, named STEP, “moves” the particle from a given position $\mathbf{r}_0$ within a body B a certain distance s in a given direction $\hat{\mathbf{d}}$. STEP also checks if the particle leaves the active medium and, when this occurs, stops the particle just after entering the new material. To do this, we must determine the intersections of the track segment $\mathbf{r}_0 + t\hat{\mathbf{d}}$ ($0 < t \leq s$) with *all* the surfaces that limit the body B (including those that limit other bodies that limit B), and check if the final position $\mathbf{r}_0 + s\hat{\mathbf{d}}$ remains in B or not. The reason for using only quadric surfaces is that these intersections are easily calculated by solving a quadratic equation.

Notice that bodies can be concave, *i.e.*, the straight segment joining any two points in a body may not be wholly contained in the body. Hence, even when the final position of the particle lies within the initial body, we must analyse all the intersections of the path segment with the limiting surfaces of B and check if the particle has left the body after any of the intersections. When the particle leaves the initial body, say after travelling a distance $s' (< s)$, we have to locate the point $\mathbf{r}' = \mathbf{r}_0 + s'\hat{\mathbf{d}}$. The easiest method consists of

computing the side pointers of *all* surfaces of the system at $\mathbf{r}'$, and determining the body B' that contains $\mathbf{r}'$ by analysing the side pointers of the different bodies in ascending order. It is clear that, for complex geometries, this is a very slow process. We can speed it up by simply disregarding those elements of the system that cannot be reached in a single step (*e.g.*, bodies that are “screened” by other bodies). Unfortunately, as a body can be limited by all the other bodies that have been defined previously, the algorithm can be improved only at the expense of providing it with additional information. We shall adopt a simple strategy that consists of lumping groups of bodies together to form modules.

A module is defined as a connected volume³, limited by quadric surfaces, that contains one or several bodies. A module can contain other modules, which will be referred to as submodules of the first. The volume of a module is filled with a homogeneous medium, 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. A body that is connected and limited only by surfaces can be declared either as a body or as a module. For the sake of 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). Notice however, that the bodies of a module are still assumed to be defined in ascending order, *i.e.*, a body is limited by its surfaces and by the previously defined bodies *of the same module*, so that inclusions and interpenetrating bodies can be easily defined.

A module (with its possible submodules) can represent a rigid part (*e.g.*, a radioactive source, an accelerator head, a detector, a phantom, etc.) of a more complex material system. To facilitate the definition of the geometry, it is useful to allow free translations and rotations of the individual modules. The definition of a module (see below) includes the parameters of a rotation $\mathcal{R}(\omega, \theta, \phi)$ and a translation $\mathcal{T}(\mathbf{t})$, which are optional and serve to modify the position and orientation of the module (and its submodules) with respect to the laboratory reference frame. As before, the rotation is applied first. All submodules and bodies of the same module are shifted and rotated together.

In practical simulations with finite geometries, the simulation of a particle should be discontinued when it leaves the material system. In PENGEO this is done automatically by assuming that the complete system is contained within a large sphere of 10^7 cm radius, centred at the origin of coordinates. This sphere defines a module, the enclosure, which comprises the whole system. It is assumed that there is a perfect vacuum outside the enclosure, and in any inner volume that is not a part of a material body or of 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.

³A space volume is said to be connected when any two points in the volume can be joined by an arc of curve that is completely contained within the volume.

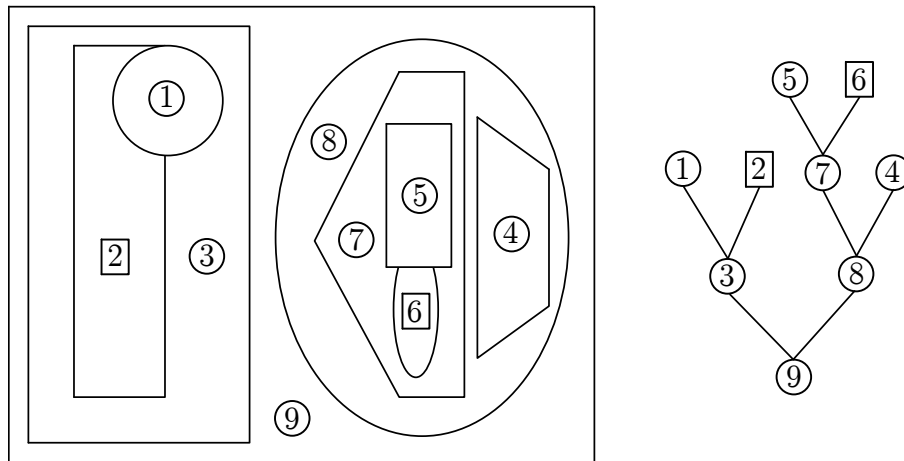


Figure 5.4: Planar cut of a geometry example, showing modules (number labels in circles) and bodies (number labels in squares), and the associated genealogical tree. Notice that a module can always be defined as a body limited by their submodules and daughter bodies, but this affects the structure of the genealogical tree and, therefore, the efficiency (speed) of the geometry operations.

For programming purposes, it is useful to imagine each module as the mother of its bodies and submodules, and as the daughter of the module that contains it. We thus have a kind of genealogical tree with various generations of modules and bodies (see Fig. 5.4). 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 body that is limited only by surfaces can be defined as a module. While bodies cannot be modified, modules can be rotated, translated and cloned (see below).

5.4 Geometry-definition file

The geometry is defined from the input file (UNIT=IRD). In principle, this permits the simulation of different geometries by using the same main program. The input file consists of a series of data sets, which define the different elements (surfaces, bodies and modules). A data set consists of a number of strictly formatted text lines; it starts and ends with a separation line filled with zeros. The first line after each separation line must start with one of the defining 8-character strings "SURFACE-", "SURFACE*", "BODY----", "MODULE--", CLONE----, "INCLUDE-" or "END-----" (here, blank characters are denoted

- The labels of an included file do not need to be different from those used in the main file or in other included files.
- PENGEOOM does not allow use of the elements of an included file to define new bodies or modules. In particular, the modules of an included file cannot be cloned in the main file. Conversely, the elements defined in the main file, before an **INCLUDE** block, are not usable in the included file.
- When we run one of the geometry viewers (see below), PENGEOOM generates the output file **geometry.rep**, which defines the same geometry as the input file. In **geometry.rep**, the elements of all the included structures are defined explicitly, and are labelled consecutively. We can rename this file to have the geometry defined in a single file.
- If the included structure is defined as a module, we can include it repeatedly with different global translation and rotation parameters. After including the structure for the first time, we can run one of the geometry viewers, and edit and rename the file output file **geometry.rep**. We can then once again include the structure in the new file, with different translation and rotation parameters, and repeat the process as many times as needed. The result is similar to that of the **CLONE** operation.

The definition of the geometry may seem somewhat more laborious than with combinatorial methods, where the system is described by combining basic bodies of several simple shapes [see Jenkins *et al.* (1988) and references therein]. In practice, however, defining the various surfaces that limit a body may be more convenient, and intuitive, than considering all the parameters needed to specify that body. The example of a right elliptical cylinder, which needs 9 parameters, is quite illustrative. With our method, this body can be defined as a module by means of two planes perpendicular to the z -axis (only one parameter if the base is the $z = 0$ plane) and a scaled cylinder (2 parameters); the rotation (3 parameters) of the module gives the required orientation and the translation (3 parameters) puts it in the required position. The definition as a proper body requires defining the three surfaces that limit the cylinder in its actual position, which is a bit more inconvenient. In any case, the important issue is not how to define the geometry, but the amount of computation needed to follow a particle through the material system.

5.5 The subroutine package PENGEOOM

The package PENGEOOM consists of the following subroutines;

- **SUBROUTINE GEOMIN(PARINP, NPINP, NMAT, NBOD, IRD, IWR)**
Reads geometry data from the input file and initialises the geometry package.
 - Input arguments:
PARINP: Array containing optional parameters, which may replace the ones entered from the input file. This array must be declared in the main program, even when

NPINP is equal to zero.

NPINP: Number of parameters defined in PARINP (positive).

IRD: Input file unit (opened in the main program).

IWR: Output file unit (opened in the main program).

◦ Output arguments:

NMAT: Number of different materials in full bodies (excluding void regions).

NBOD: Number of defined bodies and modules.

Subroutine **GEOMIN** labels elements of the various kinds (surfaces, bodies and modules) in strictly increasing order; it may also redefine some of the geometry parameters, whose actual values are entered through the array **PARINP**. A copy of the geometry definition file, with the effective parameter values and with the element labels assigned by PENGEO, is printed on the output file (UNIT IWR). This part of the output file describes the actual geometry used in the simulation.

NB: During the simulation, geometry elements are identified by the internal labels assigned by PENGEO, which are written in the output file. These labels may be different from those in the geometry definition file.

- SUBROUTINE LOCATE

Determines the body that contains the point with coordinates (X, Y, Z).

◦ Input values (through COMMON/TRACK/)⁴ :

X, Y, Z: Particle position coordinates.

U, V, W: Direction cosines of the direction of movement.

◦ Output values (through COMMON/TRACK/):

IBODY: Body where the particle moves.

MAT: Material in IBODY. The output MAT = 0 indicates that the particle is in a void region.

- SUBROUTINE STEP(DS,DSEF,NCROSS)

Used in conjunction with PENELOPE, this subroutine performs the geometrical part of the track simulation. The particle starts from the point (X,Y,Z) and proceeds to travel a length DS in the direction (U,V,W) within the material where it moves. STEP displaces the particle and stops it at the end of the step, or just after entering a new material (particles are not halted at “interfaces” between bodies of the same material). The output value DSEF is the distance travelled within the initial material. If the particle enters a void region, STEP continues the particle track, as a straight segment, until it penetrates a material body or leaves the system (the path length through inner void regions is not included in DSEF). When the particle arrives from a void region (MAT = 0), it is stopped after entering the first material body. The output value MAT = 0 indicates that the particle has escaped from the system.

⁴Most of the input/output of the geometry routines is through COMMON/TRACK/, which is the common block used by PENELOPE to transfer particle state variables (see Section 6.1.2).

- Input-output values (through `COMMON/TRACK/`):
 - `X`, `Y`, `Z`: Input: coordinates of the initial position.
Output: coordinates of the final position.
 - `U`, `V`, `W`: Direction cosines of the displacement. They are kept unaltered.
 - `IBODY` Input: initial body, *i.e.*, the one that contains the initial position.
Output: final body.
 - `MAT`: Material in body `IBODY` (automatically changed when the particle crosses an interface).
- Input argument:
 - `DS`: Distance to travel (unaltered).
- Output arguments:
 - `DSEF`: Travelled path length before leaving the initial material or completing the jump (less than `DS` if the track crosses an interface).
 - `NCROSS`: Number of interface crossings ($=0$ if the particle does not leave the initial material, greater than 0 if the particle enters a new material).

For the handling and storage of geometric information we take advantage of the structure of the genealogical tree. The geometry enclosure (see Section 5.3) is the only common ancestor for all bodies and modules. To understand the operation of the geometry routines, it is convenient to define a matrix `FLAG(KB,KS)` as follows (the indices `KS` and `KB` indicate the label of a surface and a body or module, respectively), `FLAG(KB,KS) = 1`, if `KS` is a limiting surface of `KB` and `KB` is inside `KS` (*i.e.*, side pointer = -1).
 $= 2$, if `KS` is a limiting surface of `KB` and `KB` is outside `KS` (*i.e.*, side pointer = +1).
 $= 3$, if `KB` is a body and `KS` does not directly limit `KB`, but appears in the definition of a body that limits `KB`.
 $= 4$, if `KB` is a module and `KS` limits one of its daughters (bodies and submodules), but does not appear in the definition of `KB`.
 $= 5$, otherwise.

To locate a point we call subroutine `LOCATE`, where we proceed upwards in the genealogical tree of modules. If the point is outside the enclosure, we set `MAT = 0` and return to the main program. Otherwise, we look for a module or body of the second generation that contains the point. If it exists, we continue analysing its descendants (if any) and so on. The process ends when we have determined the body `IBODY` that contains the point, or as soon as we conclude that the point is outside the material system (*i.e.*, in a void region). Notice that, when we have found that a module `KB` does contain the point, to do the next step we only need to consider the surfaces `KS` such that `FLAG(KB,KS) = 1, 2` or `4`.

After the body `IBODY` that contains the initial position of the particle has been identified, we can call subroutine `STEP` to move the particle a certain distance `DS`, dictated by `PENELOPE`, along the direction (`U,V,W`). We start by checking whether the track segment crosses any of the surfaces that limit `IBODY`. If after travelling the distance `DS` the

particle remains within the same body, DSEF is set equal to DS and control is returned to the main program. It is worth noting that the surfaces KS that define the initial body are those with FLAG(IBODY,KS)=1 and 2 (proper limiting surfaces) or =3 (limiting surfaces of limiting bodies). Although it may happen that a surface with FLAG=3 does not directly limit the body, subroutine STEP cannot know this from the information at hand and, consequently, all surfaces with FLAG=3 are analysed after each move. It is clear that, to reduce the number of surfaces to be considered, we should minimise the number of bodies used to delimit other bodies.

When the particle leaves IBODY and enters a new material, STEP stops it just after crossing the interface and determines the new body and material (in this case, the output values of IBODY and MAT are different from the input ones). To do this, the limiting surfaces of the parent module and of all the sisters of the initial body must be analysed (if they exist). If the new position is outside the parent module, we must analyse all surfaces that limit the parent's sisters and go downward in the genealogical tree to determine the module that contains the point and, if necessary, go upwards again to find out what the new body is. If the new material is the same as in the initial body, the particle is allowed to move the remaining distance. Void regions (strict vacuum) are crossed freely (*i.e.*, the distance travelled within these regions is not counted). Furthermore, when the particle starts from outside the enclosure, it is allowed to propagate freely until it reaches a material body. The particle is stopped when it penetrates a different material or when it leaves the system (*i.e.*, when, after leaving a material body, its straight trajectory does not intersect a non-void body; in this case, the value MAT=0 is returned).

Evidently, the speed of the geometry subroutines depends greatly on the structure of the modules' genealogical tree. The responsibility of optimising it rests with the user. Subroutine GEOMIN issues a geometry report containing the genealogical tree, information on redundant surfaces and an indication of the adequacy of the geometry simulation (based on the number of elements used to define the bodies and modules of the structure). This report is written on the output file `pengeom_tree.rep`. It is recommended inspecting this file and checking whether the modular tree is sufficiently ramified.

When STEP moves the particle across an interface, there is a risk that, owing to numerical truncation errors, the particle is placed on the wrong side of the interface (*i.e.*, the track is stopped just before the interface). If this occurs, the program could go into an endless loop in which STEP repeatedly tries to move the particle a very small distance (of the order of 10^{-15} cm) towards the interface but does not succeed, *i.e.*, the particle is trapped at the interface. To avoid this collapse of the trajectory, after each interface crossing, STEP applies an additional small displacement (equal to 10^{-8} cm plus $\sim 10^{-12}$ times the step length) in the direction of movement, which is physically irrelevant and sufficient to compensate for the effect of truncation errors. The same strategy is used in subroutine LOCATE: when the particle is too close to an interface, it is moved 10^{-8} cm along the surface gradient direction or its opposite, depending on whether the particle approaches or leaves the interface. Notice that this strategy requires that the direction of movement (U,V,W) be defined before calling LOCATE. The

small extra displacement effectively eliminates the risk of particle trapping at interfaces; but it also sets a limit to the space resolution (geometrical details that are less than ~ 10 Å in size cannot be described).

PENGEOM admits up to 9,999 surfaces and 5,000 bodies and modules. When the input file contains a larger number of elements, the program stops and a corresponding error message is printed. To describe such complex material systems, it is necessary to edit the source file `pengeom.f` and increase the values of the parameters `NS` (maximum number of surfaces) and `NB` (maximum number of bodies) in all subroutines. It is assumed that the number of bodies in a module is less than `NX = 250`, which is also the upper limit for the number of surfaces that can be used to define a body or a module (those with `FLAG < 5`). When this value of `NX` is insufficient, the module that causes the trouble should be decomposed into several submodules. Although it is possible to increase the parameter `NX`, this would waste a lot of memory. Moreover, simulation of a geometry with a module or body defined by such a large number of elements would be extremely slow.

5.5.1 Impact detectors

To extract information about particle fluxes within the geometrical structure, the user can define *impact detectors*. Each impact detector consists of a set of active bodies, which should have been defined as parts of the geometry. The definition of impact detectors is through the common block

```
COMMON/QKDET/KDET(NB)
```

After the call to subroutine `GEOMIN`, a body `KB` is made part of impact detector number `IDET` by setting `KDET(KB)=IDET`. The action of subroutine `STEP` for impact detectors is the following: when a transported particle enters the active body `KB` from vacuum or from another body that is not part of detector `IDET`, the particle is halted at the surface of the active body, and control is returned to the main program (particles would not be halted at interfaces between ordinary bodies of the same material). This strategy allows us to tally the energy spectrum and the angular distribution of “detected” particles or, more specifically, to generate a *phase-space file*, where the state variables of particles at the detector entrance are recorded.

Each body can only be part of one impact detector. Active bodies cannot be void, because the geometry routines would not stop particles at their limiting surfaces. In case you need to define detectors outside the material system, fill their volume with an arbitrary material of very small density to avoid perturbing the transport process. Notice that a detected particle can leave the detector volume and re-enter it afterwards. Consequently, particles can be “counted” several times. To avoid multiple counting, special action must be taken in the main program.

In its normal operation mode, `STEP` does not stop particles at surfaces that limit adjacent bodies of the same material, because this would slow down the simulation unnecessarily. Therefore, these surfaces are “invisible” from the main program. However,

if two adjacent bodies of the same material are part of different impact detectors, their common limiting surface becomes visible: when a transported particle reaches this surface it is halted, and control is transferred to the main program. Thus, by assigning each body to a different impact detector, we can keep track of all the bodies that are entered by a particle. This feature is used in the geometry viewer GVIEW2D (see below) to display the various bodies intersected by the screen plane in different colours.

5.6 Debugging and viewing the geometry

A pair of computer programs named GVIEW2D and GVIEW3D have been written to visualise the geometry and to help the user to debug the definition file. These codes generate two- and three-dimensional 24-bit colour 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 GVIEW2D is that displayed pictures are generated by using the PENGEOM package and, therefore, errors and inconsistencies in the geometry definition file that would affect the results of actual simulations are readily identified. The method to generate the image consists of following a particle that moves on a plane perpendicular to an axis of the reference frame, which is mapped on the window. The particle starts from a position that corresponds to the left-most pixel and moves along a straight trajectory to the right of the window. To do this, subroutine **STEP** is called repeatedly, maintaining the direction of movement and with a large value of **DS** (such that each body is crossed in a single step). A colour code is assigned to each material or body, and pixels are lit up with the active colour when they are crossed by the particle trajectory. The active colour is changed when the particle enters a new material or body. The final picture is a map of the bodies or materials intersected by the window plane. The orientation of the window plane, as well as the position and size of the window view, may be changed interactively by entering one of the one-character commands shown in Table 5.2, directly from the graphics window (upper- and lower-case letters may work differently). With GVIEW2D we can inspect the internal structure of the system with arbitrary magnification (limited only by the intrinsic resolution of the PENGEOM routines).

When running the GVIEW2D program, you will be asked to give the path+name of the geometry definition file, the coordinates (**XC,YC,ZC**) of the centre of the window (relative to the laboratory frame, in cm) and the display mode (materials or bodies). The window may appear black (the colour for void regions) if no material bodies are intersected. In this case, use the one-character viewer commands to reach the bodies or, more conveniently, start again and place the window centre near or within a filled body. Notice that, in the body display mode, the body labels shown on the screen are the ones used internally by PENELOPE. These internal labels, which may be different from the labels in the user definition file, are needed for activating variance-reduction methods locally, for defining impact detectors, and for scoring purposes (*e.g.*, to determine the

Table 5.2: One-character commands of the GVIEW2D geometry viewer.

```

+++++
+  x --> change window orientation, x-axis,      +
+  y --> change window orientation, y-axis,      +
+  z --> change window orientation, z-axis,      +
+  r,right --> shift right,          l,left --> shift left,  +
+  u,up    --> shift up,            d,down --> shift down,  +
+  f,pgup  --> shift front,        b,pgdn --> shift back,   +
+  i,+     --> zoom in,            o,-     --> zoom out,    +
+  1       --> actual size,        h,?     --> help,        +
+  blank, enter --> repeat last command,      q --> quit.   +
+++++

```

distribution of energy deposited within a particular body).

GVIEW3D generates three-dimensional pictures of the geometry by using a simple ray-tracing algorithm, with the source light and the camera at the same position. Bodies are displayed with the same colour code used by GVIEW2D (in the material display mode) and the intensity of each pixel is determined by the angle between the vision line and the normal to the limiting surface. This method does not produce shadows and disregards light diffusion, but makes fairly realistic three-dimensional images. The camera is assumed to be outside the system (placing the camera inside a body would stop the program). To reveal the inner structure of the system, the program can eliminate a wedge (limited by two vertical planes that intersect in the z -axis). The position and size of the system can be modified by means of one-character commands entered from the graphics window. The command keys and actions are similar to those of GVIEW2D. It is worth noting that GVIEW3D generates the image pixel by pixel, whereas GVIEW2D does it by drawing straight lines on the window; as a result, GVIEW2D is much faster.

GVIEW2D and GVIEW3D produce an output file named `geometry.rep` (which is generated by subroutine `GEOMIN`) in the working directory. The programs are stopped either when an input format is incorrect (reading error) or when a clear inconsistency in the definition file is found (*e.g.*, when the element that is being defined and the furnished information do not match). The wrong datum appears in the last printed lines of the `geometry.rep` file, usually in the last one. Error messages are also written on that file, so that the identification of inconsistencies is normally very easy. When the structure of the input file is correct, the codes do not stop and the geometry is displayed for further analysis.

Most of the possible errors in the input file can only be revealed by direct inspection of the images generated by GVIEW2D and GVIEW3D. Very frequently, errors are caused by the omission of a limiting element in the definition of a body or module. Numerical round-off errors can also cause inconsistencies in the geometry, which may be difficult to detect. Typically, these errors occur when two bodies, defined by different surfaces, are in contact; round-off errors in the surface coefficients may cause a small overlap of the two bodies, which can seriously damage the geometry.

When the operations **CLONE** and **INCLUDE** are not used, the output file **geometry.rep** is a duplicate of the input definition file. The only differences between the two files are the labels assigned to the different surfaces, bodies and modules; in **geometry.rep**, these elements are numbered in strictly increasing order. It is important to bear in mind that **PENGEOM** uses this sequential labelling internally to identify bodies and surfaces. When the input definition file contains **CLONE** or **INCLUDE** blocks, the output file **geometry.rep** provides a complete definition of the geometry, with the elements of cloned and included structures inserted and numbered sequentially with the internal labels. This file can be renamed and used as the definition file (the name **pengeom.rep** cannot be used for a geometry definition file, because subroutine **GEOMIN** assigns it to an output file); then, the definition files of the included structures are no longer needed.

5.7 A short tutorial

To prepare a new geometry definition file, it is useful to start from a file that contains a model of each data set with default values of their parameters. Placing the end-line at the beginning of the model group discontinues the geometry reading; so that the model group can be kept in the geometry file, even when this one is operative. The starting file should look like this

[illegible]

```

    PHI=(+0.0000000000000000E+00,  0) RAD          (DEFAULT=0.0)
X-SHIFT=(+0.0000000000000000E+00,  0)              (DEFAULT=0.0)
Y-SHIFT=(+0.0000000000000000E+00,  0)              (DEFAULT=0.0)
Z-SHIFT=(+0.0000000000000000E+00,  0)              (DEFAULT=0.0)
00000000000000000000000000000000000000000000000000000000000000000000
BODY   (    )    text
MATERIAL(    )
SURFACE (    ), SIDE POINTER=( 1)
BODY   (    )
MODULE (    )
00000000000000000000000000000000000000000000000000000000000000000000
MODULE (    )    text
MATERIAL(    )
SURFACE (    ), SIDE POINTER=( 1)
BODY   (    )
MODULE (    )
11111111111111111111111111111111111111111111111111111111111111111111
    OMEGA=(+0.0000000000000000E+00,  0) DEG          (DEFAULT=0.0)
    THETA=(+0.0000000000000000E+00,  0) DEG          (DEFAULT=0.0)
    PHI=(+0.0000000000000000E+00,  0) RAD          (DEFAULT=0.0)
X-SHIFT=(+0.0000000000000000E+00,  0)              (DEFAULT=0.0)
Y-SHIFT=(+0.0000000000000000E+00,  0)              (DEFAULT=0.0)
Z-SHIFT=(+0.0000000000000000E+00,  0)              (DEFAULT=0.0)
00000000000000000000000000000000000000000000000000000000000000000000
CLONE  (    )    copies one module and moves it
MODULE (    )    original module
11111111111111111111111111111111111111111111111111111111111111111111
    OMEGA=(+0.0000000000000000E+00,  0) DEG          (DEFAULT=0.0)
    THETA=(+0.0000000000000000E+00,  0) DEG          (DEFAULT=0.0)
    PHI=(+0.0000000000000000E+00,  0) RAD          (DEFAULT=0.0)
X-SHIFT=(+0.0000000000000000E+00,  0)              (DEFAULT=0.0)
Y-SHIFT=(+0.0000000000000000E+00,  0)              (DEFAULT=0.0)
Z-SHIFT=(+0.0000000000000000E+00,  0)              (DEFAULT=0.0)
00000000000000000000000000000000000000000000000000000000000000000000
INCLUDE
    FILE=(filename.ext)
00000000000000000000000000000000000000000000000000000000000000000000

```

To generate a new element, we just duplicate the corresponding data set, modify the parameter values and eliminate the lines that are unnecessary (*i.e.*, those of parameters that take their default values). Of course, the defining data set must be placed before the end-line. The progressing geometry can be visualised with GVIEW2D as soon as the first complete body or module has been defined. If GVIEW2D stops before entering the graphics mode, the geometry definition is incorrect and we should have a look at the `geometry.rep` file to identify the problem. Normally, the conflicting parameter or element appears in the last line of this file.

The basic elements of the geometry definition are quadric surfaces. These can be visualised by using the following file, which defines the inner volume of a reduced quadric as a single body,

```

-----
Visualization of reduced quadric surfaces.
Define the desired quadric (surface 1) by entering its indices.

```


the definition file. Notice that, when the detector is tilted, it is very difficult to get an idea of its geometry from the images generated by GVIEW2D. The file **saturne** describes the head of an electron accelerator, quite a complicated geometry with 96 surfaces and 44 bodies. The structure **male**, which corresponds to a mathematical anthropomorphic phantom, consists of 175 surfaces and 110 bodies and modules. The files **arrow2** and **saturne9** illustrate the use of the **CLONE** operation. The file **axes** defines three cylinders along the axes of the reference frame. This file can be visualised with GVIEW3D to study the effect of rotations; notice that the first image generated by GVIEW3D displays the position and orientation of the reference frame.

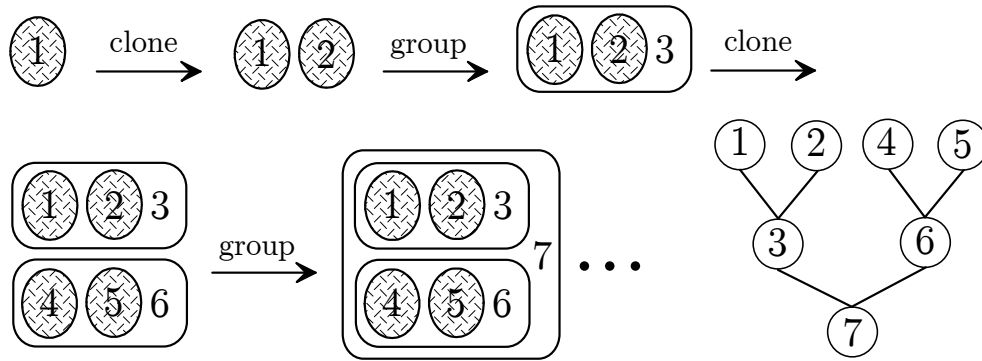


Figure 5.5: Schematic representation of the construction of an array of cells by progressively cloning and grouping modules obtained from a primary cell module. The labels of the modules are the internal labels assigned by PENGEO. Notice the evolution of the modular tree at each step.

In certain applications (*e.g.*, radiation imaging systems), we need to generate large geometrical systems with many identical elements. The **CLONE** function was devised to simplify the definition of these systems. To illustrate the use of **CLONE**, let us consider the simple case shown in Fig. 5.5. The system to be described consists of an array of identical cells, each of them defined as a module (which may have a complex internal structure). We start by defining the first cell (1). We clone this cell to produce a twin cell (2); and we group the two of them into a two-cell module (3). To build this module, we only need to specify its limiting surfaces. Then the two-cell module is cloned (6) and the resulting two modules are grouped together to produce a four-cell module (7). The process can be repeated as many times as required to generate a large array. At each step, the size (number of cells) of the array is doubled, with only a small amount of editing work. It is interesting to analyse the structure of the modular tree resulting from this construction. As shown in Fig. 5.5, each time we clone a module, the tree of the original module is duplicated; when we group the original module and its clone, their trees are joined and rooted into the new mother module, which has only two daughters. Therefore, the final tree is optimally ramified.

We cannot finish without a word of caution about the use of PENGEO, and other general-purpose geometry packages. For simple geometries, they tend to waste a lot of

time. It is always advisable to consider the possibility of handling geometric aspects directly; this may enable substantial reduction of the number of operations by taking full advantage of the peculiarities of the material system.

Chapter 6

Structure and operation of the code system

In this Chapter we describe the structure of the PENELOPE code system and its operation. The kernel of the system is the Fortran subroutine package PENELOPE, which performs “analogue” simulation of electron-photon showers (*i.e.*, the simulated showers are intended to be replicas of actual showers) in infinite (unbounded) media of various compositions. Photon histories are generated by using the detailed simulation method (see Section 1.4), *i.e.*, all interaction events are simulated in chronological succession. The generation of electron and positron tracks is performed by using the mixed procedure described in Chapter 4. Secondary particles emitted with initial energy larger than the absorption energy –see below– are stored, and simulated after completion of each primary track. Secondary particles are produced in direct interactions (hard inelastic collisions, hard bremsstrahlung emission, positron annihilation, Compton scattering, photoelectric absorption and pair production) and as fluorescent radiation (characteristic x rays and Auger electrons). PENELOPE simulates fluorescent radiation that results from vacancies produced in K, L and M shells by photoelectric absorption and Compton scattering of photons and by electron/positron impact. The relaxation of these vacancies is followed until the K, L and M shells are filled up, *i.e.*, until the vacancies have migrated to N and outer shells.

Being a subroutine package, PENELOPE cannot operate by itself. The user must provide a steering main program for his particular problem. Nevertheless, this main program is normally fairly simple because it only has to control the evolution of the tracks simulated by PENELOPE and keep score of relevant quantities. PENELOPE has been devised to do the largest part of the simulation work. It allows the user to write his own simulation program, with arbitrary geometry and scoring, without previous knowledge of the intricate theoretical aspects of scattering and transport theories. In the case of material systems with quadric geometries, the geometrical operations can be done automatically by using the package PENGEO (see Chapter 5). The distribution package also includes various examples of main programs for simple geometries (slab and cylindrical) and for general quadric geometries with limited scoring. Although they

are mostly intended to illustrate the use of the simulation routines, they do allow the study of many cases of practical interest. The complete program system is written in Fortran 77 format, but uses a few extensions to the ANSI/ISO standard Fortran 77, which are supported by most compilers. Therefore, the programs should run on any platform with a Fortran 77 or Fortran 90 compiler. For a readable programmer's guide to Fortran 77, see Page's book (1988).

6.1 PENELOPE

PENELOPE simulates coupled electron-photon transport in arbitrary material systems consisting of a number of homogeneous regions (bodies) limited by sharp (and passive) interfaces. Initially, it was devised to simulate the PENetration and Energy LOSS of Positrons and Electrons in matter; photons were introduced later. The adopted interaction models (Chapters 2 to 4), and the associated databases, allow the simulation of electron/positron and photon transport in the energy range from 50 eV to 1 GeV.

It should be borne in mind that our approximate interaction models become less accurate when the energy of the transported radiation decreases. Actually, for energies below ~ 1 keV, the DCSs are not well known, mostly because they are strongly affected by the state of aggregation (*i.e.*, by the environment of the target atom). On the other hand, for electrons and positrons, the trajectory picture ceases to be applicable (because coherent scattering from multiple centres becomes appreciable) when the de Broglie wavelength, $\lambda_B = (150 \text{ eV}/E)^{1/2} \text{ \AA}$, is similar to or greater than the interatomic spacing ($\sim 2 \text{ \AA}$). Therefore, results from simulations with PENELOPE (or with any other Monte Carlo trajectory code) for energies below 1 keV or so, should be considered to have only a qualitative (or, at most, semi-quantitative) value. We recall also that, for elements with intermediate and high atomic numbers, secondary characteristic photons with energies less than the N1-subshell absorption edge are not simulated by PENELOPE. This sets a lower limit to the energy range for which the simulation is faithful.

The source file `penelope.f` (about 11,000 lines of Fortran source code) consists of four blocks of subprograms, namely, preparatory calculations and I/O routines, interaction simulation procedures, numerical routines and transport routines. Only the latter are invoked from the main program. The interaction simulation routines implement the theory and algorithms described in Chapters 2 and 3. Although the interaction routines are not called from the main program, there are good reasons to have them properly identified. Firstly, these are the code pieces to be modified to incorporate better physics (when available) and, secondly, some of these subroutines deliver numerical values of the DCSs (which can be useful to apply certain variance-reduction techniques). To have these routines organised, we have named them according to the following convention:

- The first letter indicates the particle (E for electrons, P for positrons, G for photons).
- The second and third letters denote the interaction mechanism (EL for elastic, IN for inelastic, BR for bremsstrahlung, AN for annihilation, RA for Rayleigh, CO for Compton, PH for photoelectric and PP for pair production).

- The fourth (lowercase) letter indicates the theoretical model used to describe the interactions. This serves to distinguish the default model (denoted by the letter “a”) from alternative models.
- The random sampling routines have four-letter names. Auxiliary routines, which perform specific calculations, have longer names, with the fifth and subsequent letters and/or numbers indicating the kind of calculation (T for total cross section, D for DCS) or action (W for write data on a file, R for read data from a file, I for initialisation of simulation algorithm).

Thus, for instance, subroutine **EELa** simulates elastic collisions of electrons while subroutine **EINaT** computes total (integrated) cross sections for inelastic scattering of electrons. Subroutine **EELd** describes electron elastic collisions by using the numerical DCSs from the ELSEPA database (see Chapter 3).

6.1.1 Database and input material data file

PENELOPE reads the required physical information about each material (which includes tables of physical properties, interaction cross sections, relaxation data, etc.) from the input material data file (identified as **UNIT=IRD** in the code source listing). The material data file is created by means of the auxiliary program **MATERIAL**, which extracts atomic interaction data from the database. This program runs interactively and is self-explanatory. Basic information about the considered material is supplied by the user from the keyboard, in response to prompts from the program. The required information is: 1) chemical composition (*i.e.*, elements present and stoichiometric index of each element), 2) mass density, 3) mean excitation energy and 4) energy and oscillator strength of plasmon excitations. Alternatively, for a set of 280 prepared materials, the program **MATERIAL** can read data directly from the **pdcompos.p06** file (see below).

For compounds and mixtures, the additivity approximation is adopted to define the material cross sections, *i.e.*, the corresponding “molecular” cross section is set equal to the sum of atomic cross sections weighted with the stoichiometric index of the element. Alloys and mixtures are treated as compounds, with stoichiometric indices equal or proportional to the percent number of atoms of the elements.

The PENELOPE database consists of the following 767 ASCII files,

pdatconf.p06 ... Atomic ground-state configurations, ionisation energies (Lederer and Shirley, 1978) and central values, $J_i(p_z = 0)$, of the one-electron shell Compton profiles (Biggs *et al.*, 1975) for the elements, from hydrogen to einsteinium ($Z = 1 - 99$).

pdcompos.p06 ... This file contains composition data, densities and mean excitation energies for 280 materials, adapted from the database of the ESTAR program of Berger (1992). The list of these materials is given in Table 6.1. The first 99 entries are the elements $Z = 1 - 99$, ordered by atomic number Z . Materials 100 to 280 are compounds and mixtures, in alphabetical order. Notice that PENELOPE does not work for elements with atomic number $Z > 99$.

Table 6.1: List of the 280 pre-defined materials included in the `pdcompos.p06` file, with their identifying numbers (adapted from Berger, 1992).

*** ELEMENTS (id. number = atomic number):		
1	Hydrogen	34 Selenium
2	Helium	35 Bromine
3	Lithium	36 Krypton
4	Beryllium	37 Rubidium
5	Boron	38 Strontium
6	Amorphous carbon	39 Yttrium
7	Nitrogen	40 Zirconium
8	Oxygen	41 Niobium
9	Fluorine	42 Molybdenum
10	Neon	43 Technetium
11	Sodium	44 Ruthenium
12	Magnesium	45 Rhodium
13	Aluminum	46 Palladium
14	Silicon	47 Silver
15	Phosphorus	48 Cadmium
16	Sulfur	49 Indium
17	Chlorine	50 Tin
18	Argon	51 Antimony
19	Potassium	52 Tellurium
20	Calcium	53 Iodine
21	Scandium	54 Xenon
22	Titanium	55 Cesium
23	Vanadium	56 Barium
24	Chromium	57 Lanthanum
25	Manganese	58 Cerium
26	Iron	59 Praseodymium
27	Cobalt	60 Neodymium
28	Nickel	61 Promethium
29	Copper	62 Samarium
30	Zinc	63 Europium
31	Gallium	64 Gadolinium
32	Germanium	65 Terbium
33	Arsenic	66 Dysprosium
		67 Holmium
		68 Erbium
		69 Thulium
		70 Ytterbium
		71 Lutetium
		72 Hafnium
		73 Tantalum
		74 Tungsten
		75 Rhenium
		76 Osmium
		77 Iridium
		78 Platinum
		79 Gold
		80 Mercury
		81 Thallium
		82 Lead
		83 Bismuth
		84 Polonium
		85 Astatine
		86 Radon
		87 Francium
		88 Radium
		89 Actinium
		90 Thorium
		91 Protactinium
		92 Uranium
		93 Neptunium
		94 Plutonium
		95 Americium
		96 Curium
		97 Berkelium
		98 Californium
		99 Einsteinium
*** COMPOUNDS AND MIXTURES (in alphabetical order):		
100	Acetone	
101	Acetylene	
102	Adenine	
103	Adipose tissue (ICRP)	
104	Air, dry (near sea level)	
105	Alanine	
106	Aluminum oxide	
107	Amber	
108	Ammonia	
109	Aniline	
110	Anthracene	
111	B-100 bone-equivalent plastic	
112	Bakelite	
113	Barium fluoride	
114	Barium sulfate	
115	Benzene	

116	Beryllium oxide
117	Bismuth germanium oxide
118	Blood (ICRP)
119	Bone, compact (ICRU)
120	Bone, cortical (ICRP)
121	Boron carbide
122	Boron oxide
123	Brain (ICRP)
124	Butane
125	N-butyl alcohol
126	C-552 air-equivalent plastic
127	Cadmium telluride
128	Cadmium tungstate
129	Calcium carbonate
130	Calcium fluoride
131	Calcium oxide
132	Calcium sulfate
133	Calcium tungstate
134	Carbon dioxide
135	Carbon tetrachloride
136	Cellulose acetate, cellophane
137	Cellulose acetate butyrate
138	Cellulose nitrate
139	Ceric sulfate dosimeter solution
140	Cesium fluoride
141	Cesium iodide
142	Chlorobenzene
143	Chloroform
144	Concrete, portland
145	Cyclohexane
146	1,2-dichlorobenzene
147	Dichlorodiethyl ether
148	1,2-dichloroethane
149	Diethyl ether
150	N,n-dimethyl formamide
151	Dimethyl sulfoxide
152	Ethane
153	Ethyl alcohol
154	Ethyl cellulose
155	Ethylene
156	Eye lens (ICRP)
157	Ferric oxide
158	Ferroboration
159	Ferrous oxide
160	Ferrous sulfate dosimeter solution
161	Freon-12
162	Freon-12b2
163	Freon-13
164	Freon-13b1
165	Freon-13i1
166	Gadolinium oxysulfide
167	Gallium arsenide
168	Gel in photographic emulsion
169	Pyrex glass
170	Glass, lead

171	Glass, plate
172	Glucose
173	Glutamine
174	Glycerol
175	Graphite
176	Guanine
177	Gypsum, plaster of Paris
178	N-heptane
179	N-hexane
180	Kapton polyimide film
181	Lanthanum oxybromide
182	Lanthanum oxysulfide
183	Lead oxide
184	Lithium amide
185	Lithium carbonate
186	Lithium fluoride
187	Lithium hydride
188	Lithium iodide
189	Lithium oxide
190	Lithium tetraborate
191	Lung (ICRP)
192	M3 wax
193	Magnesium carbonate
194	Magnesium fluoride
195	Magnesium oxide
196	Magnesium tetraborate
197	Mercuric iodide
198	Methane
199	Methanol
200	Mix d wax
201	Ms20 tissue substitute
202	Muscle, skeletal (ICRP)
203	Muscle, striated (ICRU)
204	Muscle-equivalent liquid, with sucrose
205	Muscle-equivalent liquid, without sucrose
206	Naphthalene
207	Nitrobenzene
208	Nitrous oxide
209	Nylon, du Pont elvamide 8062
210	Nylon, type 6 and type 6/6
211	Nylon, type 6/10
212	Nylon, type 11 (rilsan)
213	Octane, liquid
214	Paraffin wax
215	N-pentane
216	Photographic emulsion
217	Plastic scintillator (vinyltoluene based)
218	Plutonium dioxide
219	Polyacrylonitrile
220	Polycarbonate (makrolon, lexan)
221	Polychlorostyrene
222	Polyethylene
223	Polyethylene terephthalate (mylar)
224	Polymethyl methacrylate (lucite, perspex, plexiglass)
225	Polyoxymethylene

226	Polypropylene
227	Polystyrene
228	Polytetrafluoroethylene (teflon)
229	Polytrifluorochloroethylene
230	Polyvinyl acetate
231	Polyvinyl alcohol
232	Polyvinyl butyral
233	Polyvinyl chloride
234	Polyvinylidene chloride (saran)
235	Polyvinylidene fluoride
236	Polyvinyl pyrrolidone
237	Potassium iodide
238	Potassium oxide
239	Propane
240	Propane, liquid
241	N-propyl alcohol
242	Pyridine
243	Rubber, butyl
244	Rubber, natural
245	Rubber, neoprene
246	Silicon dioxide
247	Silver bromide
248	Silver chloride
249	Silver halides in photographic emulsion
250	Silver iodide
251	Skin (ICRP)
252	Sodium carbonate
253	Sodium iodide
254	Sodium monoxide
255	Sodium nitrate
256	Stilbene
257	Sucrose
258	Terphenyl
259	Testes (ICRP)
260	Tetrachloroethylene
261	Thallium chloride
262	Tissue, soft (ICRP)
263	Tissue, soft (ICRU four-component)
264	Tissue-equivalent gas (methane based)
265	Tissue-equivalent gas (propane based)
266	Tissue-equivalent plastic (A-150)
267	Titanium dioxide
268	Toluene
269	Trichloroethylene
270	Triethyl phosphate
271	Tungsten hexafluoride
272	Uranium dicarbide
273	Uranium monocarbide
274	Uranium oxide
275	Urea
276	Valine
277	Viton fluoroelastomer
278	Water, liquid
279	Water vapor
280	Xylene

pdflist.p06 ... List of materials predefined in file **pdcompos.p06**, with their identification numbers (same information as in Table 6.1).

pdrelax.p06 ... Relaxation data for single-ionised atoms with a vacancy in a K, L or M shell. Each line in this file describes an atomic transition. The quantities listed are the atomic number of the element, the numerical labels of the active electron shells (see Table 6.2), the transition probability and the energy of the emitted x ray or electron, respectively. Transition probabilities and energies were extracted from the LLNL Evaluated Atomic Data Library (Perkins *et al.*, 1991). As the energies given in this database are roughly approximate, they were replaced by more accurate values, when available. Energies of x rays from K- and L-shell transitions were taken from the recent compilation of Deslattes *et al.* (2003). The energies of characteristic M lines are from Bearden's (1967) review.

Table 6.2: Numerical labels used to designate atomic electron shells. In the case of non-radiative transitions, the label 30 indicates shells beyond the N7 shell.

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

99 files named pdeelZZ.p06 with ZZ=atomic number (01–99). These files contain integrated cross sections for elastic scattering of electrons and positrons by neutral atoms, calculated by using the partial-wave methods described in Section 3.1 (Salvat *et al.*, 2005). The first line in each file gives the atomic number ZZ; each subsequent line has 7 columns with the following data:

- 1st column: kinetic energy (eV), in increasing order.
- 2nd column: total cross section for electrons.
- 3rd column: first transport cross section for electrons.
- 4th column: second transport cross section for electrons.
- 5th column: total cross section for positrons.
- 6th column: first transport cross section for positrons.
- 7th column: second transport cross section for positrons.

The grid of energies is approximately logarithmic, with 15 points per decade, and is the same for all elements. All cross sections are in cm^2 .

99 files named eeldxZZZ.p06 with ZZZ=atomic number (001–099). Tables of differential cross sections, and integrated cross sections, for elastic scattering of electrons by free neutral atoms. These files are part of the ELSEPA database (see Section 3.1), and they cover the energy interval from 50 eV to 100 MeV. All cross sections are in cm^2 .

99 files named pelidxZZZ.p06 with ZZZ=atomic number (001–099). Tables of differential cross sections, and integrated cross sections, for elastic scattering of positrons by free neutral atoms. These files are part of the ELSEPA database (see Section 3.1), and they cover the energy interval from 50 eV to 100 MeV. All cross sections are in cm^2 .

99 files named pdebrZZ.p06 with ZZ=atomic number (01–99). They contain the atomic bremsstrahlung scaled cross sections (energy loss spectra) and total integrated radiative cross sections of electrons, for a grid of electron kinetic energies E and reduced photon energies W/E that is dense enough to allow the use of cubic spline log-log interpolation in E and linear interpolation in W/E . The data in these files is from a database, with 32 reduced photon energies, which was provided to the authors by Steve Seltzer (a brief description of the methods used to compute the database and a reduced tabulation is given in Seltzer and Berger, 1986). The format of the bremsstrahlung database files is the following,

1) The first line contains the atomic number ZZ.

2) Each four-line block contains the electron kinetic energy E , the scaled energy-loss differential cross section at the 32 fixed reduced photon energies and the value of the integrated radiative cross section.

Energies are in eV and the values of the scaled energy-loss cross section are in millibarn (10^{-27} cm^2).

pdbrang.p06 ... Gives the parameters of the analytical shape function (angular distribution) of bremsstrahlung photons, which is expressed as a statistical mixture of two Lorentz-boosted dipole distributions, Eq. (3.160). The distribution parameters were obtained by fitting the benchmark partial-wave shapes tabulated by Kissel *et al.* (1983).

99 files named pdgppZZ.p06 with ZZ=atomic number (01–99). Total cross sections for electron-positron pair production by photons with energies up to 100 GeV in the field of neutral atoms. The data were generated by means of the XCOM program of Berger and Hubbell (1987). The first line of each file contains the atomic number ZZ; each subsequent line gives,

1st column: photon energy, in eV. The same energy grid for all elements.

2nd column: total cross section for pair+triplet production in barn (10^{-24} cm^2).

99 files named pdgphZZ.p06 with ZZ=atomic number (01–99), containing photoelectric total atomic cross sections and partial cross sections for photoionisation of

inner shells (K shell, L and M subshells) for the elements and photon energies in the range from 50 eV to 1 TeV. The data were extracted from the LLNL Evaluated Photon Data Library EPDL97 (Cullen *et al.*, 1997). The format is the following,

- 1) The first line contains the atomic number **ZZ** and the number **NS** of shells for which the partial cross section is tabulated.

- 2) Each of the following lines contains a value of the photon energy (in eV) and the corresponding total cross section and partial cross sections of the shells K, L1, L2, L3, M1, M2, M3, M4, and M5, respectively (all cross sections in barn). For low-*Z* elements, L- and M-subshells are empty and, therefore, they do not appear in the table.

The grid of energies for each element was obtained by merging a generic grid (the same for all elements, covering the energy range from 50 eV to 1 TeV) with the grid of absorption edges of the element, and adding additional points (where needed) to ensure that linear log-log interpolation will never introduce relative errors larger than 0.001.

99 files named pdesiZZ.p06 with **ZZ**=atomic number (01–99), containing total (integrated) cross sections for ionisation of inner shells (K shell, L and M subshells) by impact of electrons with kinetic energies in the range from 50 eV to 1 GeV. These cross sections were evaluated by means of the optical-data model described in Section 3.2.6 using photoelectric cross sections read from the files **pdgphZZ.p06**. The format is the following,

- 1) the first line contains the atomic number **ZZ** and the number **NS** of shells for which the partial cross section is tabulated.

- 2) each of the following lines contains a value of the kinetic energy of the projectile (in eV) and the corresponding ionisation cross sections (in cm²) of the shells K, L1, L2, L3, M1, M2, M3, M4 and M5, respectively (all cross sections in barn). For low-*Z* elements, L- and M-subshells are empty and, therefore, they do not appear in the table.

99 files named pdpsiZZ.p06 with **ZZ**=atomic number (01–99), containing total (integrated) cross sections for ionisation of inner shells (K shell, L and M subshells) by impact of positrons with kinetic energies in the range from 50 eV to 1 GeV. These cross sections were evaluated by means of the optical-data model described in Section 3.2.6 using photoelectric cross sections read from the files **pdgphZZ.p06**. The format of these files is the same as that of the **pdesiZZ.p06** files.

Atomic cross sections for coherent and incoherent scattering of photons, inelastic scattering of electrons and positrons, and positron annihilation are evaluated directly from the analytical DCSs described in Chapters 2 and 3.

In the material definition file generated by the program MATERIAL, mean free paths, transport mean free paths and stopping powers of electrons and positrons are given in mass-thickness units (1 mtu $\equiv$ 1 g/cm²) and eV/mtu, respectively. Photon mass attenuation coefficients are expressed in cm²/g. These quantities are practically independent of the material density; the only exception is the collision stopping power for

electrons and positrons with kinetic energies larger than about 0.5 MeV, for which the density-effect correction may be appreciable.

The energy-dependent quantities tabulated in the input material data file determine the most relevant characteristics of the scattering model. For instance, the MW differential cross section for electron and positron elastic scattering is completely defined by the mean free paths and transport mean free paths. Collision and radiative stopping powers read from the input file are used to renormalise the built-in analytical differential cross sections, *i.e.*, these are multiplied by an energy-dependent factor such that the input stopping powers are exactly reproduced. The mean free paths used in the simulation of photon transport are directly obtained from the input total cross sections. Natural cubic spline log-log interpolation is used to interpolate the tabulated energy-dependent quantities, except for the photoelectric attenuation coefficient, which is obtained by simple linear log-log interpolation in the intervals between consecutive absorption edges.

To simulate geometrical structures with several materials, the corresponding material data files generated by the program MATERIAL must be concatenated in a single input file. PENELOPE labels the M-th material in this file with the index **MAT=M**, which is used during the simulation to identify the material where the particle moves. The maximum number of different materials that PENELOPE can handle simultaneously is fixed by the parameter **MAXMAT**, which in the present version is set equal to 10. The required memory storage is roughly proportional to the value of this parameter. The user can increase **MAXMAT** by editing the program source files. Notice that the value of **MAXMAT** *must* be the same in all subprograms.

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 on the screen by running the GNUPLOT script **tables.gnu**. The program **tables.f** runs interactively; after generating the output tables, it can be used to obtain interpolated values of different quantities at energies specified by the user. The quantities that are made accessible in this way are the mean free paths for the various interaction mechanisms (and the corresponding total cross sections), and the particle ranges.

6.1.2 Structure of the main program

As mentioned above, PENELOPE must be complemented with a steering main program, which controls the geometry and the evolution of tracks, keeps score of the relevant quantities and performs the required averages at the end of the simulation.

The connection of PENELOPE and the main program is done via the named common block

```
→ COMMON/TRACK/E,X,Y,Z,U,V,W,WGHT,KPAR,IBODY,MAT,ILB(5)
```

that contains the following particle state variables:

- KPAR ... kind of particle (1: electron, 2: photon, 3: positron).
- E ... current particle energy (eV) (kinetic energy for electrons and positrons).
- X, Y, Z ... position coordinates (cm).
- U, V, W ... direction cosines of the direction of movement.
- WGHT ... this is a dummy variable in analogue simulations. When using variance-reduction methods, the particle weight can be stored here.
- IBODY ... this auxiliary flag serves to identify different bodies in complex material structures.
- MAT ... material where the particle moves (*i.e.*, the one in the body labelled IBODY).
- ILB(5) ... an auxiliary array of 5 labels that describe the origin of secondary particles (see below). It is useful, *e.g.*, to study partial contributions from particles originated by a given process.

The position coordinates $\mathbf{r} = (X, Y, Z)$ and the direction cosines $\hat{\mathbf{d}} = (U, V, W)$ of the direction of movement are referred to a fixed rectangular coordinate system, the “laboratory” system, which can be arbitrarily defined.

NB: During the simulation, all energies and lengths are expressed in eV and cm, respectively.

The label KPAR identifies the kind of particle: KPAR=1, electron; KPAR=2, photon; KPAR=3, positron. A particle that moves in material M is assumed to be absorbed when its energy becomes less than a value EABS(KPAR,M) (in eV) specified by the user. Positrons are assumed to annihilate, by emission of two photons, when absorbed. In dose calculations, EABS(KPAR,M) should be determined so that the residual range of particles with this energy is smaller than the dimensions of the volume bins used to tally the spatial dose distribution. As the interaction database is limited to energies above 50 eV, absorption energies EABS(KPAR,M) must be larger than this value.

The transport algorithm for electrons and positrons in each material M is controlled by the following simulation parameters,

- C1(M) ... Average angular deflection, $C_1 \simeq 1 - \langle \cos \theta \rangle$ [Eq. (4.11)], produced by multiple elastic scattering along a path length equal to the mean free path between consecutive hard elastic events. A value of the order of 0.05 is usually adequate. The maximum allowed value of C1(M) is 0.2.
- C2(M) ... Maximum average fractional energy loss, C_2 [Eq. (4.85)], between consecutive hard elastic events. Usually, a value of the order of 0.05 is adequate. The maximum allowed value of C2(M) is 0.2.
- WCC(M) ... Cutoff energy loss, W_{cc} (in eV), for hard inelastic collisions.
- WCR(M) ... Cutoff energy loss, W_{cr} (in eV), for hard bremsstrahlung emission.

These parameters determine the accuracy and speed of the simulation. To ensure accuracy, $C1(M)$ and $C2(M)$ should have small values (of the order of 0.01 or so). With larger values of $C1(M)$ and $C2(M)$ the simulation becomes faster, at the expense of a certain loss in accuracy. The cutoff energies $WCC(M)$ and $WCR(M)$ mainly influence the simulated energy distributions. The simulation speeds up by using larger cutoff energies, but if these are too large, the simulated energy distributions may be somewhat distorted. In practice, simulated energy distributions are found to be insensitive to the adopted values of $WCC(M)$ and $WCR(M)$ when these are less than the bin width used to tally the energy distributions. Thus, the desired energy resolution determines the maximum allowed cutoff energies. The reliability of the whole simulation rests on a single condition: the number of steps (or random hinges) per primary track must be “statistically sufficient”, *i.e.*, larger than 10 or so.

The simulation package is initialised from the main program with the statement
 $\rightarrow$ CALL PEINIT(EPMAX,NMAT,IRD,IWR,INFO)

Subroutine PEINIT reads the data files of the different materials, evaluates relevant scattering properties and prepares look-up tables of energy-dependent quantities that are used during the simulation. Its input arguments are:

EPMAX ... Maximum energy (in eV) of the simulated particles. Notice that if the primary particles are positrons with initial kinetic energy EP, the maximum energy of annihilation photons may be close to (but less than) $EPMAX = 1.21(EP + m_e c^2)$; in this special case, the maximum energy is larger than the initial kinetic energy.

NMAT ... Number of different materials (less than or equal to MAXMAT).

IRD ... Input unit.

IWR ... Output unit.

INFO ... Determines the amount of information that is written on the output unit. Minimal for INFO=0 and increasingly detailed for INFO=1, 2, etc.

For the preliminary computations, PEINIT needs to know the absorption energies EABS(KPAR,M) and the simulation parameters $C1(M)$, $C2(M)$, $WCC(M)$ and $WCR(M)$. This information is introduced through the named common block

$\rightarrow$ COMMON/CSIMPA/EABS(3,MAXMAT),C1(MAXMAT),C2(MAXMAT),WCC(MAXMAT),
 1 WCR(MAXMAT)

that has to be loaded before invoking subroutine PEINIT. Notice that we can employ different values of the simulation parameters for different materials. This possibility can be used to speed up the simulation in regions of lesser interest.

PENELOPE has been structured in such a way that a particle track is generated as a sequence of track segments (free flights or “jumps”); at the end of each segment, the particle suffers an interaction with the medium (a “knock”) where it loses energy, changes its direction of movement and, in certain cases, produces secondary particles. Electron-photon showers are simulated by successively calling the following subroutines:

SUBROUTINE CLEANS ... Initiates the secondary stack, where initial states of secondary particles are stored.

SUBROUTINE START ... For electrons and positrons, this subroutine forces the following interaction event to be a soft artificial one. It must be called before starting a new –primary or secondary– track and also when a track crosses an interface.

Calling **START** is strictly necessary only for electrons and positrons; for photons this subroutine has no physical effect. However, it is advisable to call **START** for any kind of particle because it checks whether the energy is within the expected range, and can thus help to detect bugs in the main program.

SUBROUTINE JUMP(DSMAX,DS) ... Determines the length **DS** of the track segment to the following interaction event.

The input parameter **DSMAX** defines the maximum allowed step length for electrons/positrons; for photons, it has no effect. As mentioned above, to limit the step length, **PENELOPE** places delta interactions along the particle track. These are fictitious interactions that do not alter the physical state of the particle. Their only effect is to interrupt the sequence of simulation operations (which requires altering the values of inner control variables to permit resuming the simulation in a consistent way). The combined effect of the soft interactions that occur along the step preceding the delta interaction is simulated by the usual random-hinge method.

To ensure the reliability of the mixed simulation algorithm, the number of artificial soft events per particle track in each body should be larger than, say, 10. For relatively thick bodies (say, thicker than 10 times the mean free path between hard interactions), this condition is automatically satisfied. In this case we can switch off the step-length control by setting **DSMAX=1.0D35** (or any other very large value). On the other hand, when the particle moves in a thin body, **DSMAX** should be given a value of the order of one tenth of the “thickness” of that body. Limiting the step length is also necessary to simulate particle transport in external electromagnetic fields.

SUBROUTINE KNOCK(DE,ICOL) ... Simulates an interaction event, computes new energy and direction of movement, and stores the initial states of the generated secondary particles, if any. On output, the arguments are:

DE ... deposited energy in the course of the event (including the energies of generated secondary particles),

ICOL ... kind of event that has been simulated, according to the convention shown in Table 6.3.

Delta interactions (**ICOL=7**) occur not only for electrons and positrons, but also for photons (because of the trick adopted to avoid interpolating the photoelectric inverse mean free path, see Section 2.5). The value **ICOL=8** is used for the “auxiliary” interactions (an additional mechanism that may be defined by the user, *e.g.*, to simulate photonuclear interactions, see the source file **penelope.f**).

SUBROUTINE SECPAR(LEFT) ... Sets the initial state of a secondary particle and removes it from the secondary stack. The output value **LEFT** is the number of secondary particles that remained in the stack at the calling time.

Table 6.3: Code number (ICOL) for the various interaction events.

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

SUBROUTINE STORES(E,X,Y,Z,U,V,W,WGHT,KPAR,ILB) ... Stores a particle in the secondary stack. Arguments have the same meaning as in **COMMON/TRACK/**, but refer to the particle that is being stored. The variables **IBODY** and **MAT** are set equal to the current values in **COMMON/TRACK/**.

Calling **STORES** from the main program is useful, *e.g.*, to store particles produced by splitting, a variance-reduction method (see Section 1.6.2).

The sequence of calls to generate a random track is independent of the kind of particle that is being simulated. The generation of random showers proceeds as follows (see Fig. 6.1):

- (i) Set the initial state of the primary particle, *i.e.*, assign values to the state variables: kind of particle **KPAR**, energy **E**, position coordinates $\mathbf{r}=(\mathbf{X},\mathbf{Y},\mathbf{Z})$ and direction of movement $\hat{\mathbf{d}}=(\mathbf{U},\mathbf{V},\mathbf{W})$. Specify the body and material where the particle moves by defining the values of **IBODY** and **MAT**, respectively. Optionally, set the values of **WGHT** and **ILB(1:5)**.
- (ii) **CALL CLEANS** to initialise the secondary stack.
- (iii) **CALL START** to initiate the simulation of the track.
- (iv) **CALL JUMP(DSMAX,DS)** to determine the length **DS** of the next track segment (for electrons and positrons, **DS** will never exceed the input value **DSMAX**).
- (v) Compute the position of the following event:

- If the track has crossed an interface, stop the particle at the position where the track intersects the interface, and shorten the step length DS accordingly. Change to the new material (the one behind the interface) by redefining the variables $IBODY$ and MAT .

When the particle escapes from the system, the simulation of the track has been finished; increment counters and go to step (vii).

Go to step (iii).

(vi) `CALL KNOCK(DE,ICOL)` to simulate the following event.

- If the energy of the particle is less than $EABS(KPAR,MAT)$, end the track, increment counters and go to step (vii).
- Go to step (iv).

(vii) `CALL SECPAR(LEFT)` to start the track of a particle in the secondary stack (this particle is then automatically removed from the stack).

- If $LEFT > 0$, go to step (iii). The initial state of a secondary particle has already been set.
- If $LEFT = 0$, the simulation of the shower produced by the primary particle has been completed. Go to step (i) to generate a new primary particle (or leave the simulation loop after simulating a sufficiently large number of showers).

Quantities of interest must be tallied by the main program, immediately after the occurrence of the relevant events. The required information is delivered by the simulation routines, either as output arguments or through the variables in common `TRACK`.

Notice that subroutines `JUMP` and `KNOCK` keep the position coordinates unaltered; the positions of successive events have to be followed by the main program (simply by performing a displacement of length DS along the direction of movement after each call to `JUMP`). The energy of the particle is automatically reduced by subroutine `KNOCK`, after generating the energy loss from the relevant probability distribution. `KNOCK` also modifies the direction of movement according to the scattering angles of the simulated event. Thus, at the output of `KNOCK`, the values of the energy E , the position $\mathbf{r} = (X,Y,Z)$ and the direction of movement $\hat{\mathbf{d}} = (U,V,W)$ define the particle state immediately after the interaction event.

The subroutine package `PENGEOM` (see Chapter 5) can be used to handle arbitrary quadric geometries. The flow diagram of a main program for `PENELOPE` and `PENGEOM` is shown in Fig. 6.2. As `PENGEOM` performs all geometrical operations automatically, the structure of the main program is relatively simple (cf. Figs. 6.1 and 6.2).

In order to avoid problems arising from possible overflows of the secondary stack, when a secondary particle is produced its energy is temporarily assumed as locally deposited. Hence, the energy E of a secondary must be subtracted from the corresponding dose counter when the secondary track is started. Occasional overflows of the secondary stack are remedied by eliminating the less energetic secondary electron or photon in the

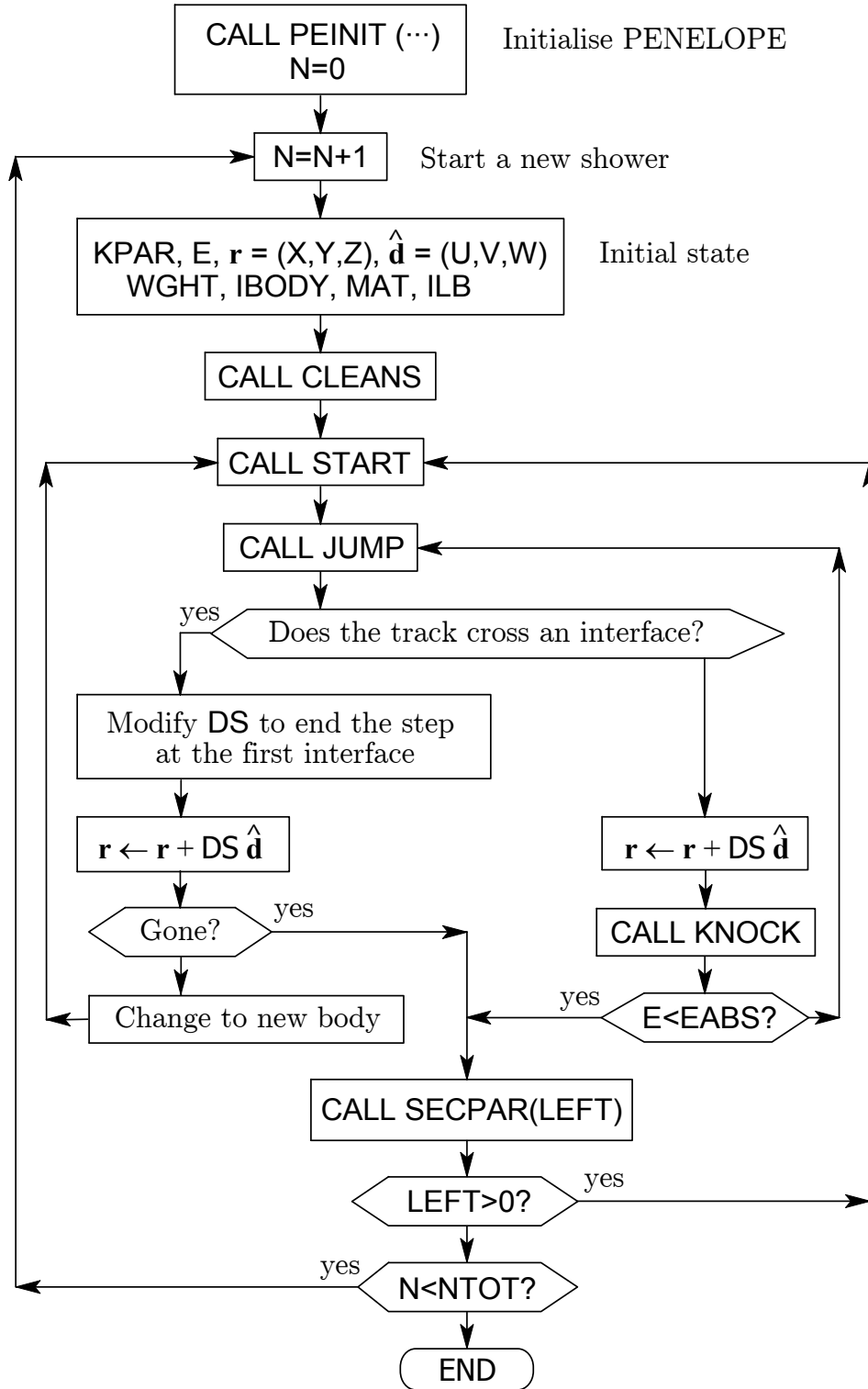


Figure 6.1: Flow diagram of the main program for simulating electron-photon showers with PENELOPE.

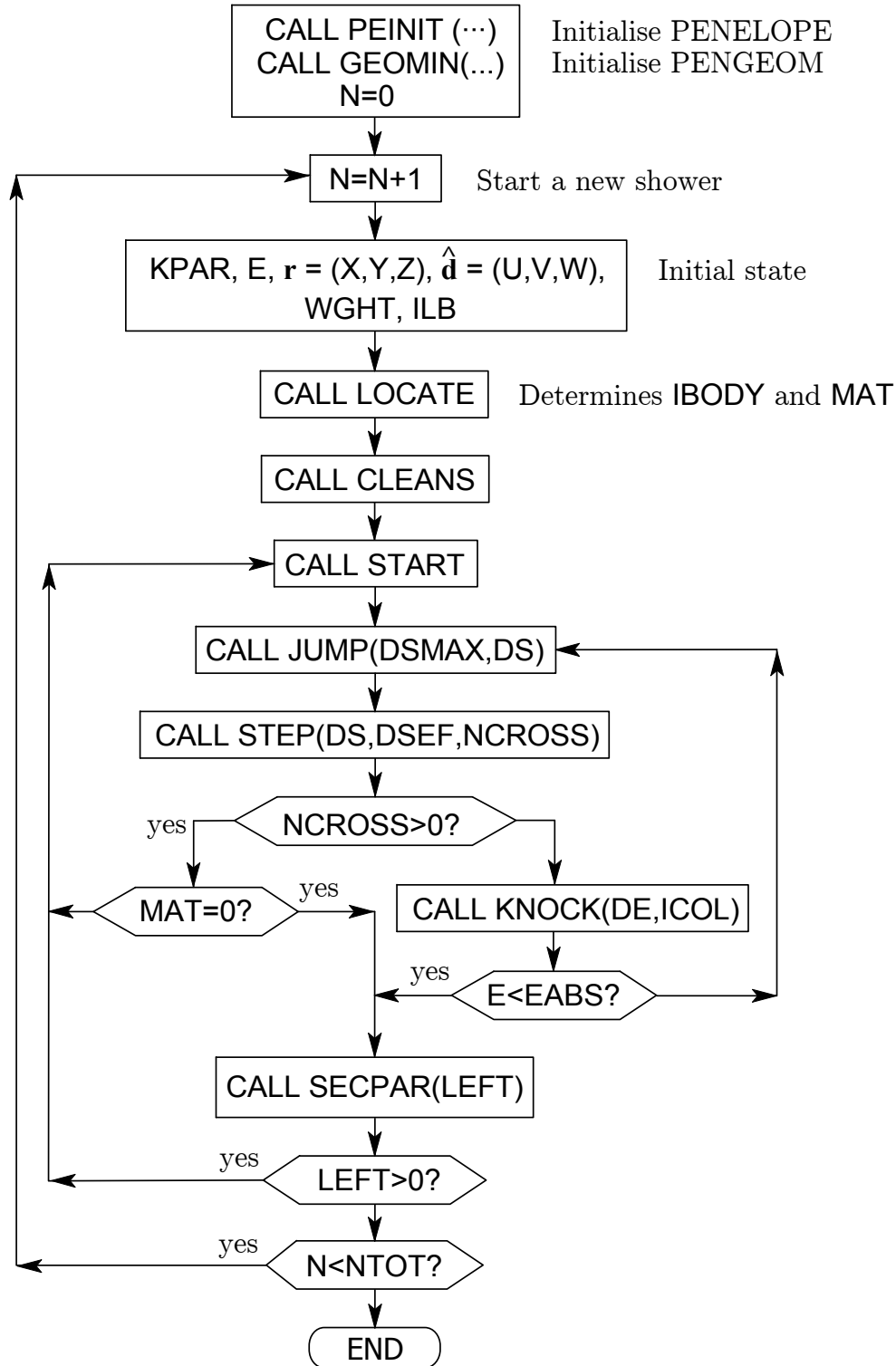


Figure 6.2: Flow diagram of the main program for simulating electron-photon showers in quadric geometries with PENELOPE and PENGEOm.

stack (positrons are not eliminated since they will eventually produce quite energetic annihilation radiation). As the main effect of secondary particles is to spread out the energy deposited by the primary particle, the elimination of the less energetic secondary electrons and photons should not invalidate local dose calculations.

It is the responsibility of the user to avoid calling subroutines `JUMP` and `KNOCK` with energies outside the interval `(EABS(KPAR,M),EPMAX)`. This could cause improper interpolation of the cross sections. The simulation is aborted (and an error message is printed in unit 26) if the conditions `EABS(KPAR)<E<EPMAX` are not satisfied when a primary or secondary track is started (whenever subroutine `START` is called at the beginning of the track).

Pseudo-random numbers uniformly distributed in the interval `(0,1)` are supplied by function `RAND(DUMMY)`, which implements a 32-bit generator due to L'Ecuyer (see Table 1.1). The seeds of the generator (two integers) are transferred from the main program through the named common block `RSEED` (see below). The random number generator can be changed by merely replacing that `FUNCTION` subprogram (the new one has to have a single dummy argument). Some compilers incorporate an intrinsic random-number generator with the same name (but with different argument lists). To avoid conflict, `RAND` should be declared as an external function in all subprograms that call it.

Notice that

- (1) In the simulation routines, real and integer variables are declared as `DOUBLE PRECISION` and `INTEGER*4`, respectively. To prevent type mismatches, the statement

$$\rightarrow \text{IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER*4 (I-N)}$$
is used in all PENELOPE routines and in the example main programs.
- (2) The main program *must* include the following three common blocks:

$$\rightarrow \text{COMMON/TRACK/E,X,Y,Z,U,V,W,WGHT,KPAR,IBODY,MAT,ILB(5)}$$

$$\rightarrow \text{COMMON/CSIMPA/EABS(3,MAXMAT),C1(MAXMAT),C2(MAXMAT),WCC(MAXMAT),}$$

$$\quad 1 \text{ WCR(MAXMAT)} \quad \quad \quad \text{! Simulation parameters.}$$

$$\rightarrow \text{COMMON/RSEED/ISEED1,ISEED2} \quad \text{! Random-number generator seeds.}$$
- (3) The subroutines in PENELOPE and PENGEOm may print warning and error messages in the unit 26, which is also the default output unit in the example main programs `penslab`, `pencyl` and `penmain` described in Section 6.2.

As mentioned above, `ILB(5)` is an array of labels that describe the origin of secondary particles. It is assumed that the user has set `ILB(1)` equal to 1 (one) when a primary (source) particle history is initiated. PENELOPE then assigns the following labels to each particle in a shower;

`ILB(1)`: generation of the particle; 1 for primary particles, 2 for their direct descendants, etc.

`ILB(2)`: kind `KPAR` of parent particle, only if `ILB(1)>1` (secondary particles).

ILB(3): interaction mechanism ICOL (see above) that originated the particle, only when ILB(1) > 1.

ILB(4): a non-zero value identifies particles emitted from atomic relaxation events and describes the atomic transition where the particle was released. The numerical value is $= Z \cdot 10^6 + \text{IS1} \cdot 10^4 + \text{IS2} \cdot 100 + \text{IS3}$, where Z is the atomic number of the parent atom and IS1, IS2 and IS3 are the labels of the active atomic electron shells (see Table 6.2).

ILB(5): this label can be defined by the user; it is transferred to all descendants of the particle.

The ILB label values are delivered by subroutine SECPAR, through common TRACK, and remain unaltered during the simulation of the track.

Owing to the long execution time, the code will usually be run in batch mode. It is advisable to limit the simulation time rather than the number of tracks to be simulated, because the time required to follow each track is difficult to predict. To this end, one can link a clock routine to the simulation code and stop the computation after exhausting the allotted time; an example of a clock routine (which runs in any Fortran 95 compiler) is included in the PENELOPE distribution package.

6.1.3 Variance reduction

The subroutine package `penelope.f` is intended to perform analogue simulation and, therefore, does not include any variance-reduction methods. The source file `penvared.f` contains subroutines to automatically perform particle splitting, Russian roulette, and interaction forcing. Splitting and Russian roulette (see Section 1.6.2) do not require changes in PENELOPE; the necessary manipulations on the numbers and statistical weights WGHT of particles could be done directly in the main program. Particles resulting from splitting can be stored in the secondary stack by calling subroutine STORES. Interaction forcing (Section 1.6.1) implies changing the mean free paths of the forced interactions and, at the same time, redefining the weights of the generated secondary particles. In principle, it is possible to apply interaction forcing from the main program by manipulating the interaction probabilities, which are made available through the named common block CJUMP0, but the process is by no means obvious.

The package `penvared.f` consists of the following subroutines;

SUBROUTINE VSPLIT(NSPLIT)

Splits the current particle into NSPLIT (> 1) identical particles, defines their weights appropriately, and stores NSPLIT – 1 of them into the secondary stack. The current particle continues with a reduced statistical weight.

SUBROUTINE VKILL(PKILL)

Applies the Russian roulette technique. The particle is killed with probability PKILL ($0 < \text{PKILL} < 1$); if it survives, its weight is increased by a factor $1/(1 - \text{PKILL})$.

SUBROUTINE JUMPF(DSMAX,DS) and SUBROUTINE KNOCKF(DEF,ICOL)

These two subroutines perform interaction forcing. Their action is to artificially insert “forced” interactions of selected kinds randomly along the particle trajectory. This is accomplished by replacing the inverse mean free path Σ_h [Eq. (4.80)] by a larger value, $\text{FORCE}(\cdot) \times \Sigma_h$. The forcing factors $\text{FORCE}(\text{IBODY}, \text{KPAR}, \text{ICOL})$ [for all the bodies in the material structure, for the three types of particles, and for the various kinds of interactions (see Table 6.3)] have to be specified by the user in the main program and transferred through the common block `COMMON/CFORCE/FORCE(NB,3,7)` with $\text{NB} = 5,000$.

Forcing factors must be larger than, or equal to, unity; obviously, the value $\text{FORCE}(\cdot) = 1$ means “no forcing”. To keep the simulation unbiased, interactions are allowed to affect the state of the projectile only with probability $\text{WFORCE}(\cdot) = 1/\text{FORCE}(\cdot)$, and, at the same time, secondary particles generated in the interactions are assigned a weight smaller than that of the projectile by a factor $= \text{WFORCE}(\cdot)$.

To apply interaction forcing, the main program must call subroutines JUMPF and KNOCKF instead of the usual subroutines JUMP and KNOCK. Moreover, subroutine *START must be called* before starting a track and after each interface crossing, even for photons. The output argument DEF of subroutine KNOCKF is the effective deposited energy (*i.e.*, the deposited energy multiplied by $\text{WFORCE}(\cdot)$).

Usually, the variance-reduction parameters [*i.e.*, the splitting number NSPLIT, the killing probability PKILL, and the forcing factors $\text{FORCE}(\cdot)$] are assumed to be independent of the particle energy. Although this scheme is flexible enough for many practical uses, the parameter values can also be varied during the simulation. NSPLIT and PKILL can be modified at any time during the simulation of a particle. The values of the forcing factors $\text{FORCE}(\cdot)$ can be changed only immediately after a call to subroutine *START*.

Although these variance-reduction routines operate automatically and are robust, they should be invoked with care. Their effect on the efficiency of the simulation is 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!). In general, particle weights must be prevented from reaching very large or very small values. In the first case, a very “heavy” particle can completely hide the information collected from many lighter particles. Conversely, it is not convenient to spend time simulating particles with very small weights, which contribute insignificant amounts to the scores. Therefore, these variance-reduction techniques should be applied only when the particle weight WGHT is within a limited range (weight window), *e.g.*, between 0.05 and 20.

Notice also that repeated splitting and interaction forcing may easily lead to saturation of the secondary stack. The stack size in the original files is $\text{NMS} = 1,000$ particles, and is large enough for most applications. However, it may be insufficient when using heavy splitting and/or interaction forcing. The user should make sure that the secondary stack does not overflow; when the stack gets full, a warning message is printed in unit 26, but the simulation is not aborted (because occasional stack overflows may

not have an appreciable effect on the results). Saturation of the stack can be avoided by increasing the lower limit of the weight window. It is also recommended to apply interaction forcing only to primary particles. In case of need, the size of the stack can be increased by editing the files `penelope.f` and `penvared.f` and changing the value of the parameter `NMS`.

It should be mentioned that, when interaction forcing is applied, energy is conserved only on average because the energy deposited into the medium is, in general, different from the energy lost by the projectile. This implies that the spectrum of deposited energy (of interest, *e.g.*, to simulate the response of scintillation detectors) is distorted. Nevertheless, interaction forcing does increase the efficiency of calculations of dose distributions and secondary radiation emission. Splitting and Russian roulette leave the simulation results unbiased. In particular, Russian roulette is the alternative of choice in front of range rejection and similar *ad hoc* techniques, which may easily bias the results.

6.2 Examples of main programs

The distribution package of PENELOPE includes three examples of main programs: `penslab` (which simulates electron-photon transport in a slab), `pencyl` (for transport in cylindrical geometries), and `penmain` (for generic quadric geometries). In these examples, we assume that a single kind of particles is emitted from the radiation source. The programs can be easily generalised to the case of multi-particle sources with continuous (or discrete) energy spectra. They provide fairly detailed information, but their capabilities are limited. It should be borne in mind that the example main programs are just this, examples. While it is feasible to design simulation algorithms to faithfully implement the adopted interaction models (*i.e.*, the physics) and to describe accurately certain types of geometries (*e.g.*, quadric geometries), it is virtually impossible to develop a generic main program that is able to compute *all* conceivable information about any practical problem (with arbitrary radiation sources). If the example programs included in the PENELOPE distribution package cannot accommodate your problem, you should consider writing your own main program.

The three example main programs operate in a similar way. They all read data from a corresponding input file and output the results in a number of files with fixed names¹. The input files of the three programs have similar structures and formats; they are described in detail in the following Sections². Each line in the 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 programs (which are case sensitive!). Notice also that the order of the data lines is important. The keyword “-----” (6 blanks, which we

¹Warning: The programs overwrite older output files that are left in the working directory. You should save all result files on a separate directory before rerunning a program.

²Although the input files of the three programs are similar, they will be described separately. Users interested in one of the programs, will find all the relevant information in the corresponding Section.

have denoted by “-”) indicates comment lines and can be placed anywhere in the file (except within the geometry definition block of `pencyl`). The programs ignore any text following the first blank after the last numerical datum, or after the character string, in each line (thus, in the input file layouts given below, the comments in square brackets are ignored). Lines with certain keywords (*e.g.*, “SPECTR”) can appear an arbitrary number of times, limited only by the allocated amount of memory. The programs assign default values to many input variables; lines that declare default values may be removed from the input file. The execution of a 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).

Each of the example main programs generates an output file with generic information, such as number of simulated showers, simulation speed, average numbers of generated secondary particles, average deposited energies, etc. The programs also generate a number of continuous distributions (depending on the options selected in the input file). Each simulated 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 programs compute and deliver the statistical uncertainties (3σ) of all evaluated quantities and distributions. Many authors quote these uncertainties as one standard deviation, which means that the probability for the actual value to lie outside the error bar is 0.317. We prefer to be more conservative and stay at the 3σ level, for which the probability of “missing the real value” is only 0.003.

Optionally, at the end of a run, each of the example main programs 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 programs 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.

6.2.1 Program penslab

The program `penslab` simulates electron/photon showers within a material slab (see Fig. 6.3). It illustrates the use of the simulation routines for the simplest geometry (as geometry operations are very simple, this program is faster than those described below). The slab is limited by the planes $z = 0$ and $z = t$, the thickness. The lateral extension of the slab is assumed to be infinite, *i.e.*, much larger than the maximum range of the particles. Primary particles of a given kind, `KPARP`, are emitted from a point source, either with fixed energy E_0 or with a specified (histogram-like) energy spectrum. The initial direction of the primary particles is sampled uniformly within a cone of semi-aperture α and with its central axis in the direction (θ_0, ϕ_0) . Thus, $\alpha = 0$ defines a

monodirectional source and $\alpha = 180$ deg corresponds to an isotropic source. Notice that when $\theta = 0$ or 180 deg, the complete arrangement has rotational invariance about the z -axis.

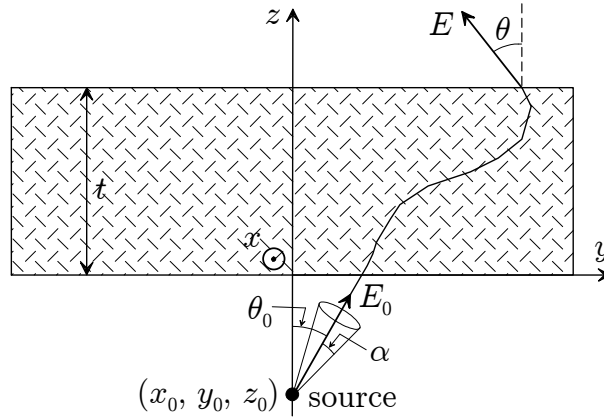


Figure 6.3: General planar geometry considered in **penslab**.

The program **penslab** generates detailed information on many quantities and distributions of physical interest. The output files contain a self-explanatory report of the simulation results, which includes:

- (i) Global results, such as fractions of primary particles that are transmitted, backscattered and absorbed, and a number of average quantities (track length within the sample; number of events of each kind per particle; energy, direction and lateral displacement of particles that leave the sample, etc.).
- (ii) Energy distributions of transmitted and backscattered primary particles.
- (iii) Angular distributions of transmitted and backscattered particles.
- (iv) Depth-dose distribution (*i.e.*, deposited energy per unit depth).
- (v) Depth-distribution of deposited charge.
- (vi) Distribution of energy deposited into the slab.
- (vii) Path-length distribution of primary particles.

The program **penslab** and its predecessors have been intensively used to analyse the reliability of **PENELOPE**. They have been applied to a variety of experimental situations, covering a wide energy range. Benchmark comparisons with experimental data have been published elsewhere (Baró *et al.*, 1995; Sempau *et al.*, 1997; Sempau *et al.*, 2003).

6.2.1.1 Structure of the input file

The structure of the penslab 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.
      >>>>>>> Source definition.
SKPAR  KPARP      [Primary particles: 1=electron, 2=photon, 3=positron]
SENERG  SE0              [Initial energy (monoenergetic sources only)]
SPECTR  Ei,Pi          [E bin: lower-end and total probability]
SPOSIT  SX0,SY0,SZ0      [Coordinates of the source centre]
SDIREC  STHETA,SPHI      [Beam axis direction angles, in deg]
SAPERT  SALPHA          [Beam aperture, in deg]
.
      >>>>>>> Material data and simulation parameters.
SIMPAR  EABS(1:3),C1,C2,WCC,WCR      [Simulation parameters]
PFNAME  mat-filename.ext      [Material definition file, 20 characters]
      >>>>>>> Geometry definition.
THICKN  THICK              [Slab thickness]
DSMAX   DSMAX              [Maximum step length]
.
      >>>>>>> Interaction forcing.
IFORCE  KPAR,ICOL,FORCER,WLOW,WHIG      [Interaction forcing]
      >>>>>>> Counter array dimensions and pdf ranges.
NBE     EMIN,EMAX,NBE      [Energy interval and no. of E-bins]
NBTH    NBTH              [No. of bins for the polar angle THETA]
NBZ     NBZ              [No. of bins for the Z-coordinate]
NBTL    TLMIN,TLMAX,NBTL   [Track-length interval and no. of TL-bins]
.
      >>>>>>> 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 list 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 wrong 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.

• Source definition

SKPAR ... Kind of primary particle (1 = electrons, 2 = photons, or 3 = positrons).
 -- Default: KPARP=1

SENERG ... For monoenergetic sources: initial energy SE0 of primary particles.
 -- Default: SE0=1.0E6

SPECTR ... For a source with continuous (stepwise constant) energy spectrum, each “SPECTR” line gives the lower end-point (Ei) of an energy bin of the spectrum and the associated relative probability (Pi), integrated over the bin. Up to NSEM = 200 lines can be entered, in arbitrary order. The upper end of the spectrum is defined by entering a line with Ei equal to the upper energy end point, and with a negative Pi value.
 -- Default: none

SPOSIT ... Coordinates (x_0 , y_0 , z_0) of the point source.
 -- Defaults: SX0=SY0=0.0, SZ0=-1.0E15

SDIREC ... Polar angle θ_0 and azimuthal angle ϕ_0 of the source beam axis direction, in deg.
 -- Defaults: STHETA=0.0, SPHI=0.0

NOTE: The angular distribution of emerging particles depends on both the polar angle θ and the azimuthal angle ϕ . The program PENSLAB generates only the PDF of θ , that is, the angular distribution averaged over ϕ . This averaged distribution represents the true distribution only when the source is symmetrical about the z axis (*i.e.*, when the incident beam axis is perpendicular to the surface of the slab).

SAPERT ... angular aperture α of the source beam, in deg.
 -- Default: SALPHA=0.0

• Material data and simulation parameters

SIMPAR ... Simulation parameters; absorption energies, EABS(1:3), elastic-scattering parameters, C1 and C2, and cutoff energy losses for inelastic collisions and bremsstrahlung emission, WCC and WCR, respectively.
 -- Defaults: EABS(1)=EABS(3)=0.01*EPMAX, EABS(2)=0.001*EPMAX
 C1=C2=0.1, WCC=EABS(1), WCR=EABS(2)

EPMAX is the maximum energy of all particles found in the simulation (depends on the source energies).

PFNAME ... Name of PENELOPE’s input material data file (up to 20 characters).
 -- Default: none

THICKN ... Slab thickness, in cm.
 -- Default: THICK=1.0E0

DSMAX ... Maximum step length for electrons and positrons, in cm.
 -- Default: DSMAX=THICK/10.0

• Interaction forcing

IFORCE ... Activates forcing of interactions of type **ICOL** of particles **KPAR**. **FORCER** is the forcing factor, $\mathcal{F}$, which must be larger than unity (see Section 6.1.3). The values **WLOW** and **WHIG** are the lower and upper 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 that kind of particle with $E = \text{EPMAX}$. 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**=**EPMAX**, **NBE**=200

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

-- **Default**: **NBTH**=180

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

NBZ ... Number of bins for the z -coordinate.

-- **Default**: **NBZ**=200

NBTL ... Limits of the interval where track-length distributions of primary particles are tallied. Number of track-length bins.

-- **Defaults**: **TLMIN**=0.0, **TLMAX**=2.5***RANGE**(**EPMAX**,**KPARP**), **NBTL**=200

• 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 one used to generate the dump file.

-- **Default**: off

DUMPTO ... Generate the dump file **dumpfile2.dat** (name given by the user, up to 20 characters) after completing the simulation run. This allows the simulation to be resumed later on 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

```

A detailed report of the simulation, including relevant input data as well as some partial results are written in a file named **penslab.dat**. The calculated continuous distributions (histograms) are written in separate files, whose names start with “**ps-**” (for PenSlab) and have the extension “.dat”. These files are in a format suited for direct visualisation with GNUPLOT (version 4.0).

6.2.2 Program pencyl

The program **pencyl** simulates electron and photon transport in multilayered cylindrical structures. The material system consists of one or several layers of given thicknesses. Each layer contains a number of concentric homogeneous rings of given compositions and radii (and thickness equal to that of the layer). The layers are perpendicular to the z -axis and the centre of the rings in each layer is specified by giving its x and y coordinates. When all the centres are on the z -axis, the geometrical structure is symmetrical under rotations about the z -axis (see Fig. 6.4).

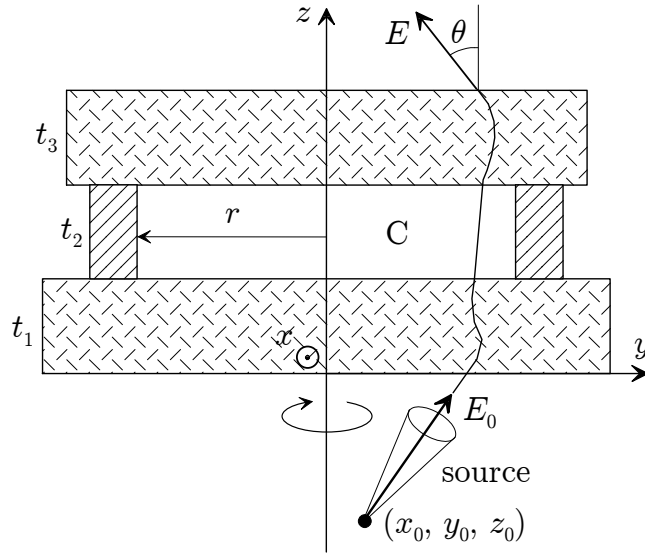


Figure 6.4: An example of cylindrical geometry, a cavity (C) with walls, with a point off-axis source. In this case, the material structure is symmetrical about the z -axis, but the radiation flux and other three-dimensional quantities (*e.g.*, absorbed dose and deposited charge distributions) depend on the azimuthal angle ϕ .

Primary particles of a given kind, KPARP, are emitted from the active volume of the source, either with fixed energy SE0 or with a specified (histogram-like) energy spectrum. The initial direction of the primary particle is sampled uniformly inside a cone of semi-aperture α and with central axis in the direction (θ_0, ϕ_0) .

The program can simulate two different types of sources:

- a) An external point or volume (cylindrical) homogeneous source, defined separately from the geometry of the material system, with its centre at the point (x_0, y_0, z_0) . The initial position of each primary particle is sampled uniformly within the volume of the source. Notice that when $x_0 = 0$, $y_0 = 0$ and $\theta_0 = 0$ or 180 deg, the source is axially symmetrical about the z -axis.
- b) A set of internal sources spread over specified bodies (cylinders or rings), each one with uniform activity density (=number of emissions per unit volume). The original position of the primary particle is sampled uniformly within the volume of the active body. In the case of several active bodies, these are selected randomly with probabilities proportional to the corresponding total activities.

In the distributed form of the program, we assume that both the source and the material structure are symmetrical about the z -axis, because this eliminates the dependence on the azimuthal angle ϕ . The program takes advantage of this symmetry to tally 3D dose distributions. It is possible to consider geometries that are not axially symmetrical, but then the program only delivers values averaged over ϕ . To obtain the dependence of the angular distributions on the azimuthal angle, we need to increase the value of the parameter NBPHM (the maximum number of bins for ϕ , which is set equal to 1 in the distributed source file) and, in the input data file, set NBPH equal to NBPHM.

The source file `pencyl.f` includes a (self-contained) set of geometry routines for tracking particles through multilayered cylindrical structures. These routines can be used for simulation even when the source is off-axis. Cylindrical geometries can be viewed with the program GVIEWC, which is similar to GVIEW2D (see Section 5.6) and runs under Microsoft Windows. This program reads the geometry definition list from the input file and displays a two-dimensional map of the materials intersected by the window plane. It is useful for debugging the geometry definition.

The program `pencyl` delivers detailed information on the transport and energy deposition, which includes energy and angular distributions of emerging particles, depth-dose distribution, depth-distribution of deposited charge, distributions of deposited energy in selected materials and 2D (depth-radius) dose and deposited charge distributions in selected bodies (cylinders). A report on the global simulation, which contains relevant input data as well as some partial results, is written in a file named `pencyl.dat`. The calculated continuous distributions (histograms) are written in separate files, whose names start with “pc-” (for PenCyl) and have the extension “.dat”. These files are in a format suited for direct visualisation with GNU PLOT (version 4.0). `pencyl` can be readily used to study radiation transport in a wide variety of practical systems, *e.g.*, planar ionisation chambers, cylindrical scintillation detectors, solid-state detectors and multilayered structures.

6.2.2.1 Structure of the input file

The structure of the pencyl 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.
GSTART >>>>>>> Beginning of the geometry definition list.
LAYER  ZLOW,ZHIG                                [Z_lower and Z_higher]
CENTRE XCEN,YCEN                                [X_centre and Y_centre]
CYLIND M,RIN,ROUT                                [Material, R_inner and R_outer]
GEND   <<<<<<<< End of the geometry definition list.
.
>>>>>>>> Source definition.
SKPAR  KPARP      [Primary particles: 1=electron, 2=photon, 3=positron]
SENERG SEO        [Initial energy (monoenergetic sources only)]
SPECTR Ei,Pi      [E bin: lower-end and total probability]
SEXTND KL,KC,RELAC [Extended source in KL,KC and rel. activity conc.]
STHICK STHICK      [Source height]
SRADII SRIN,SROUT [Source inner and outer radii]
SPOSIT SX0,SY0,SZO [Coordinates of the source centre]
SDIREC STHETA,SPHI [Beam axis direction angles, in deg]
SAPERT SALPHA      [Beam aperture, in deg]
.
>>>>>>>> Material data and simulation parameters.
NMAT   NMAT        [Number of different materials, .le.10]
SIMPAR M,EABS(1:3,M),C1,C2,WCC,WCR [Sim. parameters for material M]
PFNAME mat-filename.ext [Material definition file, 20 characters]
.
>>>>>>>> Interaction forcing.
IFORCE KL,KC,KPAR,ICOL,FORCER,WLOW,WHIG [Interaction forcing]
>>>>>>>> Counter array dimensions and pdf ranges.
NBE    EMIN,EMAX,NBE [Energy interval and no. of E-bins]
NBTH   NBTH          [No. of bins for the polar angle THETA]
NBPH   NBPH          [No. of bins for the azimuthal angle PHI]
NBZ    NBZ           [No. of bins for the Z-coordinate]
NBR    NBR           [No. of radial bins]
NBTL   TLMIN,TLMAX,NBTL [Track-length interval and no. of TL-bins]
>>>>>>>> Additional distributions to be tallied.
ABSEN  MAT          [Tally the distr. of absorbed E in material MAT]
DOSE2D KL,KC,NZ,NR  [Tally 2D dose and charge dists. in body KL,KC]
.
>>>>>>>> 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 wrong 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.

• Geometry definition

Geometry definition list ... begins with the line “**GSTART**” and ends with the line “**GEND--**” (notice the two blanks). The only allowed keywords in the geometry list are “**GSTART**”, “**LAYER-**”, “**CENTRE**”, “**CYLIND**” and “**GEND--**”. The line after “**GSTART**” must be a “**LAYER-**” line. Each “**LAYER-**” line contains the z -coordinates of its lower and upper limiting planes and is followed by a “**CENTRE**” line (optional) and by one or several “**CYLIND**” lines, which contain the inner and outer radii of the various concentric rings in the layer; empty layers are disregarded.

Layers must be defined in increasing order of heights, from bottom to top of the structure. If the “**CENTRE**” line is not entered, cylinders are assumed to be centred on the z -axis ($XCEN = YCEN = 0.0$). Cylinders have to be defined in increasing radial order, from the centre to the periphery. The two lengths in each “**LAYER-**” and “**CYLIND**” line must be entered in increasing order. The geometry definition list can be debugged/visualised with the code **GVIEWC** (operable under Microsoft Windows). Notice that **GVIEWC** reads the geometry directly from the **pencyl** input data file (*i.e.*, the first line in the geometry definition file must be the “**TITLE-**” line).

-- Default: none (the geometry must be defined in the input file)

• Source definition

SKPAR ... Kind of primary particle (1 = electrons, 2 = photons, or 3 = positrons).

-- Default: **KPARP=1**

SENERG ... For monoenergetic sources: initial energy **SE0** of primary particles.

-- Default: **SE0=1.0E6**

SPECTR ... For a source with continuous (stepwise constant) energy spectrum, each “**SPECTR**” line gives the lower end-point (**Ei**) of an energy bin of the spectrum and the associated relative probability (**Pi**), integrated over the bin. Up to **NSEM** = 200

lines can be entered, in arbitrary order. The upper end of the spectrum is defined by entering a line with **Ei** equal to the upper energy end point, and with a negative **Pi** value.

-- Default: none

SEXTND ... For internal extended sources, this line defines an active body **KL**, **KC** (the cylinder **KC** in layer **KL**) and its relative activity density, **RELAC**.

-- Default: none

NOTE: The labels **KL**, **KC** that identify each body are defined by the ordering in the input geometry list. These labels are written on the output geometry report.

STHICK ... For an external source, thickness (height) of the active volume of the source (cylinder or ring).

-- Default: **STHICK=0.0**

SRADII ... For an external source, inner and outer radii of the active volume of the source.

-- Defaults: **SRIN=0.0**, **SROUT=0.0**

SPOSIT ... For an external source, coordinates (x_0 , y_0 , z_0) of the centre of the source volume.

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

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

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

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

-- Default: **SALPHA=0.0**

• Material data and simulation parameters

NMAT ... Number of different materials (up to 10 with the original program dimensions). Materials are identified by their ordering in PENELOPE's input material data file.

-- Default: **NMAT=1**

SIMPAR ... Set of simulation parameters for the **M**-th 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)**. One line for each material.

-- Defaults: **EABS(1,M)=EABS(3,M)=0.01*EPMAX**, **EABS(2,M)=0.001*EPMAX**
C1(M)=C2(M)=0.1, **WCC(M)=EABS(1,M)**, **WCR(M)=EABS(2,M)**

EPMAX is the maximum energy of all particles found in the simulation (depends on the source energies).

PFNAME ... Name of PENELOPE's input material data file (up to 20 characters).

-- Default: none

• Interaction forcing

IFORCE ... Activates forcing of interactions of type **ICOL** of particles **KPAR** in cylinder **KC** of layer **KL**. **FORCER** is the forcing factor, $\mathcal{F}$, which must be larger than unity (see Section 6.1.3). The values **WLOW**, **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 $E = \text{EPMAX}$. 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**=**EPMAX**, **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 material system moving upwards ($W > 0$) and downwards ($W < 0$), respectively. Notice that this agrees with the usual meaning of these terms only when primary particles impinge on the system coming from below (*i.e.*, with $W > 0$).

NBZ ... Number of bins for the z -coordinate.

-- Default: **NBZ**=100

NBR ... Number of bins for the radial variable, $r = (x^2 + y^2)^{1/2}$.

-- Default: **NBR**=100

NBTL ... Limits of the interval where track-length distributions of primary particles are tallied. Number of track-length bins.

-- Defaults: **TLMIN**=0.0, **TLMAX**=2.5*RANGE(**EPMAX**,**KPARP**,1), **NBTL**=100

ABSEN ... Indicates a material **M** for which we require the program to tally the distribution of absorbed energy (up to three different materials can be selected, a separate line for each).

-- Default: off

• Dose distributions

DOSE2D ... The program will tally 2D, depth-radius, dose and deposited charge distributions in the body **KL**, **KC** (*i.e.*, the cylinder **KC** of layer **KL**). The numbers **NZ** and **NR** of z - and r -bins have to be specified by the user, they must be between 1 and 100. Up to five different bodies can be selected, a **DOSE2D** line for each body.

-- Default: off

• 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 one 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

6.2.2.2 Example

The example of input file given below (`cyld.in`) belongs to the `pencyl` file set included in the distribution package. It corresponds to the simulation of a narrow photon beam with $E_0 = 1.25$ MeV (roughly the average energy of gamma rays from ^{60}Co) entering a $3'' \times 3''$ NaI scintillation detector in an Al case, whose inner surface is partially covered by a layer of Al_2O_3 , which diffuses scintillation light back to the crystal and the photomultiplier. In the material data file `well.mat`, the order of the materials is NaI (`MAT=1`), Al_2O_3 (`MAT=2`) and Al (`MAT=3`). The incident beam photons move along the z -axis with $\theta_0 = 0$ deg (*i.e.*, upwards) and impinge normally on the surface of the detector. The geometry is shown schematically in the insets of Fig. 6.5, which displays two of the distributions generated by `pencyl`. The plotted distributions were obtained from 10 million random showers; the error bars represent statistical uncertainties (3σ), which are pretty small in this case.

- Example input file of the `pencyl` program.

```
....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
TITLE  Example from the distribution package...
GSTART NaI detector with Al cover and Al2O3 reflecting foil
LAYER      -0.24 -0.16  1
CENTRE      0.00  0.00
CYLIND    3    0.00  4.05
LAYER      -0.16  0.00  2
CYLIND    2    0.00  3.97
CYLIND    3    3.97  4.05
LAYER      0.00  7.72  3
CYLIND    1    0.00  3.81
```

```

CYLIND  2   3.81  3.97
CYLIND  3   3.97  4.05
LAYER           7.72  9.72  4
CYLIND  3   0.00  4.05
GEND
SKPAR  2           [Primary particles: 1=electron, 2=photon, 3=positron]
SENERG 1.25E6       [Initial energy (monoenergetic sources only)]
SPOSIT 0.0 0.0 -10.0 [Coordinates of the source centre]
NMAT   3           [Number of different materials, .le.10]
SIMPAR 1 1.0E5 1000 1.0E5 0.1 0.1 1.0E4 1000 [M,EABS,C1,C2,WCC,WCR]
SIMPAR 2 1.0E5 1000 1.0E5 0.1 0.1 1.0E4 1000 [M.EABS,C1,C2,WCC,WCR]
SIMPAR 3 1.0E5 1000 1.0E5 0.1 0.1 1.0E4 1000 [M,EABS,C1,C2,WCC,WCR]
PFNAME welld.mat   [Material definition file, 20 characters]
ABSEN  1           [Tally the distr. of absorbed E in this material]
DOSE2D 3 1 50 50   [Tally 2D dose and charge distr. in this body]
RESUME  cyld_dump.dat [Resume from this dump file, 20 chars]
DUMPTO  cyld_dump.dat [Generate this dump file, 20 chars]
DUMPP  60          [Dumping period, in sec]
NSIMSH 10.0e6      [Desired number of simulated showers]
TIME   2.0e5       [Allotted simulation time, in sec]
....+....1....+....2....+....3....+....4....+....5....+....6....+....7..

```

The upper plot in Fig. 6.5 shows the distribution of energy E_d deposited into the NaI crystal volume (per primary photon). The lower plot displays the distribution (per primary photon) of the energy E_b of “backscattered” photons, *i.e.*, photons that emerge from the system pointing “downwards”, with $W = \cos \theta < 0$. These distributions show three conspicuous structures that arise from backscattering of incident photons in the crystal volume or in the Al backing (B), escape of one of the ~ 511 keV x rays resulting from positron annihilation (A) and escape of ~ 30 keV iodine K x rays (C). The peak A is so small because pair production is a relatively unlikely process for 1.25 MeV photons (the energy is too close to the pair-production threshold).

6.2.3 Program penmain

This is a generic main program that performs simulations of electron-photon transport in complex material structures. **penmain** 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 package PENGEO (see Chapter 5), which is able to handle complicated geometries very efficiently (provided only that the user takes care of defining a sufficiently ramified genealogical tree of modules). The operation of **penmain** is completely controlled from the input data file. Although it is impossible to cover all possible cases with a “closed” program, **penmain** is flexible enough to solve a broad class of practical problems.

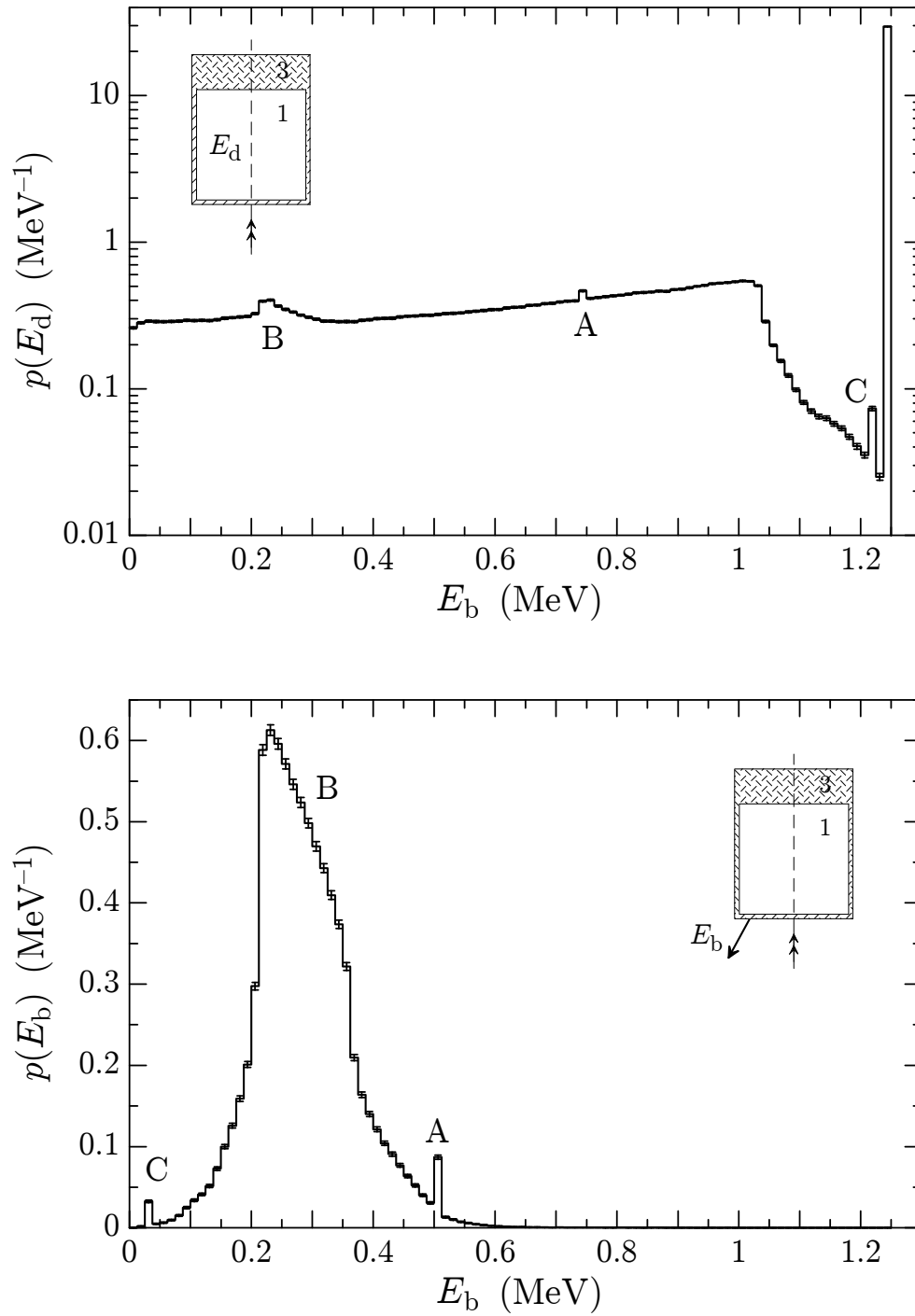


Figure 6.5: Partial results from `pencyl` for the NaI photon detector described in the text. Top: distribution of energy deposited in the NaI crystal (`MAT=1`). Bottom: energy distribution of backscattered photons.

In the default mode, **penmain** assumes that primary particles of a given kind are emitted from a point source, at a certain position, $\mathbf{r}_0 = (x_0, y_0, z_0)$, either with fixed energy or with a specified (histogram-like) energy spectrum. The initial direction of the primary particles is sampled uniformly within a cone of given semi-aperture α and central axis in the direction (θ_0, ϕ_0) [the same direction distribution assumed in the programs **penslab** and **pencyl**]. 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.

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 (see Section 5.5.1) 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. 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 in a Cartesian mesh defined by the user.

A report on the global simulation, including relevant input data as well as some partial results, is written in a file named **penmain.dat**. The calculated continuous distributions (histograms) are written in separate files, whose names start with “**pm-**” (for PenMain) and have the extension “.dat”. These files are in a format suited for direct visualisation with GNU PLOT (version 4.0).

The program **penmain** 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, the program can also write simulation results in the output files at regular time intervals.

6.2.3.1 Structure of the input file

The structure of the **penmain** 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.
.
>>>>>>> Source definition.
SKPAR  KPARP      [Primary particles: 1=electron, 2=photon, 3=positron]
SENERG SE0        [Initial energy (monoenergetic sources only)]
SPECTR Ei,Pi      [E bin: lower-end and total probability]
SPOSIT SX0,SY0,SZ0 [Coordinates of the source]
```

```

SDIREC STHETA,SPHI          [Beam axis direction angles, in deg]
SAPERT SALPHA                [Beam aperture, in deg]
>>>>>>> Input phase-space file (psf).
IPSFN  psf-filename.ext      [Input psf name, 20 characters]
IPSPLI NSPLIT                [Splitting number]
EPMAX  EPMAX                 [Maximum energy of particles in the psf]
.
>>>>>>> Material data and simulation parameters.
NMAT   NMAT                  [Number of different materials, .le.10]
SIMPAR M,EABS(1:3,M),C1,C2,WCC,WCR [Sim. parameters for material M]
PFNAME mat-filename.ext      [Material definition file, 20 chars]
>>>>>>> Geometry definition file.
GEOMFN geo-filename.ext      [Geometry definition file, 20 chars]
DSMAX  KB,DSMAX(KB)          [KB, maximum step length in body KB]
>>>>>>> Interaction forcing.
IFORCE KB,KPAR,ICOL,FORCER,WLOW,WHIG [Interaction forcing]
.
>>>>>>> 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]
.
>>>>>>> Impact detectors (up to 25 different detectors).
IMPDET EDIL,EDIU,NCHI,IPSF   [Energy window, no. of channels and IPSF]
IDPSF  pm-psf-impdet-#.dat    [Output psf file name, 20 chars]
IDSPC  pm-spc-impdet-#.dat    [Output spectrum file name, 20 chars]
IDBODY KB                    [Active body; one line for each body]
IDKPAR KPAR                  [Kind of detected particles, one line each]
>>>>>>> Energy-deposition detectors (up to 25).
ENDDET EDEL,EDEU,NCHE       [Energy window and number of channels]
EDSPC  pm-spc-enddet-#.dat    [Output spectrum file name, 20 chars]
EDBODY KB                    [Active body; one line for each body]
>>>>>>> Dose distribution.
GRIDX  XL,XU                 [X coordinates of the enclosure vertices]
GRIDY  YL,YU                 [Y coordinates of the enclosure vertices]
GRIDZ  ZL,ZU                 [Z coordinates of the enclosure vertices]
GRIDBN NX,NY,NZ              [Numbers of bins]
.
>>>>>>> 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 wrong 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.

• Source definition

SKPAR ... Kind of primary particle (1 = electrons, 2 = photons, or 3 = positrons).

-- Default: KPARP=1

SENERG ... For monoenergetic sources: initial energy SE0 of primary particles.

-- Default: SE0=1.0E6

SPECTR ... For a source with continuous (stepwise constant) energy spectrum, each “**SPECTR**” line gives the lower end-point (**Ei**) of an energy bin of the spectrum and the associated relative probability (**Pi**), integrated over the bin. Up to **NSEM** = 200 lines can be entered, in arbitrary order. The upper end of the spectrum is defined by entering a line with **Ei** equal to the upper energy end point, and with a negative **Pi** value.

-- Default: none

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

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

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

-- Defaults: STHETA=0.0, SPHI=0.0

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

-- Default: SALPHA=0.0

IPSFN ... Name of an input phase-space file (up to 20 characters).

The initial state variables of primary particles can be read directly from a set of pre-calculated phase-space files. When this option is active, previous definitions about the source are ignored.

-- Default: none

Up to 100 phase-space files may be declared. They are read sequentially.

Phase-space files are in ASCII format. Each line defines the initial state of a particle; it contains the following quantities in free format (and in the order they are listed here):

```
-- KPAR, type of particle (1 = electron; 2 = photon; 3 = positron).
-- E, energy (eV).
-- X,Y,Z, position coordinates (cm).
-- U,V,W, direction cosines.
-- WGHT, weight.
-- ILB(1),ILB(2),ILB(3),ILB(4), a set of indices that provide information on
how the particle was generated (see Section 6.1.2).
-- NSHI, incremental shower number (difference between the shower numbers of
the present particle and the one preceding it in the phase-space file).
```

Phase-space files can be generated by running `penmain` using an impact detector with the flag `IPSF = 1` or `2` (see below).

Because of the limited size of phase-space files, the results of analogue simulations tend to be “too noisy”. This can be partially corrected by splitting the particles from the phase-space file.

IPSPLI ... Splitting number. Each particle in the phase-space files will be split into `NSPLIT` equivalent particles, with weights equal to `WGHT/NSPLIT`.
 -- Default: `NSPLIT=1` (no splitting)

Notice that there is a “latent” uncertainty in the phase-space files, which sets a limit to the accuracy that can be attained by using large splitting numbers.

EPMAX ... Maximum energy (in eV) of particles in the phase-space files. **EPMAX** is the upper limit of the energy interval covered by the simulation lookup tables. To minimise interpolation errors, **EPMAX** should not be much larger than the maximum energy actually occurring during the simulation.

When the initial state variables of particles are read from a phase-space file, this parameter is required to initialise **PENELOPE** and is critical; the code crashes if it finds a particle that has energy larger than **EPMAX**.

```
-- Default: EPMAX=1.0E9 (interpolation is not optimal)
```

• Material data and simulation parameters

NMAT ... Number of different materials (up to 10 with the original program dimensions). Materials are identified by their ordering in **PENELOPE**’s input material data file.

```
-- Default: NMAT=1
```

SIMPAR ... Set of simulation parameters for the *M*-th 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)`. One line for each material.

```
-- Defaults: EABS(1,M)=EABS(3,M)=0.01*EPMAX, EABS(2,M)=0.001*EPMAX  

C1(M)=C2(M)=0.1, WCC(M)=EABS(1,M), WCR(M)=EABS(2,M)
```

EPMAX is the upper limit of the energy interval covered by the simulation lookup tables.

PFNAME ... Name of **PENELOPE**’s input material data file (up to 20 characters).

```
-- Default: none
```

• 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 (see Section 5.6). Notice that bodies must be referenced by the internal label assigned by PENGEOm; this label can be identified by running the viewer GVIEW2D or by inspecting the geometry report file written by PENGEOm.

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). The values WLOW, 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 $E = \text{EPMAX}$. 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=EPMAX, 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 material system moving upwards ($W > 0$) and downwards ($W < 0$), respectively. Notice that this agrees with the usual meaning of these terms only when primary particles impinge on the system from below (*i.e.*, with $W > 0$).

• Impact detectors

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

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

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=+1; particles that enter the detector are transported as usual. A phase-space file is created.

IPSF=-1; particles that enter the detector are transported as usual. No phase-space file is created.

IPSF=+2; the simulation of a particle is discontinued when it enters the detector. A phase-space file is created.

IPSF=-2; the simulation of a particle is discontinued when it enters the detector. No phase-space file is created.

-- Default: none

IDPSF ... Name of the output phase-space file (up to 20 characters).

-- Default: pm-psf-impdet-#.dat

IDSPC ... Name of the output energy-spectrum file (up to 20 characters).

-- Default: pm-spc-impdet-#.dat

IDBODY ... Active body of the detector. One line for each active body.

-- Default: none

NOTE: a body cannot be part of more than one impact detector.

IDKPAR ... Kind of particle that is detected (1=electrons, 2=photons or 3=positrons). One line for each kind.

-- Default: All particles are detected

• Energy-deposition detectors

ENDDET ... Starts the definition of a new energy-deposition detector. Up to 25 different detectors can be considered. EDEL and EDEU are the lower and upper limits of the energy window covered by the detector. NCHE is the number of energy channels in the output spectrum ($\leq 1,000$).

WARNING: The energy-deposition spectrum may be strongly biased when interaction forcing is applied.

EDSPC ... Name of the output spectrum file (up to 20 characters).

-- Default: pm-spc-enddet-#.dat

EDBODY ... Active body of the detector. One line for each active body.

-- Default: none

NOTE: A body cannot be part of more than one energy-deposition detector.

• Dose distributions

The program can calculate the absorbed dose distribution inside a parallelepiped (dose enclosure) whose edges are parallel to the axes of the laboratory frame. The enclosure is defined by giving the coordinates of its vertices. The dose is tallied using a uniform Cartesian grid with NX, NY and NZ (≤ 100) bins along the directions of the coordinate axes.

GRIDX ... x -coordinates of the vertices of the dose enclosure.
 -- Default: none
 GRIDY ... y -coordinates of the vertices of the dose enclosure.
 -- Default: none
 GRIDZ ... z -coordinates of the vertices of the dose enclosure.
 -- 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

• 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 one 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

6.3 Selecting the simulation parameters

The speed and accuracy of the simulation of electrons and positrons is determined by the values of the simulation parameters E_{abs} , C_1 , C_2 , W_{cc} , W_{cr} and s_{max} , which are selected by the user for each material in the simulated structure³. Here we summarise the rules for assigning “safe” values to these parameters.

The absorption energies E_{abs} are determined either by the characteristics of the experiment or by the required space resolution. If we want to tally dose or deposited-charge

³To 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).

distributions, E_{abs} should be such that the residual range $R(E_{\text{abs}})$ of electrons/positrons is less than the typical dimensions of the volume bins used to tally these distributions⁴. In other cases, it is advisable to run short simulations (for the considered body alone) with increasing values of E_{abs} (starting from 50 eV) to study the effect of this parameter on the results.

The allowed values of the elastic-scattering parameters C_1 and C_2 are limited to the interval $[0, 0.2]$. Due to the consideration of the energy dependence of the cross sections for soft interactions and of the hard mean free paths (see Sections 4.2 and 4.3), these parameters have a very weak influence on the results. Our recommended practice is to set $C_1 = C_2 = 0.05$, which is fairly conservative, as shown by the example described below. 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).

We have already indicated that the cutoff energies W_{cc} and W_{cr} 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 energy distributions are of no interest, our recommendation is to set these cutoff energies equal to one hundredth of the typical energy of primary particles. Note that, for the sake of consistency, W_{cc} should not be larger than the absorption energy of electrons in the material, $\text{EABS}(1, \text{M})$ (otherwise, we would lose secondary electrons that have energies larger than $\text{EABS}(1, \text{M})$). Similarly, W_{cr} should be less than the photon absorption energy $\text{EABS}(2, \text{M})$.

The maximum allowed step length s_{max} (denoted by **DSMAX** in the Fortran source files) should be about, or less than one tenth of the characteristic thickness of the body where the particle moves. This ensures that, on average, there will be more than 13 soft events⁵ (hinges) along a typical electron/positron track through that body, which is enough to “wash out” the details of the artificial distributions used to sample these events. Notice however that PENELOPE internally forces the step length to be less than $\sim 4\lambda_{\text{T}}^{(\text{h})}$ (see Section 4.4). Therefore, for thick bodies (thicker than $\sim 15\lambda_{\text{T}}^{(\text{h})}$), the average number of hinges along each track is larger than about 20, and it is not necessary to limit the length of the steps. In this case, we can set $s_{\text{max}} = 10^{35}$ cm, or some other very large value, to switch off the external step-length control.

The example main program **penslab** (see Section 6.2.1) can be readily used to study the effect of the simulation parameters for a material body of a given characteristic thickness. As an example, Figs. 6.6 and 6.7 display partial results from **penslab** simulations for a parallel electron beam of 500 keV impinging normally on the surface of a 200- μm -thick aluminium slab. The absorption energies were set equal to 10 keV (for all kinds of particles) and W_{cr} was given a negative value, which compels PENELOPE to set $W_{\text{cr}} = 10$ eV, and to disregard emission of soft bremsstrahlung (with $W < 10$ eV). That

⁴PENELOPE prints tables of electron and positron ranges if subroutine **PEINIT** is invoked with **INFO=3** or larger. Alternatively, these tables can be generated by running the program **tables**.

⁵PENELOPE randomises s_{max} in such a way that the actual step length never exceeds the value s_{max} set by the user and, on average, is equal to $0.75s_{\text{max}}$.

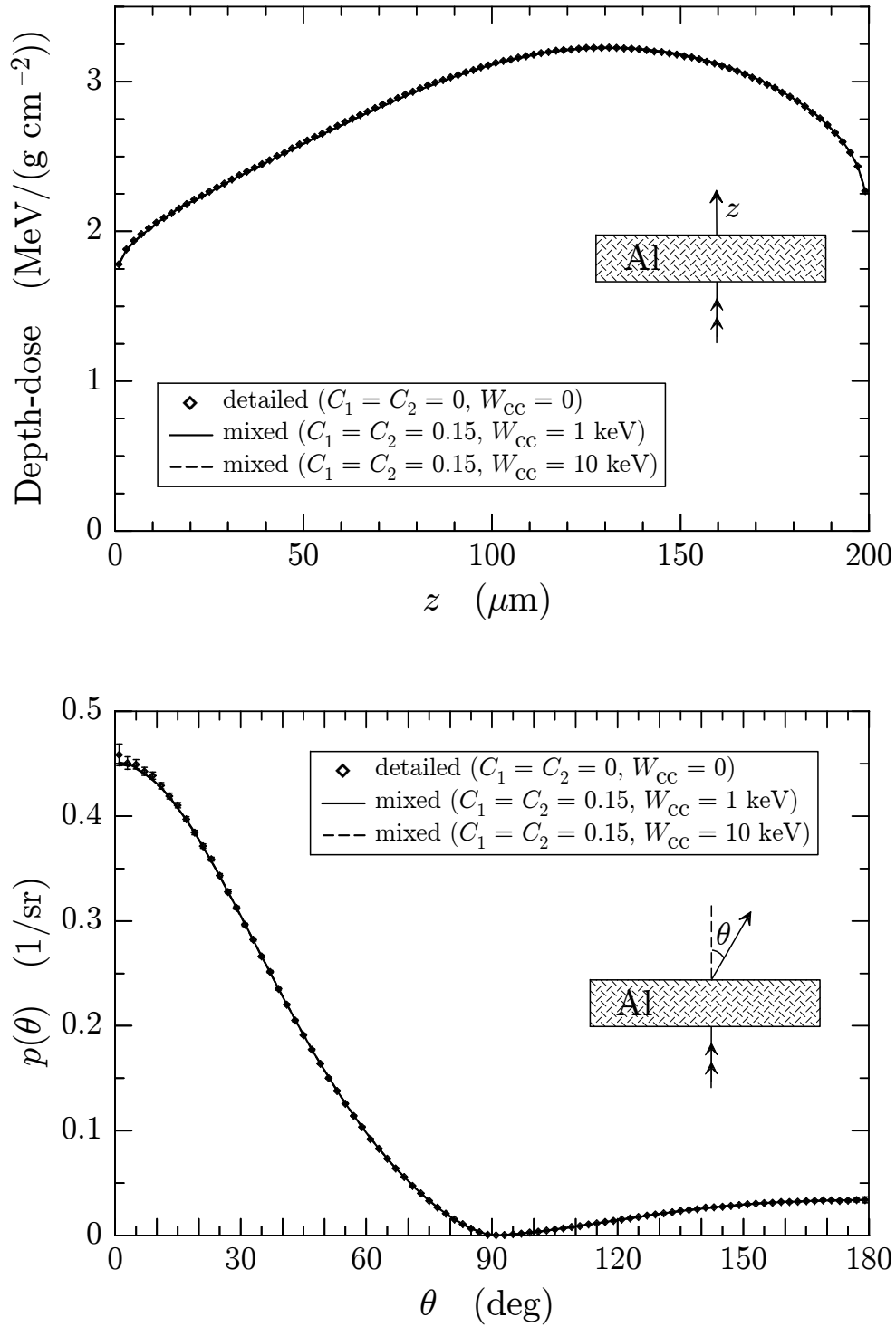


Figure 6.6: Results from *penslab* for a 500 keV electron beam impinging normally on the surface of a 200-μm-thick aluminium slab (further details are given in the text). Top: depth-dose distribution within the slab. Bottom: angular distribution of emerging (transmitted and backscattered) electrons (primary and secondary).

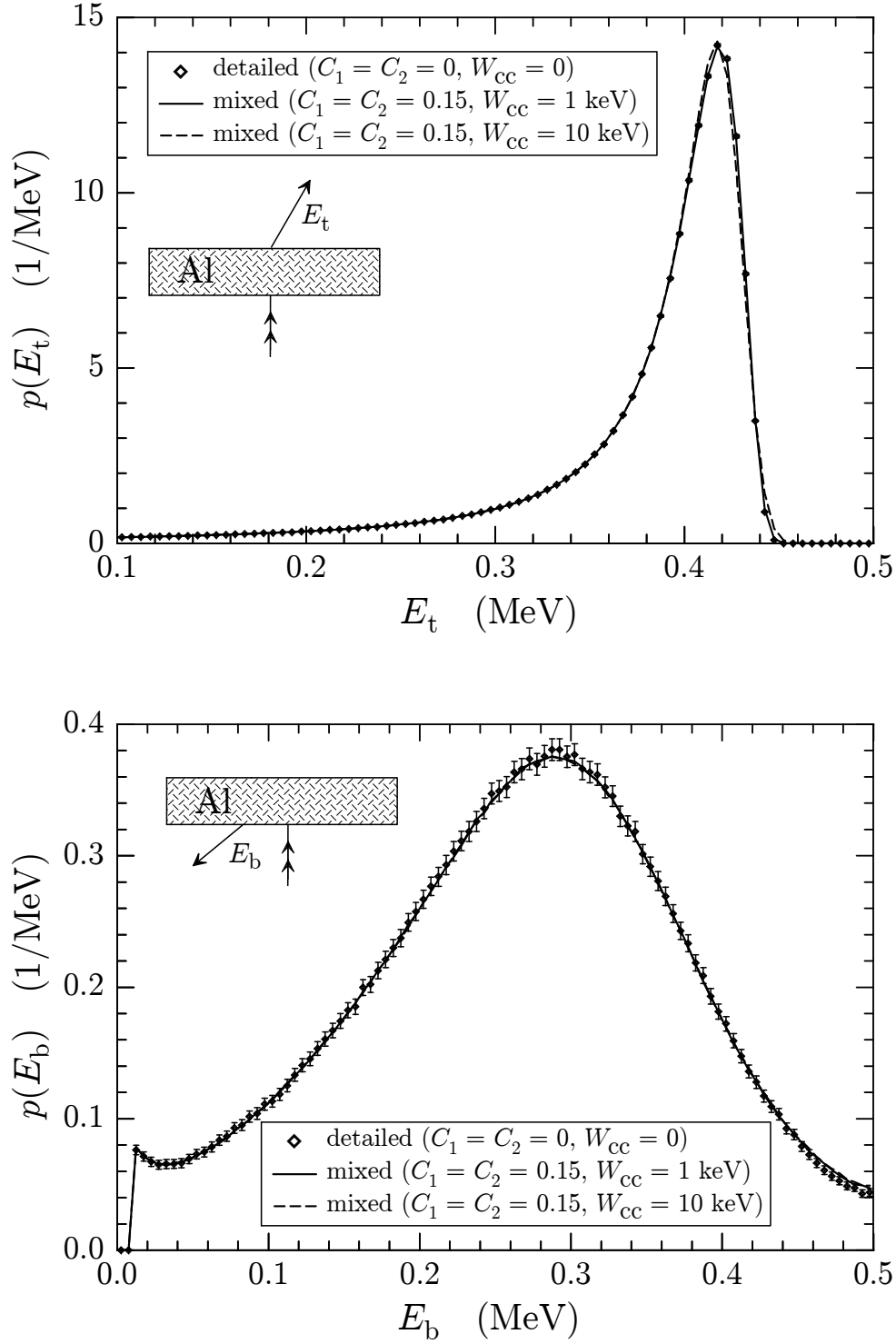


Figure 6.7: Results from *penslab* for a 500 keV electron beam impinging normally on the surface of a 200- μm -thick aluminium slab (further details are given in the text). Top: energy distribution of transmitted electrons. Bottom: energy distribution of backscattered electrons. Secondary electrons are included in both cases.

is, radiative events that yield bremsstrahlung photons with energy higher than 10 eV were simulated individually. First, we ran `penslab.exe` using $W_{cc} = 0$ and $C_1 = C_2 = 0$; in this case, PENELOPE performs a purely detailed, collision by collision, simulation and, therefore, it provides exact results (affected only by statistical uncertainties and by inaccuracies of the physical interaction model). Differences between these results and those from mixed simulations are then completely attributable to the approximations in our mixed transport algorithm. To our knowledge, no other high-energy transport code allows detailed simulation and this kind of direct validation of electron/positron transport mechanics.

Figures 6.6 and 6.7 show results from this detailed simulation, obtained from 10 million random showers⁶; the error bars indicate statistical uncertainties (3σ). The continuous lines in Figs. 6.6 and 6.7 represent results from a mixed simulation with $C_1 = C_2 = 0.15$, $s_{\max} = 0.002$ cm, $W_{cc} = 1$ keV (and $W_{cr} = -10$ eV, as before). With these relatively high values of C_1 and C_2 , the mixed simulation is quite fast, the speed (generated showers per second) is about 40 times higher than that of the detailed simulation⁷. The continuous lines in Figs. 6.6 and 6.7 represent results from this mixed simulation obtained from 100 million showers; the associated statistical uncertainties are much (a factor ~ 3.3) smaller than those of the detailed simulation. As shown in the plots, the results from the mixed simulation are practically equivalent to those from detailed simulation. The only visible difference is in the energy distribution of backscattered electrons (bottom plot in Fig. 6.7), where mixed simulation tends to overestimate slightly the backscattering of electrons with energies close to that of the incident beam. It should be noted that electron (and positron) backscattering is one of the most difficult cases to study because it involves transport near and across an interface.

To illustrate the effect of the cutoff energy loss W_{cc} , we ran a second mixed simulation with this parameter changed to 10 keV (*i.e.*, one fiftieth of the incident beam energy, twice the value recommended above). The results, from 100 million simulated showers, agreed closely with those of the previous mixed simulation with $W_{cc} = 1$ keV, except for the energy distribution of transmitted electrons, which were slightly distorted (see the top plot in Fig. 6.7); a similar distortion occurs for the distribution of energy deposited in the slab. On the other hand, the simulation was only 15 % faster than with $W_{cc} = 1$ keV. Usually, when W_{cc} and W_{cr} are larger than ~ 5 keV, the simulation speed is fairly insensitive to variations of these cutoff values. This is not unexpected, because the energy-loss DCSs for inelastic collisions and bremsstrahlung emission decrease rapidly with W (roughly as W^{-2} and W^{-1} , respectively) and, therefore, most of these interactions involve energy losses much smaller than ~ 5 keV.

⁶The program was compiled with the GNU GFORTRAN compiler and ran on a Pentium M 2.00 GHz processor. The generation of the 10 million showers took 39 hours.

⁷The simulation speed here is bound by the fact that each particle must undergo a minimum number of soft interactions (hinges). For thick bodies, much higher simulation speeds can be reached.

6.4 The code SHOWER

Monte Carlo simulation has proven to be a very 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 distribution package includes a binary file named SHOWER that generates electron-photon showers within a slab (of one of the 280 materials defined in `pdcompos.p06`, see Table 6.1) 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 colours and intensities that vary with the energy of the particle. It is worth mentioning 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 the 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 shower. Observation of single showers projected on a revolving plane gives a truly three-dimensional perspective of the transport process.

6.5 Installation

The complete PENELOPE code system is distributed as a single ZIP compressed file named `penelope.zip`, which contains the Fortran source files, the example main programs and auxiliary programs, the database, and documentation. To install PENELOPE on your computer, simply inflate (unzip) this file, keeping its directory organisation. The directory structure and contents of the code system are the following (see Fig. 6.8):

- **Directory `fsource`.** It contains the following 5 Fortran source files:

- `penelope.f` ... basic subroutine package for the simulation of coupled electron-photon transport in unbounded media.
- `pengeom.f` ... subroutine package for tracking particles through modular quadric geometries (handles systems with up to 10,000 surfaces and 5,000 bodies).

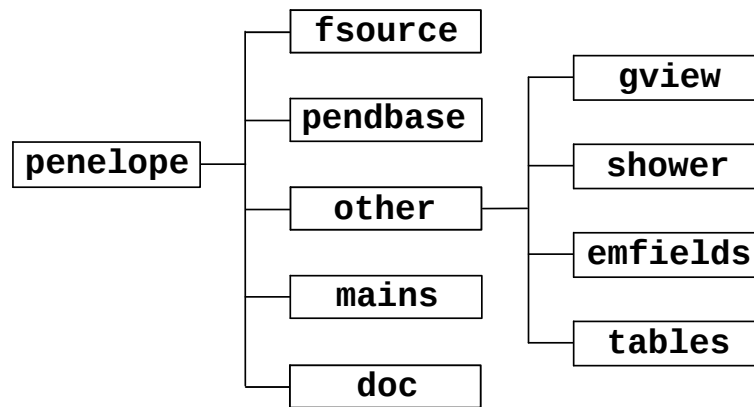


Figure 6.8: Directory tree of the PENELOPE code system.

penvared.f ... variance-reduction subroutines (splitting, Russian roulette and interaction forcing).

timer.f ... timing subroutines, based on standard intrinsic procedures of Fortran 95. They also work with some Fortran 77 and Fortran 90 compilers.

material.f ... main program to generate material data files.

To obtain the executable binary file of **material**, compile and link the source files **material.f** and **penelope.f**. Notice that **penelope.f** is declared through an include statement inside **material.f** and does not have to be listed in the compilation command. The executable file **material.exe** must be placed and run in the same subdirectory as the database files (**pendbase**).

- **Directory pendbase.** PENELOPE database. 797 files with the extension ".p06" (see Section 6.1.1).

- **Directory other.** Consists of the following 4 subdirectories,

gview Contains the geometry viewers **GVIEW2D** and **GVIEW3D** (see Section 5.6), which are operable under Microsoft Windows, and several examples of geometry definition files.

shower ... Contains the executable binary file of the code **SHOWER** (see Section 6.4), which operates under Microsoft Windows. This code generates electron-photon showers within a slab (of one of the 280 materials defined in **pdcompos.p06**) and displays the showers projected on the screen plane. To use **SHOWER**, just copy the file **shower.exe** into the directory **pendbase** and run it from there. This little tool is particularly useful for teaching purposes, it makes radiation physics "visible".

emfields ... Contains the subroutine package **penfield.f**, which does simulation of electron/positron transport under external static magnetic (and electric) fields, and examples of programs that use it.

tables This directory contains the program **tables.f**, which reads a material data file and generates tables of interaction data (cross sections, mean free paths, ranges, ...), and the GNUPLOT script **tables.gnu**, which plots the most relevant of these quantities as functions of the energy. The program **tables.exe** also calculates interpolated values of these quantities at specified energies.

- **Directory mains.** Examples of main programs, 3 subdirectories.

penslab Contains the main program **penslab.f** for particle transport in a slab, an example of input file and the associated material data file. The file **penslab.gnu** is a GNUPLOT script (see below) that displays the continuous distributions generated by the program **penslab**; **psource.gnu** displays the energy spectrum of primary particles (if the source is not monoenergetic).

pencil Contains the main program **pencil.f** for particle transport in cylindrical structures, the geometry viewer **gviewc.exe** for these structures (which operates under Microsoft Windows), examples of input and material-data files and GNUPLOT scripts to visualise the results.

penmain Contains the generic main program **penmain.f** for particle transport in quadric geometries, input files for two sample problems and various GNUPLOT scripts to display the simulated distributions.

The executable files of **penslab**, **pencil** and **penmain** are obtained by compiling and linking the following groups of source files:

penslab: **penslab.f**, **penelope.f**, **penvared.f**, **timer.f**

pencil : **pencil.f**, **penelope.f**, **penvared.f**, **timer.f**

penmain: **penmain.f**, **penelope.f**, **pengeom.f**, **penvared.f**, **timer.f**

NOTE: To simplify the typing of compilation commands, all the modules used by a main program have been declared through include statements within the program (see, for example, the source file **penmain.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.

The example main programs **penslab**, **pencil** and **penmain** generate multiple files with simulated probability distributions. Each output file has a heading describing its content, which is in a format ready for visualisation with a plotting program. We use GNUPLOT, which is small in size, available for various platforms (including Linux and Windows) and free; this software can be downloaded from the distribution sites listed at the Gnuplot Central site, <http://www.gnuplot.info>. Each main program 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 **penslab.exe** you can visualise the results by simply 1) copying the file **penslab.gnu** from the directory **penslab** to the directory that contains the results and 2) entering the command “**wgnuplot penslab.gnu**” (or clicking the icon of the script). Note that, depending on the options selected in the input data file, certain

distributions may not be generated; in this case, the corresponding GNUPLOT script will not work (nothing will be displayed).

- **Directory doc.** Documentation, 2 files.

`tutorial.pdf` ... A guided tour through the PENELOPE code system. Includes brief descriptions of the programs and subroutine packages, instructions to install the system, and to build binary executable files.

`penelope-2006-nea.pdf` ... The present manual.

The simulation programs 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 compiler⁸ or the Fortran 95 compiler GFORTRAN⁹, both from the Free Software Foundation. Another free compiler, available for different platforms is G95¹⁰. Salford Software¹¹ offers his Fortran 95 compiler Salford FTN95 Personal Edition free of charge for personal, non-commercial usage; on execution, programs compiled with FTN95PE display a banner.

⁸<http://www.geocities.com/Athens/Olympus/5564/> In this site you can also find C.G. Page's book, *Professional Programmer's Guide to Fortran 77*, in pdf format.

⁹<http://gcc.gnu.org/wiki/GFortran>

¹⁰<http://g95.org/>

¹¹<http://www.salfordsoftware.co.uk/>

Appendix A

Collision kinematics

To cover the complete energy range of interest in radiation transport studies we use relativistic kinematics. Let $\mathbf{P}$ denote the energy-momentum 4-vector of a particle, *i.e.*,

$$\mathbf{P} = (\mathcal{W}c^{-1}, \mathbf{p}), \quad (\text{A.1})$$

where $\mathcal{W}$ and $\mathbf{p}$ are the total energy (including the rest energy) and momentum respectively and c is the velocity of light in vacuum. The product of 4-vectors, defined by

$$\mathbf{P} \cdot \mathbf{P}' = \mathcal{W}\mathcal{W}'c^{-2} - \mathbf{p} \cdot \mathbf{p}', \quad (\text{A.2})$$

is invariant under Lorentz transformations. The rest mass m of a particle determines the invariant length of its energy-momentum,

$$\mathbf{P} \cdot \mathbf{P} = \mathcal{W}^2c^{-2} - \mathbf{p}^2 = (mc)^2. \quad (\text{A.3})$$

The kinetic energy E of a massive particle ($m \neq 0$) is defined as

$$E = \mathcal{W} - mc^2, \quad (\text{A.4})$$

where mc^2 is the rest energy. The magnitude of the momentum is given by

$$(cp)^2 = E(E + 2mc^2). \quad (\text{A.5})$$

In terms of the velocity $\mathbf{v}$ of the particle, we have

$$E = (\gamma - 1)mc^2 \quad \text{and} \quad \mathbf{p} = \beta\gamma mc\hat{\mathbf{v}}, \quad (\text{A.6})$$

where

$$\beta \equiv \frac{v}{c} = \sqrt{\frac{\gamma^2 - 1}{\gamma^2}} = \sqrt{\frac{E(E + 2mc^2)}{(E + mc^2)^2}} \quad (\text{A.7})$$

is the velocity of the particle in units of c and

$$\gamma \equiv \sqrt{\frac{1}{1 - \beta^2}} = \frac{E + mc^2}{mc^2} \quad (\text{A.8})$$

is the total energy in units of the rest energy. From the relation (A.5), it follows that

$$E = \sqrt{(cp)^2 + m^2c^4} - mc^2 \quad (\text{A.9})$$

and

$$\frac{dp}{dE} = \frac{1}{v} = \frac{1}{c\beta}. \quad (\text{A.10})$$

For a photon (and any other particle with $m = 0$), the energy and momentum are related by

$$E = cp. \quad (\text{A.11})$$

A.1 Two-body reactions

Consider a reaction in which a projectile “1” collides with a target “2” initially at rest in the laboratory frame of reference. We limit our study to the important case of two-body reactions in which the final products are two particles, “3” and “4”. The kinematics of such reactions is governed by energy and momentum conservation.

We take the direction of movement of the projectile to be the z -axis, and set the x -axis in such a way that the reaction plane (*i.e.*, the plane determined by the momenta of particles “1”, “3” and “4”) is the x - z plane. The energy-momentum 4-vectors of the projectile, the target and the reaction products are then (see Fig. A.1)

$$\mathbf{P}_1 = (\mathcal{W}_1c^{-1}, 0, 0, p_1), \quad (\text{A.12a})$$

$$\mathbf{P}_2 = (m_2c, 0, 0, 0), \quad (\text{A.12b})$$

$$\mathbf{P}_3 = (\mathcal{W}_3c^{-1}, p_3 \sin \theta_3, 0, p_3 \cos \theta_3), \quad (\text{A.12c})$$

$$\mathbf{P}_4 = (\mathcal{W}_4c^{-1}, -p_4 \sin \theta_4, 0, p_4 \cos \theta_4). \quad (\text{A.12d})$$

Energy and momentum conservation is expressed by the 4-vector equation

$$\mathbf{P}_1 + \mathbf{P}_2 = \mathbf{P}_3 + \mathbf{P}_4. \quad (\text{A.13})$$

From this equation, the angles of emergence of the final particles, θ_3 and θ_4 , are uniquely determined by their energies, $\mathcal{W}_3$ and $\mathcal{W}_4$. Thus,

$$\begin{aligned} m_4^2c^2 &= \mathbf{P}_4 \cdot \mathbf{P}_4 = (\mathbf{P}_1 + \mathbf{P}_2 - \mathbf{P}_3) \cdot (\mathbf{P}_1 + \mathbf{P}_2 - \mathbf{P}_3) \\ &= \mathbf{P}_1 \cdot \mathbf{P}_1 + \mathbf{P}_2 \cdot \mathbf{P}_2 + \mathbf{P}_3 \cdot \mathbf{P}_3 + 2\mathbf{P}_1 \cdot \mathbf{P}_2 - 2\mathbf{P}_1 \cdot \mathbf{P}_3 - 2\mathbf{P}_2 \cdot \mathbf{P}_3 \\ &= m_1^2c^2 + m_2^2c^2 + m_3^2c^2 + 2\mathcal{W}_1\mathcal{W}_2c^{-2} \\ &\quad - 2(\mathcal{W}_1\mathcal{W}_3c^{-2} - p_1p_3 \cos \theta_3) - 2\mathcal{W}_2\mathcal{W}_3c^{-2}, \end{aligned} \quad (\text{A.14})$$

and it follows that

$$\cos \theta_3 = \frac{m_4^2c^4 - m_1^2c^4 - m_2^2c^4 - m_3^2c^4 + 2\mathcal{W}_1(\mathcal{W}_3 - \mathcal{W}_2) + 2\mathcal{W}_2\mathcal{W}_3}{2(\mathcal{W}_1^2 - m_1^2c^4)^{1/2}(\mathcal{W}_3^2 - m_3^2c^4)^{1/2}}. \quad (\text{A.15})$$

Clearly, by symmetry, we can obtain a corresponding expression for $\cos \theta_4$ by interchanging the indices 3 and 4

$$\cos \theta_4 = \frac{m_3^2 c^4 - m_1^2 c^4 - m_2^2 c^4 - m_4^2 c^4 + 2\mathcal{W}_1(\mathcal{W}_4 - \mathcal{W}_2) + 2\mathcal{W}_2\mathcal{W}_4}{2(\mathcal{W}_1^2 - m_1^2 c^4)^{1/2}(\mathcal{W}_4^2 - m_4^2 c^4)^{1/2}}. \quad (\text{A.16})$$

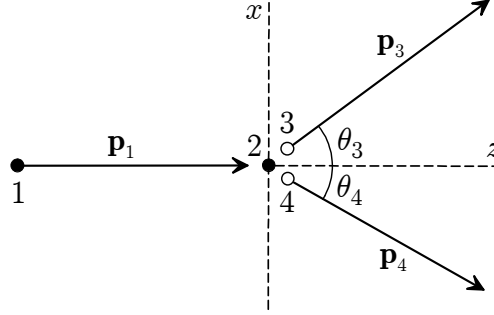


Figure A.1: Kinematics of two-body reactions.

The different two-body reactions found in Monte Carlo simulation of coupled electron-photon transport can be characterised by a single parameter, namely the energy of one of the particles that result from the reaction. The energy of the second particle is determined by energy conservation. Equations (A.15) and (A.16) then fix the polar angles, θ_3 and θ_4 , of the final directions. Explicitly, we have

- Binary collisions of electrons and positrons with free electrons at rest.

Projectile:	Electron or positron	$m_1 = m_e, \quad \mathcal{W}_1 = E + m_e c^2.$
Target:	Electron	$m_2 = m_e, \quad \mathcal{W}_2 = m_e c^2.$
Scattered particle:		$m_3 = m_e, \quad \mathcal{W}_3 = E - W + m_e c^2.$
Recoil electron:		$m_4 = m_e, \quad \mathcal{W}_4 = W + m_e c^2.$

$$\cos \theta_3 = \left(\frac{E - W}{E} \frac{E + 2m_e c^2}{E - W + 2m_e c^2} \right)^{1/2}, \quad (\text{A.17})$$

$$\cos \theta_4 = \left(\frac{W}{E} \frac{E + 2m_e c^2}{W + 2m_e c^2} \right)^{1/2}. \quad (\text{A.18})$$

- Compton scattering of photons by free electrons at rest.

Projectile:	Photon	$m_1 = 0, \quad \mathcal{W}_1 = E \equiv \kappa m_e c^2.$
Target:	Electron	$m_2 = m_e, \quad \mathcal{W}_2 = m_e c^2.$
Scattered photon:		$m_3 = 0, \quad \mathcal{W}_3 \equiv \tau E.$
Recoil electron:		$m_4 = m_e, \quad \mathcal{W}_4 = m_e c^2 + (1 - \tau)E.$

$$\cos \theta_3 = \frac{1}{\kappa} \left(\kappa + 1 - \frac{1}{\tau} \right), \quad (\text{A.19})$$

$$\cos \theta_4 = (\kappa + 1) \left(\frac{1 - \tau}{\kappa [2 + \kappa(1 - \tau)]} \right)^{1/2}. \quad (\text{A.20})$$

- Annihilation of positrons with free electrons at rest.

Projectile:	Positron	$m_1 = m_e, \quad \mathcal{W}_1 = E + m_e c^2 \equiv \gamma m_e c^2.$
Target:	Electron	$m_2 = m_e, \quad \mathcal{W}_2 = m_e c^2.$
Annihilation photons:		$m_3 = 0, \quad \mathcal{W}_3 \equiv \zeta(E + 2m_e c^2).$
		$m_4 = 0, \quad \mathcal{W}_4 = (1 - \zeta)(E + 2m_e c^2).$

$$\cos \theta_3 = (\gamma^2 - 1)^{-1/2} (\gamma + 1 - 1/\zeta), \quad (\text{A.21})$$

$$\cos \theta_4 = (\gamma^2 - 1)^{-1/2} \left(\gamma + 1 - \frac{1}{1 - \zeta} \right). \quad (\text{A.22})$$

A.1.1 Elastic scattering

By definition, elastic collisions keep the internal structure (*i.e.*, the mass) of the projectile and target particles unaltered. Let us consider the kinematics of elastic collisions of a projectile of mass m ($= m_1 = m_3$) and kinetic energy E with a target particle of mass M ($= m_2 = m_4$) at rest (see Fig. A.2). After the interaction, the target recoils with a certain kinetic energy W and the kinetic energy of the projectile is reduced to $E' = E - W$. The angular deflection of the projectile $\cos \theta$ and the energy transfer W are related through Eq. (A.15), which now reads

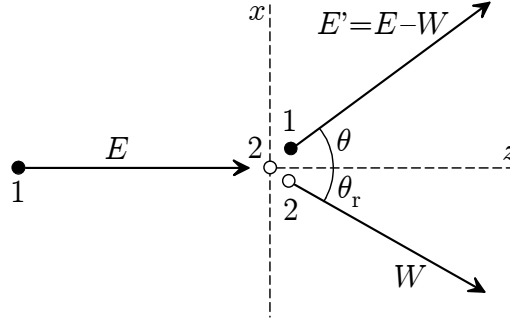
$$\cos \theta = \frac{E(E + 2mc^2) - W(E + mc^2 + Mc^2)}{\sqrt{E(E + 2mc^2)(E - W)(E - W + 2mc^2)}}. \quad (\text{A.23})$$

The target recoil direction is given by Eq. (A.16),

$$\cos \theta_r = \frac{(E + mc^2 + Mc^2)W}{\sqrt{E(E + 2mc^2)W(W + 2mc^2)}}. \quad (\text{A.24})$$

Solving Eq. (A.23), we obtain the following expression for the energy transfer W corresponding to a given scattering angle θ ,

$$\begin{aligned} W = & \left[(E + mc^2) \sin^2 \theta + Mc^2 - \cos \theta \sqrt{M^2 c^4 - m^2 c^4 \sin^2 \theta} \right] \\ & \times \frac{E(E + 2mc^2)}{(E + mc^2 + Mc^2)^2 - E(E + 2mc^2) \cos^2 \theta}. \end{aligned} \quad (\text{A.25})$$

**Figure A.2:** Kinematics of elastic collisions.

In the case of collisions of particles with equal mass, $m = M$, this expression simplifies to

$$W = \frac{E(E + 2mc^2) \sin^2 \theta}{E \sin^2 \theta + 2mc^2} \quad \text{if } M = m. \quad (\text{A.26})$$

In this case, θ can only take values less than 90 deg. For $\theta = 90$ deg, we have $W = E$ (*i.e.*, the full energy and momentum of the projectile are transferred to the target). Notice that for binary collisions of electrons and positrons ($m = m_e$), the relation (A.26) becomes identical to (A.17).

For elastic collisions of electrons by atoms and ions, the mass of the target is much larger than that of the projectile and Eq. (A.25) becomes

$$W = \frac{[(E + mc^2) \sin^2 \theta + Mc^2(1 - \cos \theta)] E(E + 2mc^2)}{(E + Mc^2)^2 - E(E + 2mc^2) \cos^2 \theta} \quad \text{if } M \gg m. \quad (\text{A.27})$$

The non-relativistic limit ($c \rightarrow \infty$) of this expression is

$$W = \frac{2m}{M}(1 - \cos \theta)E \quad \text{if } M \gg m \text{ and } E \ll mc^2. \quad (\text{A.28})$$

A.2 Inelastic collisions of charged particles

We consider here the kinematics of inelastic collisions of charged particles of mass m and velocity $\mathbf{v}$ as seen from a frame of reference where the stopping medium is at rest (laboratory frame). Let $\mathbf{p}$ and E be the momentum and the kinetic energy of the projectile just before an inelastic collision, the corresponding quantities after the collision are denoted by $\mathbf{p}'$ and $E' = E - W$, respectively. Evidently, for positrons the maximum energy loss is $W_{\max} = E$. In the case of ionisation by electron impact, owing to the indistinguishability between the projectile and the ejected electron, the maximum energy loss is $W_{\max} \simeq E/2$ (see Section 3.2). The momentum transfer in the collision is $\mathbf{q} \equiv \mathbf{p} - \mathbf{p}'$. It is customary to introduce the recoil energy Q defined by

$$Q(Q + 2m_e c^2) = (cq)^2 = c^2 (p^2 + p'^2 - 2pp' \cos \theta), \quad (\text{A.29})$$

where m_e is the electron rest mass and $\theta = \arccos(\hat{\mathbf{p}} \cdot \hat{\mathbf{p}}')$ is the scattering angle. Equivalently, we can write

$$Q = \sqrt{(cq)^2 + m_e^2 c^4} - m_e c^2. \quad (\text{A.30})$$

Notice that, when the collision is with a free electron at rest, the energy loss is completely transformed into kinetic energy of the recoiling electron, *i.e.*, $Q = W$. For collisions with bound electrons, the relation $Q \simeq W$ still holds for hard ionising collisions (that is, when the energy transfer W is much larger than the ionisation energy of the target electron so that binding effects are negligible).

The kinematically allowed recoil energies lie in the interval $Q_- < Q < Q_+$, with end points given by Eq. (A.29) with $\cos \theta = +1$ and -1 , respectively. That is

$$\begin{aligned} Q_{\pm} &= \sqrt{(cp \pm cp')^2 + m_e^2 c^4} - m_e c^2 \\ &= \sqrt{\left[\sqrt{E(E + 2mc^2)} \pm \sqrt{(E - W)(E - W + 2mc^2)} \right]^2 + m_e^2 c^4} - m_e c^2. \end{aligned} \quad (\text{A.31})$$

Notice that, for $W < E$, Q_+ is larger than W and $Q_- < W$. When $W \ll E$, expression (A.31) is not suitable for evaluating Q_- since it involves the subtraction of two similar quantities. In this case, it is more convenient to use the approximate relation

$$cp - cp' \simeq c \left(\frac{dp}{dE} W - \frac{1}{2} \frac{d^2 p}{dE^2} W^2 \right) = \frac{W}{\beta} \left(1 + \frac{1}{2\gamma(\gamma + 1)} \frac{W}{E} \right) \quad (\text{A.32})$$

and calculate Q_- as

$$Q_- \simeq \sqrt{(cp - cp')^2 + m_e^2 c^4} - m_e c^2 \quad (\text{A.33})$$

or, if $cp - cp' \ll m_e c^2$,

$$Q_- \simeq \frac{1}{2} \frac{(cp - cp')^2}{m_e c^2} - \frac{1}{8} \frac{(cp - cp')^4}{(m_e c^2)^3}. \quad (\text{A.34})$$

Thus, for $E \gg W$,

$$Q_-(Q_- + 2m_e c^2) \simeq W^2 / \beta^2. \quad (\text{A.35})$$

In the non-relativistic limit,

$$Q \equiv q^2 / 2m_e, \quad Q_{\pm} = [E^{1/2} \pm (E - W)^{1/2}]^2. \quad (\text{A.36})$$

From (A.31), it is clear that the curves $Q = Q_-(W)$ and $Q = Q_+(W)$ vary monotonously with W and intersect at $W = E$. Thus, they define a single continuous function $W = W_m(Q)$ in the interval $0 < Q < Q_+(0)$. By solving the Eqs. $Q = Q_{\pm}(W_m)$ we obtain

$$W_m(Q) = E + mc^2 - \sqrt{\left[\sqrt{E(E + 2mc^2)} - \sqrt{Q(Q + 2m_e c^2)} \right]^2 + m^2 c^4}, \quad (\text{A.37})$$

which, when $W \ll E$, reduces to

$$W_m(Q) \simeq \beta \sqrt{Q(Q + 2m_e c^2)}. \quad (\text{A.38})$$

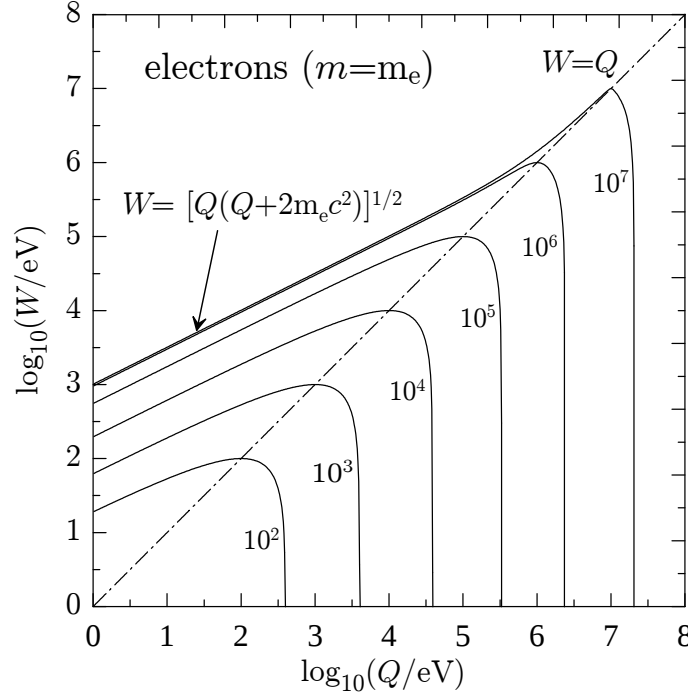


Figure A.3: Domains of kinematically allowed transitions in the (Q, W) plane for electrons/positrons. The curves represent the maximum allowed energy loss $W_m(Q)$, given by Eq. (A.37), for electrons with the indicated kinetic energies (in eV). When E increases, $W_m(Q)$ approaches the vacuum photon line, $W = [Q(Q + 2m_e c^2)]^{1/2}$, which is an absolute upper bound for the allowed energy losses.

Now it follows that, for given values of E and Q [$< Q_+(0)$], the only kinematically allowed values of the energy loss are those in the interval $0 < W < W_m(Q)$ (see Fig. A.3).

For a given energy loss W , the quantity

$$q_{\min} \equiv c^{-1} \sqrt{Q_-(Q_- + 2m_e c^2)}, \quad (\text{A.39})$$

is the minimum value of the momentum transfer in an inelastic collision, which occurs when $\theta = 0$. $q_{\min}$ is always larger than W/c . When the energy of the projectile increases, $\beta \rightarrow 1$ and $q_{\min}$ decreases approaching (but never reaching) the value W/c . It is worth recalling that a photon of energy W in vacuum has a linear momentum $q = W/c$ and, hence, interactions consisting of emission of bare photons would be located on the line $Q(Q + 2m_e c^2) = W^2$ of the (Q, W) plane, the so-called vacuum photon line. This line, lies outside the kinematically allowed region, *i.e.*, the “recoil” energy of the photon is less than Q_- (see Fig. A.3). Therefore, when the target is a single atom, the emission of photons by the projectile is not possible¹. When the energy E of the projectile increases,

¹In a condensed medium, ultrarelativistic projectiles can emit real photons (Cerenkov radiation) under certain, quite restricting circumstances (see, *e.g.*, Jackson, 1975).

Q_- decreases and tends to the photon line when β tends to unity. Hence, emission of photons by ultrarelativistic projectiles in low-density media is barely prevented by energy and momentum conservation. Generally speaking, as the interaction involves the exchange of a virtual photon, the DCS increases as the photon becomes more real, that is as we approach the photon line. For a dilute gas, this causes a gradual increase of the cross section with the projectile energy when $\beta \rightarrow 1$.

The scattering angle θ is related to the energy loss through

$$\cos \theta = \frac{(cp)^2 + (cp')^2 - Q(Q + 2m_e c^2)}{2(cp)(cp')}. \quad (\text{A.40})$$

The recoil angle θ_r between $\mathbf{p}$ and $\mathbf{q}$ is given by

$$\cos \theta_r = \frac{(cp)^2 - (cp')^2 + (cq)^2}{2(cp)(cq)}, \quad (\text{A.41})$$

which can also be written in the form

$$\cos^2 \theta_r = \frac{W^2/\beta^2}{Q(Q + 2m_e c^2)} \left(1 + \frac{Q(Q + 2m_e c^2) - W^2}{2W(E + mc^2)} \right)^2. \quad (\text{A.42})$$

For heavy ($m \gg m_e$) high-energy projectiles and collisions such that $Q \ll E$ and $W \ll E$,

$$\cos^2 \theta_r \simeq \frac{W^2/\beta^2}{Q(Q + 2m_e c^2)} \simeq \frac{Q_-(Q_- + 2m_e c^2)}{Q(Q + 2m_e c^2)}. \quad (\text{A.43})$$

Appendix B

Numerical tools

B.1 Cubic spline interpolation

In this Section we follow the presentation of Maron (1982). Suppose that a function $f(x)$ is given in numerical form, *i.e.*, as a table of values

$$f_i = f(x_i) \quad (i = 1, \dots, N). \quad (\text{B.1})$$

The points (knots) x_i do not need to be equispaced, but we assume that they are in (strictly) increasing order

$$x_1 < x_2 < \dots < x_N. \quad (\text{B.2})$$

A function $\varphi(x)$ is said to be an interpolating cubic spline if

- 1) It reduces to a cubic polynomial within each interval $[x_i, x_{i+1}]$, *i.e.*, if $x_i \leq x \leq x_{i+1}$

$$\varphi(x) = a_i + b_i x + c_i x^2 + d_i x^3 \equiv p_i(x) \quad (i = 1, \dots, N-1). \quad (\text{B.3})$$

- 2) The polynomial $p_i(x)$ matches the values of $f(x)$ at the endpoints of the i -th interval,

$$p_i(x_i) = f_i, \quad p_i(x_{i+1}) = f_{i+1} \quad (i = 1, \dots, N-1), \quad (\text{B.4})$$

so that $\varphi(x)$ is continuous in $[x_1, x_N]$.

- 3) The first and second derivatives of $\varphi(x)$ are continuous in $[x_1, x_N]$

$$p'_i(x_{i+1}) = p'_{i+1}(x_{i+1}) \quad (i = 1, \dots, N-2), \quad (\text{B.5})$$

$$p''_i(x_{i+1}) = p''_{i+1}(x_{i+1}) \quad (i = 1, \dots, N-2). \quad (\text{B.6})$$

Consequently, the curve $y = \varphi(x)$ interpolates the table (B.1) and has a continuously turning tangent.

To obtain the spline coefficients a_i, b_i, c_i, d_i ($i = 1, \dots, N-1$) we start from the fact that $\varphi''(x)$ is linear in $[x_i, x_{i+1}]$. Introducing the quantities

$$h_i \equiv x_{i+1} - x_i \quad (i = 1, \dots, N-1) \quad (\text{B.7})$$

and

$$\sigma_i \equiv \varphi''(x_i) \quad (i = 1, \dots, N), \quad (\text{B.8})$$

we can write the obvious identity

$$p_i''(x) = \sigma_i \frac{x_{i+1} - x}{h_i} + \sigma_{i+1} \frac{x - x_i}{h_i} \quad (i = 1, \dots, N-1). \quad (\text{B.9})$$

Notice that x_{i+1} must be larger than x_i in order to have $h_i > 0$. Integrating Eq. (B.9) twice with respect to x , gives for $i = 1, \dots, N-1$

$$p_i(x) = \sigma_i \frac{(x_{i+1} - x)^3}{6h_i} + \sigma_{i+1} \frac{(x - x_i)^3}{6h_i} + A_i(x - x_i) + B_i(x_{i+1} - x), \quad (\text{B.10})$$

where A_i and B_i are constants. These can be determined by introducing the expression (B.10) into eqs. (B.4), this gives the pair of eqs.

$$\sigma_i \frac{h_i^2}{6} + B_i h_i = f_i \quad \text{and} \quad \sigma_{i+1} \frac{h_i^2}{6} + A_i h_i = f_{i+1}. \quad (\text{B.11})$$

Finally, solving for A_i and B_i and substituting the result in (B.10), we obtain

$$\begin{aligned} p_i(x) = & \frac{\sigma_i}{6} \left[\frac{(x_{i+1} - x)^3}{h_i} - h_i(x_{i+1} - x) \right] + f_i \frac{x_{i+1} - x}{h_i} \\ & + \frac{\sigma_{i+1}}{6} \left[\frac{(x - x_i)^3}{h_i} - h_i(x - x_i) \right] + f_{i+1} \frac{x - x_i}{h_i}. \end{aligned} \quad (\text{B.12})$$

To be able to use $\varphi(x)$ to approximate $f(x)$, we must find the second derivatives σ_i ($i = 1, \dots, N$). To this end, we impose the conditions (B.5). Differentiating (B.12) gives

$$p_i'(x) = \frac{\sigma_i}{6} \left[-\frac{3(x_{i+1} - x)^2}{h_i} + h_i \right] + \frac{\sigma_{i+1}}{6} \left[\frac{3(x - x_i)^2}{h_i} - h_i \right] + \delta_i, \quad (\text{B.13})$$

where

$$\delta_i = \frac{y_{i+1} - y_i}{h_i}. \quad (\text{B.14})$$

Hence,

$$p_i'(x_{i+1}) = \sigma_i \frac{h_i}{6} + \sigma_{i+1} \frac{h_i}{3} + \delta_i, \quad (\text{B.15a})$$

$$p_i'(x_i) = -\sigma_i \frac{h_i}{3} - \sigma_{i+1} \frac{h_i}{6} + \delta_i \quad (\text{B.15b})$$

and, similarly,

$$p'_{i+1}(x_{i+1}) = -\sigma_{i+1}\frac{h_{i+1}}{3} - \sigma_{i+2}\frac{h_{i+1}}{6} + \delta_{i+1}. \quad (\text{B.15c})$$

Replacing (B.15a) and (B.15c) in (B.5), we obtain

$$h_i\sigma_i + 2(h_i + h_{i+1})\sigma_{i+1} + h_{i+1}\sigma_{i+2} = 6(\delta_{i+1} - \delta_i) \quad (i = 1, \dots, N-2). \quad (\text{B.16})$$

The system (B.16) is linear in the N unknowns σ_i ($i = 1, \dots, N$). However, since it contains only $N - 2$ equations, it is underdetermined. This means that we need either to add two additional (independent) equations or to fix arbitrarily two of the N unknowns. The usual practice is to adopt *endpoint strategies* that introduce constraints on the behaviour of $\varphi(x)$ near x_1 and x_N . An endpoint strategy fixes the values of σ_1 and σ_N , yielding an $(N - 2) \times (N - 2)$ system in the variables σ_i ($i = 2, \dots, N - 1$). The resulting system is, in matrix form,

$$\begin{pmatrix} H_2 & h_2 & 0 & \cdots & 0 & 0 & 0 \\ h_2 & H_3 & h_3 & \cdots & 0 & 0 & 0 \\ 0 & h_3 & H_4 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & H_{N-3} & h_{N-3} & 0 \\ 0 & 0 & 0 & \cdots & h_{N-3} & H_{N-2} & h_{N-2} \\ 0 & 0 & 0 & \cdots & 0 & h_{N-2} & H_{N-1} \end{pmatrix} \begin{pmatrix} \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \vdots \\ \sigma_{N-3} \\ \sigma_{N-2} \\ \sigma_{N-1} \end{pmatrix} = \begin{pmatrix} D_2 \\ D_3 \\ D_4 \\ \vdots \\ D_{N-3} \\ D_{N-2} \\ D_{N-1} \end{pmatrix}, \quad (\text{B.17})$$

where

$$H_i = 2(h_{i-1} + h_i) \quad (i = 2, \dots, N-1) \quad (\text{B.18})$$

and

$$\begin{aligned} D_2 &= 6(\delta_2 - \delta_1) - h_1\sigma_1 \\ D_i &= 6(\delta_i - \delta_{i-1}) \quad (i = 3, \dots, N-2) \\ D_{N-1} &= 6(\delta_{N-1} - \delta_{N-2}) - h_{N-1}\sigma_N. \end{aligned} \quad (\text{B.19})$$

(σ_1 and σ_N are removed from the first and last equations, respectively). The matrix of coefficients is symmetric, tridiagonal and diagonally dominant (the larger coefficients are in the diagonal), so that the system (B.17) can be easily (and accurately) solved by Gauss elimination. The spline coefficients a_i , b_i , c_i , d_i ($i = 1, \dots, N - 1$) —see Eq. (B.3)— can then be obtained by expanding the expressions (B.12):

$$\begin{aligned} a_i &= \frac{1}{6h_i} [\sigma_i x_{i+1}^3 - \sigma_{i+1} x_i^3 + 6(f_i x_{i+1} - f_{i+1} x_i)] + \frac{h_i}{6} (\sigma_{i+1} x_i - \sigma_i x_{i+1}), \\ b_i &= \frac{1}{2h_i} [\sigma_{i+1} x_i^2 - \sigma_i x_{i+1}^2 + 2(f_{i+1} - f_i)] + \frac{h_i}{6} (\sigma_i - \sigma_{i+1}), \\ c_i &= \frac{1}{2h_i} (\sigma_i x_{i+1} - \sigma_{i+1} x_i), \\ d_i &= \frac{1}{6h_i} (\sigma_{i+1} - \sigma_i). \end{aligned} \quad (\text{B.20})$$

When accurate values of $f''(x)$ are known, the best strategy is to set $\sigma_1 = f''(x_1)$ and $\sigma_N = f''(x_N)$, since this will minimise the spline interpolation errors near the endpoints x_1 and x_N . Unfortunately, the exact values $f''(x_1)$ and $f''(x_N)$ are not always available.

The so-called *natural spline* corresponds to taking $\sigma_1 = \sigma_N = 0$. It results in a $y = \varphi(x)$ curve with the shape that would be taken by a flexible rod (such as a draughtman's spline) if it were bent around pegs at the knots but allowed to maintain its natural (straight) shape outside the interval $[x_1, x_N]$. Since $\sigma_1 = \sigma_N = 0$, extrapolation of $\varphi(x)$ outside the interval $[x_1, x_N]$ by straight segments gives a continuous function with continuous first and second derivatives [*i.e.*, a cubic spline in $(-\infty, \infty)$].

The accuracy of the spline interpolation is mainly determined by the density of knots in the regions where $f(x)$ has strong variations. For constant, linear, quadratic and cubic functions the interpolation errors can be reduced to zero by using the exact values of σ_1 and σ_N (in these cases, however, the natural spline may introduce appreciable errors near the endpoints). It is important to keep in mind that a cubic polynomial has, at most, one inflexion point. As a consequence, we should have at least a knot between each pair of inflexion points of $f(x)$ to ensure proper interpolation. Special care must be taken when interpolating functions that have a practically constant value in a partial interval, since the spline tends to wiggle instead of staying constant. In this particular case, it may be more convenient to use linear interpolation.

Obviously, the interpolating cubic spline $\varphi(x)$ can be used not only to obtain interpolated values of $f(x)$ between the knots, but also to calculate integrals such as

$$\int_a^b f(x) dx \simeq \int_a^b \varphi(x) dx, \quad x_1 \leq a \quad \text{and} \quad b \leq x_N, \quad (\text{B.21})$$

analytically. It is worth noting that derivatives of $\varphi(x)$ other than the first one may differ significantly from those of $f(x)$.

To obtain the interpolated value $\varphi(x_c)$ —see Eq. (B.3)— of $f(x)$ at the point x_c , we must first determine the interval $(x_i, x_{i+1}]$ that contains the point x_c . To reduce the effort to locate the point, we use the following binary search algorithm:

- (i) Set $i = 1$ and $j = N$.
- (ii) Set $k = [(i + j)/2]$.
- (iii) If $x_k < x_c$, set $i = k$; otherwise set $j = k$.
- (iv) If $j - i > 1$, go to step (ii).
- (v) Deliver i .

Notice that the maximum delivered value of i is $N - 1$.

B.2 Numerical quadrature

In many cases, we need to calculate integrals of the form

$$\int_A^B f(z) \, dz, \quad (\text{B.22})$$

where the integrand is coded as an external function subprogram, which gives nominally exact values. These integrals are evaluated by using the Fortran 77 external function **SUMGA**, which implements the twenty-point Gauss method with an adaptive bipartition scheme to allow for error control. This procedure is comparatively fast and is able to deal even with functions that have integrable singularities located at the endpoints of the interval $[A, B]$, a quite exceptional feature.

B.2.1 Gauss integration

We use the twenty-point Gauss formula (see, *e.g.*, Abramowitz and Stegun, 1974), given by

$$\int_a^b f(z) \, dz = \frac{b-a}{2} \sum_{i=1}^{20} w_i f(z_i) \quad (\text{B.23})$$

with

$$z_i = \frac{b-a}{2} x_i + \frac{b+a}{2}. \quad (\text{B.24})$$

The abscissa x_i ($-1 < x_i < 1$) is the i -th zero of the Legendre polynomial $P_{20}(x)$, the weights w_i are defined as

$$w_i = \frac{2}{(1-x_i^2) [P'_{20}(x_i)]^2}. \quad (\text{B.25})$$

The numerical values of the abscissas and weights are given in Table B.1. The difference between the exact value of the integral and the right-hand side of Eq. (B.23) is

$$\Delta_{20} = \frac{(b-a)^{41} (20!)^4}{41 (40!)^3} f^{(40)}(\xi), \quad (\text{B.26})$$

where ξ is a point in the interval $[a, b]$.

The Gauss method gives an estimate of the integral of $f(z)$ over the interval $[a, b]$, which is obtained as a weighted sum of function values at fixed points inside the interval. We point out that (B.23) is an open formula, *i.e.*, the value of the function at the endpoints of the interval is never required. Owing to this fact, function **SUMGA** can integrate functions that are singular at the endpoints. As an example, the integral of $f(x) = x^{-1/2}$ over the interval $[0, 1]$ is correctly evaluated. This would not be possible with a method based on a closed formula (*i.e.*, one that uses the values of the integrand at the interval endpoints).

Table B.1: Abscissas and weights for twenty-point Gauss integration.

$\pm x_i$	w_i
7.6526521133497334D-02	1.5275338713072585D-01
2.2778585114164508D-01	1.4917298647260375D-01
3.7370608871541956D-01	1.4209610931838205D-01
5.1086700195082710D-01	1.3168863844917663D-01
6.3605368072651503D-01	1.1819453196151842D-01
7.4633190646015079D-01	1.0193011981724044D-01
8.3911697182221882D-01	8.3276741576704749D-02
9.1223442825132591D-01	6.2672048334109064D-02
9.6397192727791379D-01	4.0601429800386941D-02
9.9312859918509492D-01	1.7614007139152118D-02

B.2.2 Adaptive bipartition

Function **SUMGA** exploits the fact that the error Δ_{20} , Eq. (B.26), of the calculated integral decreases when the interval length is reduced. Thus, halving the interval and applying the Gauss method to each of the two subintervals gives a much better estimate of the integral, provided only that the function $f(x)$ is smooth enough over the initial interval. Notice that the error decreases by a factor of about 2^{-40} (!).

The algorithm implemented in **SUMGA** is as follows. The integration interval (A, B) is successively halved so that each iteration gives a doubly finer partition of the initial interval. We use the term “ n -subinterval” to denote the subintervals obtained in the n -th iteration. In each iteration, the integrals over the different n -subintervals are evaluated by the Gauss method, Eq. (B.23). Consider that the integral over a given n -subinterval is S_1 . In the following iteration, this n -subinterval is halved and the integrals over each of the two resulting $(n + 1)$ -subintervals are evaluated, giving values S_{1a} and S_{1b} . If $S'_1 = S_{1a} + S_{1b}$ differs from S_1 in less than the selected tolerance, S'_1 is the sought value of the integral in the considered n -subinterval; the value S'_1 is then accumulated and this n -subinterval is no longer considered in subsequent iterations. Each iteration is likely to produce new holes (eliminated subintervals) in the regions where the function is smoother and, hence, the numerical effort progressively concentrates in the regions where $f(x)$ has stronger variations. The calculation terminates when the exploration of the interval (A, B) has been successfully completed or when a clear indication of an anomalous behaviour of $f(x)$ is found (*e.g.*, when there is a persistent increase of the number of remaining n -subintervals in each iteration). In the second case a warning message is printed in unit 26 and the control is returned to the calling program.

Appendix C

Electron/positron transport in electromagnetic fields

In this appendix, we consider the transport of electrons/positrons in static external electromagnetic (EM) fields, in vacuum and in condensed media. We assume that, in the region where particles move, there is an electric field $\mathcal{E}$ and a magnetic field $\mathcal{B}$, which are set up by external sources and do not vary with time. For practical purposes, we also consider that both $\mathcal{E}$ and $\mathcal{B}$ are continuous functions of the position vector $\mathbf{r}$.

The interactions with the medium will be described by means of PENELOPE. In each individual interaction event, the electron/positron loses a discrete amount of kinetic energy and changes its direction of motion. In the absence of EM fields, the electron travels freely between consecutive interaction events, *i.e.*, following a straight trajectory segment at constant speed. To simulate electron transport with static external EM fields, we assume that the interaction properties of electrons with the medium are not substantially affected by the field. Consequently, to account for the effect of the EM field, we only need to consider that along each “free flight” the electron is driven by the EM force. With a proper selection of the simulation parameters (*i.e.*, the energy loss and angular cutoff values), trajectory segments may have macroscopic lengths. Therefore, in material media it is appropriate to consider the macroscopic EM fields $\mathbf{D}$ and $\mathbf{H}$ rather than the microscopic fields $\mathcal{E}$ and $\mathcal{B}$.

It should be noted that, under the action of an electric field, the kinetic energy of the electron can vary substantially along a single trajectory segment. This conflicts with one of the basic assumptions in PENELOPE, namely that the energy of the particle stays practically constant along the segment. In practice, however, we can always limit the maximum segment length by means of the parameter $s_{\max}$. Then, the effect of the EM field can be treated independently of that of the interactions with the medium. In other words, for simulation purposes, we only need an efficient method to generate particle trajectories in the EM field *in vacuum*. It is also important to recall that strong electric fields in material media accelerate unbound charged particles, even when they are at rest (*i.e.*, electrons are never absorbed, simulated tracks can only terminate when they

leave the field volume). Injection of a single electron in the medium may give rise to a complex cascade of delta rays, that accelerate in the direction opposite to the electric field. To describe these cascades we need accurate cross sections for ionisation of outer atomic shells by impact of low-energy electrons, much more accurate than the simple ones implemented in PENELOPE. Therefore, PENELOPE *is not* expected to yield a reliable description of this process. The simulation algorithm described here is applicable only to magnetic fields and, cautiously, to weak electric fields. Notice also that we disregard the emission of radiation by the charged particle when it is accelerated by the external EM field (see, *e.g.*, Jackson, 1975); this approximation is not valid for very strong magnetic and electric fields.

C.1 Tracking particles in vacuum.

Let us begin by describing a “brute force” method to calculate trajectories of charged particles in arbitrary static electric and magnetic fields in vacuum. We start from the Lorentz force equation¹ for an electron ($Z_0 = -1$) or positron ($Z_0 = +1$),

$$\frac{d\mathbf{p}}{dt} = Z_0 e \left(\boldsymbol{\mathcal{E}} + \frac{\mathbf{v}}{c} \times \boldsymbol{\mathcal{B}} \right), \quad (\text{C.1})$$

which we write as

$$\frac{d(\gamma\beta\hat{\mathbf{v}})}{dt} = \frac{Z_0 e}{m_e c} (\boldsymbol{\mathcal{E}} + \beta\hat{\mathbf{v}} \times \boldsymbol{\mathcal{B}}), \quad (\text{C.2})$$

with $\hat{\mathbf{v}} = \mathbf{v}/v$, $\beta = v/c$ and $\gamma = (1 - \beta^2)^{-1/2}$. We note that

$$\frac{d(\gamma\beta\hat{\mathbf{v}})}{dt} = \gamma^3 \frac{d\beta}{dt} \hat{\mathbf{v}} + \gamma\beta \frac{d\hat{\mathbf{v}}}{dt} \quad (\text{C.3})$$

where the vectors $\hat{\mathbf{v}}$ and $d\hat{\mathbf{v}}/dt$ are orthogonal. Then, projecting Eq. (C.2) into the directions of these two vectors, we obtain

$$\frac{d\beta}{dt} = \frac{Z_0 e}{m_e c \gamma} (1 - \beta^2) (\boldsymbol{\mathcal{E}} \cdot \hat{\mathbf{v}}) \quad (\text{C.4})$$

and

$$\frac{d\hat{\mathbf{v}}}{dt} = \frac{Z_0 e}{m_e c \beta \gamma} [\boldsymbol{\mathcal{E}} - (\boldsymbol{\mathcal{E}} \cdot \hat{\mathbf{v}}) \hat{\mathbf{v}} + \beta \hat{\mathbf{v}} \times \boldsymbol{\mathcal{B}}]. \quad (\text{C.5})$$

It then follows that

$$\begin{aligned} \frac{d\beta\hat{\mathbf{v}}}{dt} &= \frac{d\beta}{dt} \hat{\mathbf{v}} + \beta \frac{d\hat{\mathbf{v}}}{dt} \\ &= \frac{Z_0 e}{m_e c \gamma} [\boldsymbol{\mathcal{E}} - \beta^2 (\boldsymbol{\mathcal{E}} \cdot \hat{\mathbf{v}}) \hat{\mathbf{v}} + \beta \hat{\mathbf{v}} \times \boldsymbol{\mathcal{B}}], \end{aligned} \quad (\text{C.6})$$

¹In this appendix, electromagnetic quantities are expressed in the Gaussian system of units.

which we cast in the form

$$\frac{d\mathbf{v}}{dt} = \mathbf{A}, \quad \mathbf{A} \equiv \frac{Z_0 e}{m_e \gamma} [\boldsymbol{\mathcal{E}} - \beta^2 (\boldsymbol{\mathcal{E}} \cdot \hat{\mathbf{v}}) \hat{\mathbf{v}} + \beta \hat{\mathbf{v}} \times \boldsymbol{\mathcal{B}}]. \quad (\text{C.7})$$

Notice that, for arbitrary fields $\boldsymbol{\mathcal{E}}$ and $\boldsymbol{\mathcal{B}}$, the “acceleration” $\mathbf{A}$ is a function of the particle’s position $\mathbf{r}$, energy E and direction of motion $\hat{\mathbf{v}}$.

Implicit integration of Eq. (C.7) gives the equations of motion

$$\mathbf{v}(t) = \mathbf{v}_0 + \int_0^t \mathbf{A}(\mathbf{r}(t'), E(t'), \hat{\mathbf{v}}(t')) dt', \quad (\text{C.8})$$

$$\mathbf{r}(t) = \mathbf{r}_0 + \int_0^t \mathbf{v}(t') dt'. \quad (\text{C.9})$$

Evidently, these equations are too complex for straight application in a simulation code and we must have recourse to approximate solution methods. We shall adopt the approach proposed by Bielajew (1988), which is well suited to transport simulations. The basic idea is to split the trajectory into a number of conveniently short steps such that *the acceleration $\mathbf{A}$ does not change much over the course of a step*. Along each step, we then have

$$\mathbf{v}(t) = \mathbf{v}_0 + t \mathbf{A}(\mathbf{r}_0, E_0, \hat{\mathbf{v}}_0) \quad (\text{C.10})$$

$$\mathbf{r}(t) = \mathbf{r}_0 + t \mathbf{v}_0 + t^2 \frac{1}{2} \mathbf{A}(\mathbf{r}_0, E_0, \hat{\mathbf{v}}_0), \quad (\text{C.11})$$

where the subscript “0” indicates values of the various quantities at the starting point ($t = 0$). The travelled path length s and the flying time t are related by

$$t = \int_0^s \frac{ds'}{v}, \quad (\text{C.12})$$

which to first order becomes

$$t = s/v_0. \quad (\text{C.13})$$

Then, to first order in the electromagnetic force,

$$\begin{aligned} \mathbf{v}(s) &= \mathbf{v}_0 + s \frac{\mathbf{A}(\mathbf{r}_0, E_0, \hat{\mathbf{v}}_0)}{c\beta_0} \\ \mathbf{r}(s) &= \mathbf{r}_0 + s \hat{\mathbf{v}}_0 + s^2 \frac{1}{2} \frac{\mathbf{A}(\mathbf{r}_0, E_0, \hat{\mathbf{v}}_0)}{c^2 \beta_0^2}. \end{aligned}$$

That is,

$$\mathbf{r}(s) = \mathbf{r}_0 + s \hat{\mathbf{v}}_0 + s^2 \frac{1}{2} \frac{Z_0 e [\boldsymbol{\mathcal{E}}_0 - \beta_0^2 (\boldsymbol{\mathcal{E}}_0 \cdot \hat{\mathbf{v}}_0) \hat{\mathbf{v}}_0 + \beta_0 \hat{\mathbf{v}}_0 \times \boldsymbol{\mathcal{B}}_0]}{m_e c^2 \gamma_0 \beta_0^2}. \quad (\text{C.14})$$

The particle’s velocity can be calculated directly from Eq. (C.10), which to first order gives

$$\mathbf{v}(s) = \mathbf{v}_0 + \Delta \mathbf{v} \quad (\text{C.15})$$

with

$$\Delta \mathbf{v} = s \frac{Z_0 e [\boldsymbol{\mathcal{E}}_0 - \beta_0^2 (\boldsymbol{\mathcal{E}}_0 \cdot \hat{\mathbf{v}}_0) \hat{\mathbf{v}}_0 + \beta_0 \hat{\mathbf{v}}_0 \times \boldsymbol{\mathcal{B}}_0]}{m_e c \gamma_0 \beta_0}. \quad (\text{C.16})$$

In the tracking algorithm, the velocity is used to determine the direction vector at the end of the step,

$$\hat{\mathbf{v}}(s) = \frac{\mathbf{v}_0 + \Delta \mathbf{v}}{|\mathbf{v}_0 + \Delta \mathbf{v}|}. \quad (\text{C.17})$$

Owing to the action of the electromagnetic force, the kinetic energy E of the particle varies along the step. As the trajectory is accurate only to first order, it is not advisable to compute the kinetic energy from the velocity of the particle. It is preferable to calculate $E(t)$ as

$$E(s) = E_0 + Z_0 e [\varphi(\mathbf{r}_0) - \varphi(\mathbf{r}(s))] \quad (\text{C.18})$$

where $\varphi(\mathbf{r})$ is the electrostatic potential, $\boldsymbol{\mathcal{E}} = -\nabla \varphi$. Notice that this ensures energy conservation, *i.e.*, it gives the exact energy variation in going from the initial to the final position.

This tracking method is valid only if

1) the fields do not change too much along the step

$$\frac{|\boldsymbol{\mathcal{E}}(\mathbf{r}(s)) - \boldsymbol{\mathcal{E}}(\mathbf{r}_0)|}{|\boldsymbol{\mathcal{E}}(\mathbf{r}_0)|} < \delta_{\boldsymbol{\mathcal{E}}} \ll 1, \quad \frac{|\boldsymbol{\mathcal{B}}(\mathbf{r}(s)) - \boldsymbol{\mathcal{B}}(\mathbf{r}_0)|}{|\boldsymbol{\mathcal{B}}(\mathbf{r}_0)|} < \delta_{\boldsymbol{\mathcal{B}}} \ll 1 \quad (\text{C.19})$$

and

2) the relative changes in kinetic energy and velocity (or direction of motion) are small

$$\left| \frac{E(s) - E_0}{E_0} \right| < \delta_E \ll 1, \quad \frac{|\Delta \mathbf{v}|}{v_0} < \delta_v \ll 1. \quad (\text{C.20})$$

These conditions set an upper limit on the allowed step length, $s_{\max}$, which depends on the local fields *and* on the energy and direction of the particle. The method is robust, in the sense that it converges to the exact trajectory when the maximum allowed step length tends to zero. In practical calculations, we shall specify the values of the δ -parameters (which should be of the order of 0.05 or less) and consider step lengths consistent with the above conditions. Thus, the smallness of the δ -parameters determines the accuracy of the generated trajectories.

To test the accuracy of a tracking algorithm, it is useful to consider the special cases of a uniform electric field (with $\boldsymbol{\mathcal{B}} = 0$) and a uniform magnetic field (with $\boldsymbol{\mathcal{E}} = 0$), which admit relatively simple analytical solutions of the equations of motion.

C.1.1 Uniform electric fields

Let us study first the case of a uniform electric field $\boldsymbol{\mathcal{E}}$. The equation of the trajectory of an electron/positron that starts at $t = 0$ from the point $\mathbf{r}_0$ with velocity $\mathbf{v}_0$ can be

expressed in the form (adapted from Bielaew, 1988)

$$\mathbf{r}(t) = \mathbf{r}_0 + t\mathbf{v}_{0\perp} + \frac{1}{a} \left[\cosh(act) - 1 + \frac{v_{0\parallel}}{c} \sinh(act) \right] \hat{\mathcal{E}}, \quad (\text{C.21})$$

where $\mathbf{v}_{0\parallel}$ and $\mathbf{v}_{0\perp}$ are the components of $\mathbf{v}_0$ parallel and perpendicular to the direction of the field,

$$\mathbf{v}_{0\parallel} = (\mathbf{v}_0 \cdot \hat{\mathcal{E}}) \hat{\mathcal{E}}, \quad \mathbf{v}_{0\perp} = \mathbf{v}_0 - (\mathbf{v}_0 \cdot \hat{\mathcal{E}}) \hat{\mathcal{E}} \quad (\text{C.22})$$

and

$$a \equiv \frac{Z_0 e \mathcal{E}}{m_e c^2 \gamma_0} = \frac{Z_0 e \mathcal{E}}{E_0}. \quad (\text{C.23})$$

The velocity of the particle is

$$\begin{aligned} \mathbf{v}(t) &= \mathbf{v}_{0\perp} + [c \sinh(act) + v_{0\parallel} \cosh(act)] \hat{\mathcal{E}} \\ &= \mathbf{v}_0 + \{c \sinh(act) + v_{0\parallel} [\cosh(act) - 1]\} \hat{\mathcal{E}}. \end{aligned} \quad (\text{C.24})$$

Since the scalar potential for the constant field is $\varphi(\mathbf{r}) = -\mathcal{E} \cdot \mathbf{r}$, the kinetic energy of the particle varies with time and is given by

$$E(t) = E_0 - Z_0 e \mathcal{E} \cdot [\mathbf{r}_0 - \mathbf{r}(t)]. \quad (\text{C.25})$$

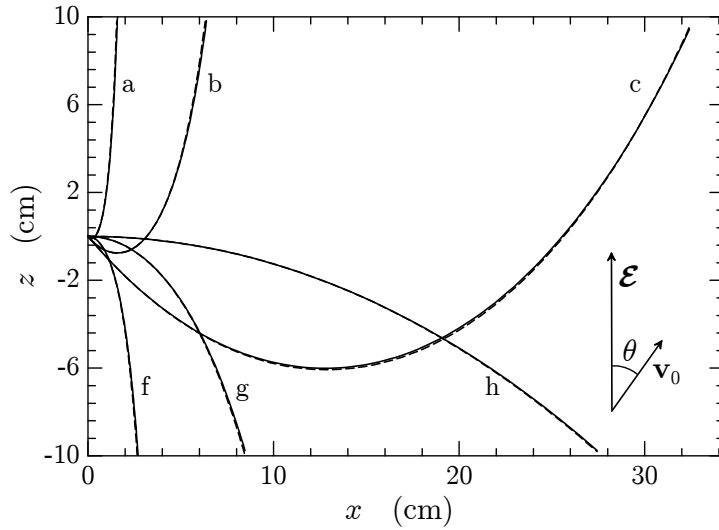


Figure C.1: Trajectories of electrons and positrons in a uniform electric field of 511 kV/cm. Continuous curves represent exact trajectories obtained from Eq. (C.21). The dashed lines are obtained by using the first-order numerical tracking method described by Eqs. (C.14)-(C.20) with $\delta \mathcal{E} = \delta_E = \delta_v = 0.02$. The displayed trajectories correspond to the following cases. a: positrons, $E_0 = 0.1$ MeV, $\theta = 135$ deg. b: positrons, $E_0 = 1$ MeV, $\theta = 135$ deg. c: positrons, $E_0 = 10$ MeV, $\theta = 135$ deg. f: electrons, $E_0 = 0.2$ MeV, $\theta = 90$ deg. g: electrons, $E_0 = 2$ MeV, $\theta = 90$ deg. h: electrons, $E_0 = 20$ MeV, $\theta = 90$ deg.

Figure C.1 displays trajectories of electrons and positrons with various initial energies and directions of motion in a uniform electric field of 511 kV/cm directed along the positive z -axis. Particles start from the origin ($\mathbf{r}_0 = 0$), with initial velocity in the xz -plane forming an angle θ with the field, *i.e.*, $\mathbf{v}_0 = (\sin \theta, 0, \cos \theta)$, so that the whole trajectories lie in the xz -plane. Continuous curves represent exact trajectories obtained from the analytical formula (C.21). The dashed curves are the results from the first-order tracking algorithm described above [Eqs. (C.14)-(C.20)] with $\delta_{\mathcal{E}} = \delta_E = \delta_v = 0.02$. We show three positron trajectories with initial energies of 0.1, 1 and 10 MeV, initially moving in the direction $\theta = 135$ deg. Three trajectories of electrons that initially move perpendicularly to the field ($\theta = 90$ deg) with energies of 0.2, 2 and 20 MeV are also depicted. We see that the tracking algorithm gives quite accurate results. The error can be further reduced, if required, by using shorter steps, *i.e.*, smaller δ -values.

C.1.2 Uniform magnetic fields

We now consider the motion of an electron/positron, with initial position $\mathbf{r}_0$ and velocity $\mathbf{v}_0$, in a uniform magnetic field $\mathbf{B}$. Since the magnetic force is perpendicular to the velocity, the field does not alter the energy of the particle and the speed $v(t) = v_0$ is a constant of the motion. It is convenient to introduce the precession frequency vector $\boldsymbol{\omega}$, defined by (notice the sign)

$$\boldsymbol{\omega} \equiv -\frac{Z_0 e \mathbf{B}}{m_e \gamma c} = -\frac{Z_0 e c \mathbf{B}}{E_0}, \quad (\text{C.26})$$

and split the velocity $\mathbf{v}$ into its components parallel and perpendicular to $\boldsymbol{\omega}$,

$$\mathbf{v}_{\parallel} = (\mathbf{v} \cdot \hat{\boldsymbol{\omega}}) \hat{\boldsymbol{\omega}}, \quad \mathbf{v}_{\perp} = \mathbf{v} - (\mathbf{v} \cdot \hat{\boldsymbol{\omega}}) \hat{\boldsymbol{\omega}}. \quad (\text{C.27})$$

Then, the equation of motion (C.7) becomes

$$\frac{d\mathbf{v}_{\parallel}}{dt} = 0, \quad \frac{d\mathbf{v}_{\perp}}{dt} = \boldsymbol{\omega} \times \mathbf{v}_{\perp}. \quad (\text{C.28})$$

The first of these equations says that the particle moves with constant velocity $\mathbf{v}_{0\parallel}$ along the direction of the magnetic field. From the second equation we see that, in the plane perpendicular to $\mathbf{B}$, the particle describes a circle with angular frequency ω and speed $v_{0\perp}$ (which is a constant of the motion). The radius of the circle is $R = v_{0\perp}/\omega$. That is, the trajectory is an helix with central axis along the $\mathbf{B}$ direction, radius R and pitch angle $\alpha = \arctan(v_{0\parallel}/v_{0\perp})$. The helix is right-handed for electrons and left-handed for positrons (see Fig. C.2).

In terms of the path length $s = tv_0$, the equation of motion takes the form

$$\mathbf{r}(s) = \mathbf{r}_0 + \frac{s}{v_0} \mathbf{v}_{0\parallel} + R [1 - \cos(s_{\perp}/R)] (\hat{\boldsymbol{\omega}} \times \hat{\mathbf{v}}_{0\perp}) + R \sin(s_{\perp}/R) \hat{\mathbf{v}}_{0\perp}, \quad (\text{C.29})$$

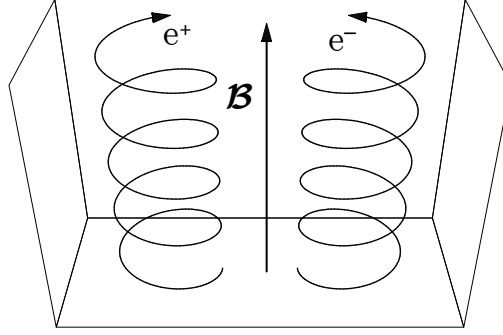


Figure C.2: Trajectories of electrons and positrons in a uniform magnetic field. The two particles start from the base plane with equal initial velocities.

where $\hat{\mathbf{v}}_{0\perp} \equiv \mathbf{v}_{0\perp}/v_{0\perp}$ and $s_{\perp} = sv_{0\perp}/v_0$. Equivalently,

$$\mathbf{r}(s) = \mathbf{r}_0 + s\hat{\mathbf{v}}_0 - \frac{s}{v_0}\mathbf{v}_{0\perp} + \frac{1}{\omega} [1 - \cos(s\omega/v_0)] (\hat{\omega} \times \mathbf{v}_{0\perp}) + \frac{1}{\omega} \sin(s\omega/v_0) \mathbf{v}_{0\perp}. \quad (\text{C.30})$$

After the path length s , the particle velocity is

$$\mathbf{v}(s) = v_0 \frac{d\mathbf{r}}{ds} = \mathbf{v}_0 + [\cos(s\omega/v_0) - 1] \mathbf{v}_{0\perp} + \sin(s\omega/v_0) (\hat{\omega} \times \mathbf{v}_{0\perp}). \quad (\text{C.31})$$

In Fig. C.3 we compare exact trajectories of electrons and positrons in a uniform magnetic field obtained from the analytical formula (C.30) with results from the first-order tracking algorithm [Eqs. (C.14)-(C.20)] with $\delta_{\mathbf{B}} = \delta_E = \delta_v = 0.02$. The field strength is 0.2 tesla. The depicted trajectories correspond to 0.5 MeV electrons (a) and 3 MeV positrons (b) that initially move in a direction forming an angle of 45 deg with the field. We see that the numerical algorithm is quite accurate for small path lengths, but it deteriorates rapidly for increasing s . In principle, the accuracy of the algorithm can be improved by reducing the value of δ_v , *i.e.*, the length of the step length. In practice, however, this is not convenient because it implies a considerable increase of numerical work, which can be easily avoided.

C.2 Exact tracking in homogeneous magnetic fields

In our first-order tracking algorithm [see Eqs. (C.14) and (C.16)], the effects of the electric and magnetic fields are uncoupled, *i.e.*, they can be evaluated separately. For uniform electric fields, the algorithm offers a satisfactory solution since it usually admits relatively large step lengths. In the case of uniform magnetic fields (with $\mathcal{E} = 0$), the kinetic energy is a constant of the motion and the only effective constraint on the step length is that the change in direction $|\Delta\mathbf{v}|/v_0$ has to be small. Since the particle trajectories on the plane perpendicular to the field $\mathbf{B}$ are circles and the first-order algorithm generates each step as a parabolic segment, we need to move in sub-steps of

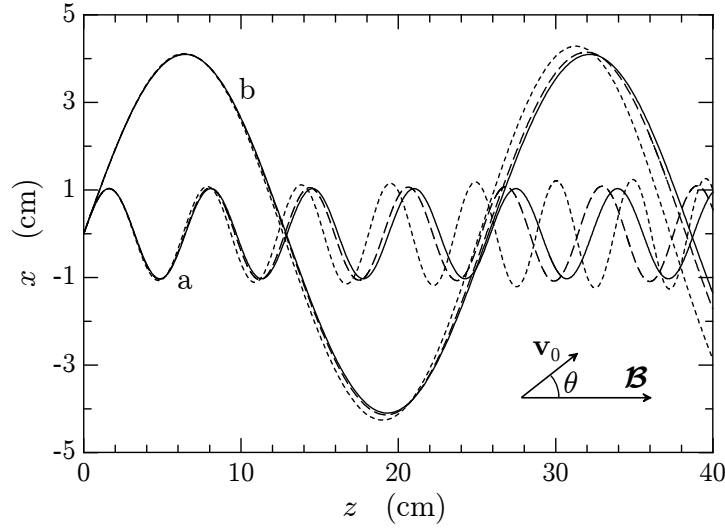


Figure C.3: Trajectories of electrons and positrons in a uniform magnetic field of 0.2 tesla. Continuous curves are exact trajectories calculated from Eq. (C.30). The short-dashed lines are obtained by using the numerical tracking method described in the text with $\delta_v = 0.02$. Long-dashed curves are the results from the tracking algorithm with $\delta_v = 0.005$. a: electrons, $E_0 = 0.5$ MeV, $\theta = 45$ deg. b: positrons, $E_0 = 3$ MeV, $\theta = 45$ deg.

length much less than the radius R (*i.e.*, δ_v must be given a very small value) and this makes the calculation slow. On the other hand, the action of the uniform magnetic field is described by simple analytical expressions [Eqs. (C.30) and (C.31)], that are amenable for direct use in the simulation code. These arguments suggest the following obvious modification of the tracking algorithm.

As before, we assume that the fields are essentially constant along each trajectory step and write

$$\mathbf{r}(s) = \mathbf{r}_0 + s\hat{\mathbf{v}}_0 + (\Delta\mathbf{r})_{\mathcal{E}} + (\Delta\mathbf{r})_{\mathcal{B}}, \quad (\text{C.32})$$

where $(\Delta\mathbf{r})_{\mathcal{E}}$ and $(\Delta\mathbf{r})_{\mathcal{B}}$ are the displacements caused by the electric and magnetic fields, respectively. For $(\Delta\mathbf{r})_{\mathcal{E}}$ we use the first-order approximation [see Eq. (C.14)],

$$(\Delta\mathbf{r})_{\mathcal{E}} = s^2 \frac{1}{2} \frac{Z_0 e [\mathcal{E}_0 - \beta_0^2 (\mathcal{E}_0 \cdot \hat{\mathbf{v}}_0) \hat{\mathbf{v}}_0]}{m_e c^2 \gamma_0 \beta_0^2}. \quad (\text{C.33})$$

The displacement caused by the magnetic field is evaluated using the result (C.30), *i.e.*,

$$(\Delta\mathbf{r})_{\mathcal{B}} = -\frac{s}{v_0} \mathbf{v}_{0\perp} + \frac{1}{\omega} [1 - \cos(s\omega/v_0)] (\hat{\boldsymbol{\omega}} \times \mathbf{v}_{0\perp}) + \frac{1}{\omega} \sin(s\omega/v_0) \mathbf{v}_{0\perp} \quad (\text{C.34})$$

with

$$\omega \equiv -\frac{Z_0 e c \mathcal{B}_0}{E_0}, \quad \text{and} \quad \mathbf{v}_{0\perp} = \mathbf{v}_0 - (\mathbf{v}_0 \cdot \hat{\boldsymbol{\omega}}) \hat{\boldsymbol{\omega}}. \quad (\text{C.35})$$

Similarly, the particle velocity along the step is expressed as

$$\mathbf{v}(s) = \mathbf{v}_0 + (\Delta\mathbf{v})_{\mathcal{E}} + (\Delta\mathbf{v})_{\mathcal{B}} \quad (\text{C.36})$$

with [see Eqs. (C.16) and (C.31)]

$$(\Delta \mathbf{v})_{\mathcal{E}} = s \frac{Z_0 e [\mathcal{E}_0 - \beta_0^2 (\mathcal{E}_0 \cdot \hat{\mathbf{v}}_0) \hat{\mathbf{v}}_0]}{m_e c \gamma_0 \beta_0} \quad (\text{C.37})$$

and

$$(\Delta \mathbf{v})_{\mathcal{B}} = [\cos(s\omega/v_0) - 1] \mathbf{v}_{0\perp} + \sin(s\omega/v_0) (\hat{\boldsymbol{\omega}} \times \mathbf{v}_{0\perp}). \quad (\text{C.38})$$

In our implementation of this tracking algorithm, the allowed step lengths s are limited by the following constraints [see Eqs. (C.19) and (C.20)]

$$\frac{|\mathcal{E}(\mathbf{r}(s)) - \mathcal{E}(\mathbf{r}_0)|}{|\mathcal{E}(\mathbf{r}_0)|} < \delta_{\mathcal{E}} \ll 1, \quad \frac{|\mathcal{B}(\mathbf{r}(s)) - \mathcal{B}(\mathbf{r}_0)|}{|\mathcal{B}(\mathbf{r}_0)|} < \delta_{\mathcal{B}} \ll 1 \quad (\text{C.39})$$

and

$$\left| \frac{E(s) - E_0}{E_0} \right| < \delta_E \ll 1, \quad \frac{|(\Delta \mathbf{v})_{\mathcal{E}} + (\Delta \mathbf{v})_{\mathcal{B}}|}{v_0} < \delta_v \ll 1. \quad (\text{C.40})$$

The algorithm is robust, *i.e.*, the accuracy of the generated trajectories increases when the δ -parameters are reduced. In many practical cases, a good compromise between accuracy and simulation speed is obtained by setting $\delta_{\mathcal{E}} = \delta_{\mathcal{B}} = \delta_E = \delta_v = 0.02$. Notice that, in the case of a uniform magnetic field, the tracking algorithm is now exact, irrespective of the step length.

This tracking algorithm has been implemented in the subroutine package `PENFIELD`, which is devised to work linked to `PENELOPE` and `PENGEO`. To simulate radiation transport in a given field/material configuration, the user must provide the steering main program as well as specific routines that define the EM field (see the examples and comments in the source file `penfield.f`).

Bibliography

- Abramowitz M. and I.A. Stegun (1974), eds., *Handbook of Mathematical Functions* (Dover, New York).
- Acosta E., X. Llovet, E. Coleoni, J.A. Riveros and F. Salvat (1998), “Monte Carlo simulation of x-ray emission by kilovolt electron bombardment”, *J. Appl. Phys.* **83**, 6038–6049.
- Agostinelli S., *et al.*(2003), “Geant4—a simulation toolkit”, *Nucl. Instrum. Meth. A* **506**, 250–303.
- Allison J., *et al.*(2006), “Geant4 developments and applications”, *IEEE Trans. Nucl. Sci.* **53**, 270–278.
- Andreo P. (1991), “Monte Carlo techniques in medical radiation physics”, *Phys. Med. Biol.* **36**, 861–920.
- Andreo P. and A. Brahme (1984), “Restricted energy-loss straggling and multiple scattering of electrons in mixed Monte Carlo procedures”, *Rad. Res.* **100**, 16–29.
- Baró J., M. Roteta, J.M. Fernández-Varea and F. Salvat (1994a), “Analytical cross sections for Monte Carlo simulation of photon transport”, *Radiat. Phys. Chem.* **44**, 531–552.
- Baró J., J. Sempau, J.M. Fernández-Varea and F. Salvat (1994b), “Simplified Monte Carlo simulation of elastic electron scattering in limited media”, *Nucl. Instrum. Meth. B* **84**, 465–483.
- Baró J., J. Sempau, J.M. Fernández-Varea and F. Salvat (1995), “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.
- Bearden J.A. (1967), “X-ray wavelengths”, *Rev. Mod. Phys.* **39**, 78–124.
- Benedito E., J.M. Fernández-Varea and F. Salvat (2001), “Mixed simulation of the multiple elastic scattering of electrons and positrons using partial-wave differential cross sections”, *Nucl. Instrum. Meth. B* **174**, 91–110.
- Berger M.J. (1963), “Monte Carlo calculation of the penetration and diffusion of fast charged particles”, in *Methods in Computational Physics*, vol. 1, eds. B. Alder, S. Fernbach and M. Rotenberg (Academic Press, New York) pp. 135–215.
- Berger M.J. (1992), “ESTAR, PSTAR and ASTAR: computer programs for calculating stopping-power and range tables for electrons, protons and helium ions”, *Report*

- NISTIR 4999* (National Institute of Standards and Technology, Gaithersburg, MD).
- Berger M.J. (1998), “Applicability of the condensed-random-walk Monte Carlo method at low energies in high- Z materials”, *Radiat. Phys. Chem.* **53**, 191–203.
- Berger M.J. and J.H. Hubbell (1987), “XCOM: Photon Cross Sections on a Personal Computer”, *Report NBSIR 87-3597* (National Bureau of Standards, Gaithersburg, MD).
- Berger M.J. and S.M. Seltzer (1972), “Response functions for sodium iodide scintillation detectors”, *Nucl. Instrum. Meth.* **104**, 317–332.
- Berger M.J. and S.M. Seltzer (1982), “Stopping Power of Electrons and Positrons”, *Report NBSIR 82-2550* (National Bureau of Standards, Gaithersburg, MD).
- Berger M.J. and S.M. Seltzer (1988), chapters 7, 8 and 9, in *Monte Carlo Transport of Electrons and Photons*, eds. T.M. Jenkins, W.R. Nelson and A. Rindi (Plenum, New York).
- Bethe, H.A. (1930), “Zur Theorie des Durchgangs schneller Korpuskularstrahlen durch Materie”, *Ann. Physik* **5**, 325–400.
- Bethe, H.A. (1932), “Bremsformel für Elektronen relativistischer Geschwindigkeit”, *Z. Physik* **76**, 293–299.
- Bethe H.A. and W. Heitler (1934), “On the stopping of fast particles and on the creation of positive electrons”, *Proc. R. Soc. (London) A* **146**, 83–112.
- Bhabha H.J. (1936), “The scattering of positrons by electrons with exchange on Dirac’s theory of the positron”, *Proc. R. Soc. (London) A* **154**, 195–206.
- Bielajew A.F. (1988), “Electron transport in $\vec{E}$ and $\vec{B}$ fields”, in *Monte Carlo Transport of Electrons and Photons*, eds. T.M. Jenkins, W.R. Nelson and A. Rindi (Plenum, New York) pp. 421–434.
- Bielajew A.F. (1995), “HOWFAR and HOWNEAR: Geometry Modeling for Monte Carlo Particle Transport”, *Report PIRS-0341* (National Research Council of Canada, Ottawa).
- Bielajew A.F. and D.W.O. Rogers (1987), “PRESTA: The parameter reduced electron-step transport algorithm for electron Monte Carlo transport”, *Nucl. Instrum. Meth. B* **18**, 165–181.
- Bielajew A.F. and D.W.O. Rogers (1988), “Variance-reduction techniques”, in *Monte Carlo Transport of Electrons and Photons*, eds. T.M. Jenkins, W.R. Nelson and A. Rindi (Plenum, New York) pp. 407–419.
- Bielajew A.F. and F. Salvat (2001), “Improved electron transport mechanics in the PENELOPE Monte-Carlo model”, *Nucl. Instrum. Meth. B* **173**, 332–343.
- Biggs F., L.B. Mendelsohn and J.B. Mann (1975), “Hartree-Fock Compton profiles for the elements”, *At. Data Nucl. Data Tables* **16**, 201–309.
- Blunck O. and S. Leisegang (1950), “Zum Energieverlust schneller Elektronen in dünnen Schichten”, *Z. Physik* **128**, 500–505.

- Born M. (1969), *Atomic Physics* (Blackie and Son, London).
- Bransden B.H. and C.J. Joachain (1983), *Physics of Atoms and Molecules* (Longman, Essex, England).
- Brun R., F. Bruyant, M. Maire, A.C. McPherson and P. Zancarini (1986), “GEANT3”, *Report DD/EE/84-1* (CERN, Geneva).
- Brusa D., G. Stutz, J.A. Riveros, J.M. Fernández-Varea and F. Salvat (1996), “Fast sampling algorithm for the simulation of photon Compton scattering”, *Nucl. Instrum. Meth. A* **379**, 167–175.
- Chan H.-P. and K. Doi (1983), “The validity of Monte Carlo simulation in studies of scattered radiation in diagnostic radiology”, *Phys. Med. Biol.* **28**, 109–129.
- Cooper M. (1971), “Compton scattering and electron momentum distributions”, *Adv. Phys.* **20**, 453–491.
- Cullen D.E., M.H. Chen, J.H. Hubbell, S.T. Perkins, E.F. Plechaty, J.A. Rathkopf and J.H. Scofield (1989), “Tables and graphs of photon-interaction cross sections from 10 eV to 100 GeV derived from the LLNL evaluated photon data library (EPDL)”, *Report UCRL-50400* vol. 6, rev. 4, parts A and B. (Lawrence Livermore National Laboratory, Livermore, CA).
- Cullen D.E., J.H. Hubbell and L. Kissel (1997), “EPDL97 The evaluated data library, '97 version”, *Report UCRL-50400* vol. 6, rev. 5 (Lawrence Livermore National Laboratory, Livermore, CA).
- Davies H., H.A. Bethe and L.C. Maximon (1954), “Theory of bremsstrahlung and pair production. II. Integral cross section for pair production”, *Phys. Rev.* **93**, 788–795.
- Desclaux J.P. (1975), “A multiconfiguration relativistic Dirac-Fock program”, *Comput. Phys. Commun.* **9**, 31–45. Erratum: *ibid.* **13** (1977) 71.
- Deslattes R.D., E.G. Kessler, P. Indelicato, L. de Billy, E. Lindroth and J. Anton (2003) “X-ray transition energies: new approach to a comprehensive evaluation”, *Rev. Mod. Phys.* **75**, 36–99.
- Doyle P.A. and P.S. Turner (1968), “Relativistic Hartree-Fock X-ray and electron scattering factors”, *Acta Cryst. A* **24**, 390–397.
- Edmonds A.R. (1960), *Angular Momentum in Quantum Mechanics*, 2nd edition (Princeton University Press, Princeton, NJ).
- Fano U. (1954), “Inelastic collisions and the Molière theory of multiple scattering”, *Phys. Rev.* **93**, 117–120.
- Fano U. (1963), “Penetration of protons, alpha particles and mesons”, *Ann. Rev. Nucl. Sci.* **13**, 1–66.
- Fernández-Varea J.M., R. Mayol, D. Liljequist and F. Salvat (1993a), “Inelastic scattering of electrons in solids from a generalized oscillator strength model using optical and photoelectric data”, *J. Phys: Condens. Matter* **5**, 3593–3610.

- Fernández-Varea J.M., R. Mayol, J. Baró and F. Salvat (1993b), “On the theory and simulation of multiple elastic scattering of electrons”, *Nucl. Instrum. Meth. B* **73**, 447–473.
- Ferrari A., P.R. Sala, A. Fassò and J. Ranft (2005), “FLUKA: a multi-particle transport code (Program version 2005)”, *CERN-2005-10, INFN/TC-05/11, SLAC-R-773* (CERN, Geneva).
- Furness J.B. and I.E. McCarthy (1973), “Semiphenomenological optical model for electron scattering on atoms”, *J. Phys. B: At. Mol. Phys.* **6**, 2280–2291.
- Goudsmit S. and J.L. Saunderson (1940a), “Multiple scattering of electrons”, *Phys. Rev.* **57**, 24–29.
- Goudsmit S. and J.L. Saunderson (1940b), “Multiple scattering of electrons. II”, *Phys. Rev.* **58**, 36–42.
- Halbleib J.A., R.P. Kensek, T.A. Mehlhorn, G.D. Valdez, S.M. Seltzer and M.J. Berger (1992), “ITS version 3.0: the integrated TIGER series of coupled electron/photon Monte Carlo transport codes”, *Report SAND91-1634* (Sandia National Laboratories, Albuquerque, NM).
- Hahn B., D.G. Ravenhall and R. Hofstadter (1956), “High-energy electron scattering”, *Phys. Rev.* **101**, 1131–1142.
- Haug E. (1975), “Bremsstrahlung and pair production in the field of free electrons”, *Z. Naturforsch.* **30a**, 1099–1113.
- Hayward E. and J. Hubbell (1954), “The albedo of various materials for 1-Mev photons”, *Phys. Rev.* **93**, 955–956.
- Heinrich K.F.J. and D.E. Newbury (1991), eds., *Electron Probe Quantitation* (Plenum Press, New York).
- Heitler W. (1954), *The Quantum Theory of Radiation* (Oxford Univ. Press, London).
- Hippler R. (1990), “Plane wave Born calculations of K-shell ionization at low velocities”, *Phys. Lett.* **A144**, 81–85.
- Hirayama H., Y. Namito, A.F. Bielajew, S.J. Wilderman and W.R. Nelson (2005), “The EGS5 Code System”, *SLAC-R-730 and KEK Report 2005-8* (Stanford Linear Accelerator Center, Stanford, CA).
- Hubbell J.H. (1989), “Bibliography and current status of K, L, and higher shells fluorescence yields for computations of photon energy-absorption coefficients”, *Report NISTIR 89-4144* (National Institute of Standards and Technology, Gaithersburg, MD).
- Hubbell J.H., H.A. Gimm and I. Øverbø (1980), “Pair, triplet, and total atomic cross sections (and mass attenuation coefficients) for 1 MeV–100 GeV photons in elements $Z = 1$ to 100”, *J. Phys. Chem. Ref. Data* **9**, 1023–1147.
- Hubbell J.H., Wm.J. Veigele, E.A. Briggs, R.T. Brown, D.T. Cromer and R.J. Howerton (1975), “Atomic form factors, incoherent scattering functions, and photon scattering

- cross sections", *J. Phys. Chem. Ref. Data* **4**, 471–538. Erratum: *ibid.* **6** (1977) 615–616.
- ICRU 37 (1984), *Stopping Powers for Electrons and Positrons* (ICRU, Bethesda, MD).
- Inokuti M. (1971), "Inelastic collisions of fast charged particles with atoms and molecules—the Bethe theory revisited", *Rev. Mod. Phys.* **43**, 297–347.
- Inokuti M., Y. Itikawa and J.E. Turner (1978), "Addenda: inelastic collisions of fast charged particles with atoms and molecules—the Bethe theory revisited", *Rev. Mod. Phys.* **50**, 23–35.
- Inokuti M. and D.Y. Smith (1982), "Fermi density effect on the stopping power of metallic aluminum", *Phys. Rev. B* **25**, 61–66.
- Jablonski A. (1987), "Effects of Auger electron elastic scattering in quantitative AES", *Surf. Science* **188**, 164–180.
- Jablonski A., F. Salvat and C.J. Powell (2003), "NIST Electron Elastic-Scattering Cross-Section Database—Version 3.1" (National Institute of Standards and Technology, Gaithersburg MD). <http://www.nist.gov/srd/nist64.htm>.
- Jackson J.D. (1975), *Classical Electrodynamics* (John Wiley and Sons, New York).
- James F. (1980), "Monte Carlo theory and practice", *Rep. Prog. Phys.* **43**, 1145–1189.
- James F. (1990), "A review of pseudorandom number generators", *Comput. Phys. Commun.* **60**, 329–344.
- Jenkins T.M., W.R. Nelson and A. Rindi (1988), eds., *Monte Carlo Transport of Electrons and Photons* (Plenum, New York).
- Kalos M.H. and P.A. Whitlock (1986), *Monte Carlo Methods*, vol. 1 (Wiley, New York).
- Kane P.P., L. Kissel, R.H. Pratt and S.C. Roy (1986), "Elastic scattering of γ -rays and X-rays by atoms", *Phys. Rep.* **140**, 75–159.
- Kawrakow I. and D.W.O. Rogers (2001), "The EGSnrc code system: Monte Carlo simulation of electron and photon transport", *Report PIRS-701* (National Research Council of Canada, Ottawa).
- Kim L., R.H. Pratt, S.M. Seltzer and M.J. Berger (1986), "Ratio of positron to electron bremsstrahlung energy loss: an approximate scaling law", *Phys. Rev. A* **33**, 3002–3009.
- Kirkpatrick P. and L. Wiedmann (1945), "Theoretical continuous X-ray energy and polarization", *Phys. Rev.* **67**, 321–339.
- Kissel L., C.A. Quarles and R.H. Pratt (1983), "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.
- Kittel C. (1976), *Introduction to Solid State Physics* (John Wiley and Sons, New York).
- Koch H.W. and J.W. Motz (1959), "Bremsstrahlung cross-section formulas and related data", *Rev. Mod. Phys.* **31**, 920–955.

- Landau L. (1944), “On the energy loss of fast particles by ionisation”, *J. Phys. U.S.S.R.* **8**, 201–207.
- L’Ecuyer P. (1988), “Efficient and portable combined random number generators”, *Commun. ACM* **31**, 742–749.
- Lederer C.M. and V.S. Shirley (1978), eds., *Table of Isotopes*, 7th edition (Wiley, New York) appendix III.
- Lewis H.W. (1950), “Multiple scattering in an infinite medium”, *Phys. Rev.* **78**, 526–529.
- Lewis H.W. (1952), “Range straggling of a nonrelativistic charged particle”, *Phys. Rev.* **85**, 20–24.
- Liljequist D. (1983), “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.
- Liljequist D. (1987), “Critical path length for the similarity of elastic multiple scattering processes”, *J. Appl. Phys.* **62**, 333–341.
- Lindhard J. and A. Winther (1964), “Stopping power of electron gas and equipartition rule”, *Mat. Fys. Medd. Dan. Vid. Selsk.* **34**, 1–22.
- Ljungberg M. and S.-E. Strand (1989), “A Monte Carlo program for the simulation of scintillation camera characteristics”, *Comput. Meth. Programs Biomed.* **29**, 257–272.
- Manson S.T. (1972), “Theoretical study of generalized oscillator strengths in atoms: comparison with experiment and other calculations”, *Phys. Rev. A* **5**, 668–677.
- Maron M.J. (1982), *Numerical Analysis: A Practical Approach* (Macmillan, New York).
- Mayol R. and F. Salvat (1990), “Cross sections for K-shell ionisation by electron impact”, *J. Phys. B: At. Mol. Opt. Phys.* **23**, 2117–2130.
- Mohr P.J. and B.N. Taylor (2005), “CODATA recommended values of the fundamental physical constants: 2002”, *Rev. Mod. Phys.* **77**, 1–107.
- Molière G. (1947), “Theorie der Streuung schneller geladener Teilchen I. Einzelstreuung am abgeschirmten Coulomb-Feld”, *Z. Naturforsch.* **2a**, 133–145.
- Molière G. (1948), “Theorie der Streuung schneller geladener Teilchen II. Mehrfach- und Vielfachstreuung”, *Z. Naturforsch.* **3a**, 78–97.
- Møller C. (1932), “Zur Theorie des Durchgangs schneller Elektronen durch Materie”, *Ann. Physik* **14**, 531–585.
- Mott N.F. and H.S.W. Massey (1965), *The Theory of Atomic Collisions*, 3rd edition (Oxford Univ. Press, London).
- Motz J.W., H.A. Olsen and H.W. Koch (1969), “Pair production by photons”, *Rev. Mod. Phys.* **41**, 581–639.

- Namito Y., S. Ban and H. Hirayama (1994), "Implementation of the Doppler broadening of a Compton-scattered photon into the EGS4 code", *Nucl. Instrum. Meth. A* **349**, 489–494.
- Nelson W.R., H. Hirayama and D.W.O. Rogers (1985), "The EGS4 Code System", *Report SLAC-265* (Stanford Linear Accelerator Center, Stanford, CA).
- 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).
- Pratt R.H., A. Ron and H.K. Tseng (1973), "Atomic photoelectric effect above 10 keV", *Rev. Mod. Phys.* **45**, 273–325. Erratum: *ibid.* **45** (1973) 663–664.
- Pratt R.H., H.K. Tseng, C.M. Lee and L. Kissel (1977), "Bremsstrahlung energy spectra from electrons of kinetic energy $1 \text{ keV} \leq T_1 \leq 2000 \text{ keV}$ incident on neutral atoms $2 \leq Z \leq 92$ ", *At. Data Nucl. Data Tables* **20**, 175–209. Erratum: *ibid.* **26** (1981) 477–481.
- Press W.H. and S.A. Teukolsky (1992), "Portable random number generators", *Computers in Physics* **6**, 522–524.
- Reimer L. (1985), *Scanning Electron Microscopy* (Springer, Berlin).
- Reimer L. and E.R. Krefting (1976), "The effect of scattering models on the results of Monte Carlo calculations", *National Bureau of Standards Special Publication 460* (US Government Printing Office, Washington DC) pp. 45–60.
- Reimer L., U. Zepke, J. Moesch, St. Schulze-Hillert, M. Ross-Messemer, W. Probst and E. Weimer (1992), *EEL Spectroscopy* (Carl Zeiss, Oberkochen).
- Ribberfors R. (1983), "X-ray incoherent scattering total cross sections and energy-absorption cross sections by means of simple calculation routines", *Phys. Rev. A* **27**, 3061–3070.
- Ribberfors R. and K.-F. Berggren (1982), "Incoherent-x-ray-scattering functions and cross section $(d\sigma/d\Omega)_{\text{incoh}}$ by means of a pocket calculator", *Phys. Rev. A* **26**, 3325–3333.
- Rubinstein R.Y. (1981), *Simulation and the Monte Carlo Method* (Wiley, New York).
- Sakurai J.J. (1967), *Advanced Quantum Mechanics* (Addison and Wesley, New York).
- Saloman E.B., J.H. Hubbell and J.H. Scofield (1988), "X-ray attenuation cross sections for energies 100 eV to 100 keV and elements $Z = 1$ to $Z = 92$ ", *At. Data Nucl. Data Tables* **38**, 1–197.
- Salvat F. (1987), "Algorithms for random sampling from single-variate distributions", *Comput. Phys. Commun.* **46**, 427–436.
- Salvat F. (1998), "Simulation of electron multiple elastic scattering", *Radiat. Phys. Chem.* **53**, 247–256.
- Salvat F. (2003), "Optical-model potential for electron and positron elastic scattering by atoms", *Phys. Rev. A* **68**, 012708.

- Salvat F. and J.M. Fernández-Varea (1992), “Semiempirical cross sections for the simulation of the energy loss of electrons and positrons in matter”, *Nucl. Instrum. Meth. B* **63**, 255–269.
- 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.
- Sauter F. (1931), “Über den atomaren Photoeffekt in der K-Schale nach der relativistischen Wellenmechanik Diracs”, *Ann. Phys.* **11**, 454–488.
- Schiff L.I. (1968), *Quantum Mechanics*, 3rd edition (McGraw-Hill Kogakusha Ltd., Tokyo).
- Schneider H., I. Tobehn, F. Ebel and R. Hippler (1993), “Absolute cross sections for inner shell ionization by lepton impact”, *Phys. Rev. Lett.* **71**, 2707–2709.
- Schultz P.J. and K.G. Lynn (1988), “Interaction of positron beams with surfaces, thin films, and interfaces”, *Rev. Mod. Phys.* **60**, 701–770.
- Scofield J.H. (1978) “K- and L-shell ionization of atoms by relativistic electrons”, *Phys. Rev. A* **18**, 963–970.
- 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.
- Seltzer S.M. and M.J. Berger (1986), “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.
- Sempau J., E. Acosta, J. Baró, J.M. Fernández-Varea and F. Salvat (1997), “An algorithm for Monte Carlo simulation of coupled electron-photon transport”, *Nucl. Instrum. Meth. B* **132**, 377–390.
- Sempau J., J.M. Fernández-Varea, E. Acosta and F. Salvat (2003), “Experimental benchmarks of the Monte Carlo code PENELOPE”, *Nucl. Instrum. Meth. B* **207**, 107–123.
- Sevier K.D. (1972), *Low Energy Electron Spectrometry* (Wiley Interscience, New York).
- Shiles E., T. Sasaki, M. Inokuti and D.Y. Smith (1980), “Self-consistency and sum-rule tests in the Kramers-Kronig analysis of optical data: applications to aluminum”, *Phys. Rev. B* **22**, 1612–1628.
- Snyder W.S., M.R. Ford, G.G. Warner and H.L. Fisher Jr. (1969), “Estimates of absorbed fractions for monoenergetic photon sources uniformly distributed in various organs of a heterogeneous phantom”, MIRD Pamphlet No. 5, *J. Nucl. Med.* **10**, suppl. No. 3, 5–52.
- Statham P.J. (1976), “The generation, absorption and anisotropy of thick-target bremsstrahlung and the implications for quantitative energy dispersive analysis”, *X-Ray Spectrom.* **5**, 154–168.

- Sternheimer R.M. (1952), “The density effect for the ionization loss in various materials”, *Phys. Rev.* **88**, 851–859.
- Sternheimer R.M., S.M. Seltzer and M.J. Berger (1982), “Density effect for the ionization loss of charged particles in various substances”, *Phys. Rev. B* **26**, 6067–6076. Erratum: *ibid.* **27** (1983) 6971.
- Titus F. (1970), “Measurements of the energy response functions of scintillators for monoenergetic electrons”, *Nucl. Instrum. Meth.* **89**, 93–100.
- Tofterup A.L. (1986), “Theory of elastic and inelastic scattering of electrons emitted from solids: energy spectra and depth profiling in XPS/AES”, *Surf. Science* **167**, 70–100.
- Tsai Y.S. (1974), “Pair production and bremsstrahlung of charged leptons”, *Rev. Mod. Phys.* **46**, 815–851.
- Walker A.J. (1977), “An efficient method for generating discrete random variables with general distributions”, *ACM Trans. Math. Software* **3**, 253–256.
- Walker D.W. (1968), “Spin polarization in electron scattering from molecules”, *Phys. Rev. Lett.* **20**, 827–828.
- Walker D.W. (1971), “Relativistic effects in low energy electron scattering from atoms”, *Adv. Phys.* **20**, 257–323.
- Wentzel G. (1927), “Zwei Bemerkungen über die Zerstreuung korpuskularer Strahlen als Beugungserscheinung”, *Z. Phys.* **40**, 590–593.
- X-5 Monte Carlo Team (2003), “MCNP—A general Monte Carlo N-particle transport code, version 5”, *Report LA-UR-03-1987* (Los Alamos National Laboratory, Los Alamos, NM).
- Yates A.C. (1968), “Calculations of electron spin polarization for electron-molecule collisions”, *Phys. Rev. Lett.* **20**, 829–831.
- Zerby C.D. (1963), “A Monte Carlo calculation of the response of gamma-ray scintillation counters”, in *Methods in Computational Physics*, vol. 1, eds. B. Alder, S. Fernbach and M. Rotenberg (Academic Press, New York) pp. 89–134.
- Zheng-Ming L. and A. Brahme (1993), “An overview of the transport theory of charged particles”, *Radiat. Phys. Chem.* **41**, 673–703.
-