Data Bank
NEA/MBDAV/R(2019)1
September 2019
www.oecd-nea.org

PENELOPE-2018: A Code System for Monte Carlo Simulation of Electron and Photon Transport

Workshop Proceedings
Barcelona, Spain
28 January – 1 February 2019

OECD
BETTER POLICIES FOR BETTER LIVES

NEA
NUCLEAR ENERGY AGENCY

NEA/MBDAV/R(2019)1

Data Bank

PENELOPE-2018

A Code System for Monte Carlo Simulation of Electron and Photon Transport

Workshop Proceedings
Barcelona, Spain
28 January – 1 February 2019

©OECD 2019

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 36 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, Chile, the Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Israel, Italy, Japan, Korea, Latvia, Lithuania, Luxembourg, Mexico, the Netherlands, New Zealand, Norway, Poland, Portugal, the Slovak Republic, Slovenia, Spain, Sweden, Switzerland, Turkey, the United Kingdom and the United States. The European Commission 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.

NUCLEAR ENERGY AGENCY

The OECD Nuclear Energy Agency (NEA) was established on 1 February 1958. Current NEA membership consists of 33 countries: Argentina, Australia, Austria, Belgium, Canada, the Czech Republic, Denmark, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Japan, Korea, Luxembourg, Mexico, the Netherlands, Norway, Poland, Portugal, Romania, Russia, the Slovak Republic, Slovenia, Spain, Sweden, Switzerland, Turkey, the United Kingdom and the United States. The European Commission and the International Atomic Energy Agency also take 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 sound and economical use of nuclear energy for peaceful purposes;
- 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 analyses in areas such as energy and the sustainable development of low-carbon economies.

Specific areas of competence of the NEA include the safety and regulation of nuclear activities, radioactive waste management and decommissioning, 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.

This document, as well as any data and map included herein, are without prejudice to the status of or sovereignty over any territory, to the delimitation of international frontiers and boundaries and to the name of any territory, city or area.

Corrigenda to OECD publications may be found online at: *www.oecd.org/publishing/corrigenda*.

© OECD 2019

You can copy, download or print OECD content for your own use, and you can include excerpts from OECD publications, databases and multimedia products in your own documents, presentations, blogs, websites and teaching materials, provided that suitable acknowledgement of the OECD as source and copyright owner is given. All requests for public or commercial use and translation rights should be submitted to *neapub@oecd-nea.org*. Requests for permission to photocopy portions of this material for public or commercial use shall be addressed directly to the Copyright Clearance Center (CCC) at *info@copyright.com* or the Centre français d'exploitation du droit de copie (CFC) *contact@cfcopies.com*.

Foreword

The Nuclear Energy Agency (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 of CPS collects, complies and verifies programs in an appropriate computer environment. This ensures that the computer program package is complete and adequately documented before it is added to the library and distributed to users on request.

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 is radiation transport. Workshops and training courses, including the use of computer codes, have been organised in the field of neutral particle transport, both on codes using deterministic methods and on programs relying on stochastic approaches. The area of charged particle transport, and in particular electron-photon transport, has received increased attention for a number of technological and medical applications.

In 2001, the NEA Data Bank released the computer code "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 widely used by radiation physicists and since then, PENELOPE workshops with hands-on training were organised on a yearly basis. The NEA Nuclear Science Committee endorsed this activity 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. Regular updates of the code were developed, improving the physics model and algorithm used in PENELOPE.

These proceedings contain the manual and teaching notes of the PENELOPE-2018 workshop held on 28 January – 1 February 2019 in Barcelona, Spain.

NEA/MBDAV/R(2019)1

PENELOPE, a code system for Monte Carlo simulation of electron and photon transport

Francesc Salvat

Facultat de Física (FQA and ICC), Universitat de Barcelona.
Diagonal 645, E-08028 Barcelona, Spain.

Abstract

The computer code system PENELOPE (version 2018) 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 PENGEOM permits the generation of random electron-photon showers in material systems consisting of homogeneous bodies limited by quadric surfaces, *i.e.*, planes, spheres, cylinders, cones, *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. Constructive quadric geometry.

Symbols and numerical values of constants frequently used in the text (Mohr *et al.*, 2010).

Quantity	Symbol	Value
Avogadro's number	N_{A}	$6.022\,141\ \ 29{\times}10^{23}$ mol^{-1}
Velocity of light in vacuum	c	$2.997\,924\,58{\times}10^{8}$ m s^{-1}
Reduced Planck's constant	$\hbar = h/(2\pi)$	$6.582\,119\,28{\times}10^{-16}$ eV s
Electron charge	e	$1.602\,176\,565{\times}10^{-19}$ C
Electron mass	$\mathrm{m_e}$	$9.109\,382\,91{\times}10^{-31}$ kg
Electron rest energy	$\mathrm{m_e}c^2$	510.998 928 keV
Classical electron radius	$r_{\mathrm{e}} = e^2/(\mathrm{m_e}c^2)$	$2.817\,940\,326\,7{\times}10^{-15}$ m
Fine-structure constant	$\alpha = e^2/(\hbar c)$	1/137.035 999 074
Bohr radius	$a_0 = \hbar^2/(\mathrm{m_e}e^2)$	$0.529\,177\,210\,92{\times}10^{-10}$ m
Hartree energy	$E_{\mathrm{h}} = e^2/a_0$	27.211 385 05 eV

Revision date: 3 September, 2019

Table of contents

Preface **vii**

List of acronyms **xiii**

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 10

1.2.3 Discrete distributions 11

1.2.3.1 Walker's aliasing method 12

1.2.4 Numerical inverse transform for continuous PDFs 14

1.2.4.1 Determining the interpolation grid 16

1.2.4.2 Sampling algorithm 17

1.2.5 Rejection methods 19

1.2.6 Two-dimensional variables: Composition methods 21

1.2.6.1 Examples 22

1.3 Monte Carlo integration 24

1.3.1 Monte Carlo *vs.* numerical quadrature 27

1.4 Simulation of radiation transport 30

1.4.1 Interaction cross sections 30

1.4.2 Mean free path 32

1.4.3 Scattering model and probability distributions 33

1.4.4 Generation of random tracks . . . 35
1.4.5 Particle transport as a Markov process . . . 37
1.5 Statistical averages and uncertainties . . . 39
1.6 Variance reduction . . . 43
1.6.1 Interaction forcing . . . 44
1.6.2 Splitting and Russian roulette . . . 45
1.6.3 Delta scattering of photons . . . 46
1.6.4 Other methods . . . 46

2 Photon interactions 49
2.1 Photoelectric effect . . . 50
2.1.1 Simulation of photoelectron emission . . . 53
2.1.1.1 Initial direction of photoelectrons . . . 54
2.2 Coherent (Rayleigh) scattering . . . 56
2.2.1 Simulation of coherent scattering events . . . 58
2.3 Incoherent (Compton) scattering . . . 60
2.3.1 Analytical Compton profiles . . . 67
2.3.2 Simulation of incoherent scattering events . . . 69
2.4 Electron-positron pair production . . . 73
2.4.1 Simulation of pair-production events . . . 78
2.4.1.1 Angular distribution of the produced particles . . . 80
2.4.1.2 Compound materials . . . 80
2.5 Attenuation coefficients . . . 81
2.6 Atomic relaxation . . . 84
2.6.1 Photon propagation time and atomic lifetimes . . . 88
2.7 Scattering of polarised photons . . . 90
2.7.1 Rayleigh scattering . . . 94
2.7.2 Compton scattering . . . 96
2.8 Simulation of interactions of polarised photons . . . 99
2.8.1 Rayleigh and Compton scattering . . . 99
2.8.2 Angular distribution of photoelectrons . . . 102

3 Electron and positron interactions 105

3.1 Elastic collisions . . . 106
3.1.1 Partial-wave cross sections . . . 110
3.1.1.1 Simulation of single-scattering events . . . 114
3.1.2 The modified Wentzel (MW) model . . . 116
3.1.2.1 Simulation of single elastic events with the MW model . 120
3.2 Inelastic collisions . . . 121
3.2.1 GOS model . . . 125
3.2.2 Differential cross sections . . . 129
3.2.2.1 Distant interactions with inner-shell electrons . . . 131
3.2.2.2 DCS for close collisions of electrons . . . 133
3.2.2.3 DCS for close collisions of positrons . . . 133
3.2.3 Integrated cross sections . . . 134
3.2.4 Stopping power of high-energy electrons and positrons . . . 139
3.2.5 Simulation of hard inelastic collisions . . . 141
3.2.5.1 Hard distant interactions . . . 142
3.2.5.2 Hard close collisions of electrons . . . 143
3.2.5.3 Hard close collisions of positrons . . . 145
3.2.5.4 Secondary electron emission . . . 145
3.2.6 Ionisation of inner shells . . . 146
3.2.6.1 Simulation of ionizing collisions . . . 149
3.3 Bremsstrahlung emission . . . 150
3.3.1 The energy-loss scaled DCS . . . 151
3.3.2 Integrated cross sections . . . 154
3.3.2.1 CSDA range and radiative yield . . . 156
3.3.3 Angular distribution of emitted photons . . . 160
3.3.3.1 Analytical shape functions . . . 161
3.3.4 Simulation of hard radiative events . . . 164
3.3.4.1 Sampling of the photon energy . . . 165
3.3.4.2 Angular distribution of emitted photons . . . 166
3.4 Positron annihilation . . . 167
3.4.1 Generation of emitted photons . . . 169

4 Electron/positron transport mechanics 171
4.1 Elastic scattering . . . 172
4.1.1 Multiple elastic scattering theory . . . 172
4.1.2 Mixed simulation of elastic scattering . . . 173
4.1.2.1 Angular deflections in soft scattering events . . . 177
4.1.3 Simulation of soft events . . . 178
4.2 Soft energy losses . . . 181
4.2.1 Energy dependence of the soft DCS . . . 185
4.3 Combined scattering and energy loss . . . 187
4.3.1 Variation of $\lambda_{\mathrm{T}}^{(\mathrm{h})}$ with energy . . . 189
4.3.2 Scattering by atomic electrons . . . 193
4.3.3 Energy deposition and flight times . . . 195
4.4 Generation of random tracks . . . 197
4.4.1 Stability of the simulation algorithm . . . 200

5 Radiation fields and dosimetry 203
5.1 Radiation fields . . . 204
5.1.1 Current density and flux density . . . 205
5.1.2 Radiometric quantities . . . 207
5.2 Monte Carlo simulation and dosimetry . . . 212
5.2.1 Monte Carlo calculation of the linear energy absorption . . . 215
5.2.1.1 LEA of photons . . . 215
5.2.1.2 LEA of electrons and positrons . . . 216
5.2.2 Absorbed dose . . . 218
5.2.3 Track length estimator of the absorbed dose . . . 221
5.3 Dosimetry of photon fields . . . 222
5.4 Dosimetry of charged-particle fields . . . 227

6 Constructive quadric geometry 231
6.1 Surfaces and their transformations . . . 233
6.2 Constructive quadric geometry and ray tracing . . . 236
6.2.1 Quadric surfaces . . . 238
6.2.2 Fuzzy quadric surfaces . . . 242

6.2.3 Modules and genealogical tree . . . 246
6.3 Geometry-definition file . . . 250
6.4 The subroutine package PENGEOM . . . 259
6.5 Impact detectors . . . 265
6.6 Debugging and viewing the geometry . . . 267
6.7 A short tutorial . . . 270

7 Structure and operation of the code system 275
7.1 PENELOPE . . . 276
7.1.1 Database and input material data file . . . 277
7.1.2 Structure of the main program . . . 286
7.1.3 Layout of a generic main program . . . 299
7.1.4 Variance reduction . . . 302
7.2 Examples of main programs . . . 305
7.2.1 Program `pencyl` . . . 307
7.2.1.1 Structure of the input file . . . 309
7.2.1.2 Example . . . 317
7.2.2 Program `penmain` . . . 319
7.2.2.1 Structure of the input file . . . 321
7.3 Selecting the simulation parameters . . . 334
7.4 The code SHOWER . . . 339
7.5 Installation . . . 340

A Collision kinematics 345
A.1 Two-body reactions . . . 346
A.1.1 Elastic scattering . . . 348
A.2 Inelastic collisions of charged particles . . . 349

B Numerical tools 353
B.1 Cubic spline interpolation . . . 353
B.2 Numerical quadrature . . . 357
B.2.1 Gauss integration . . . 357
B.2.2 Adaptive bipartition . . . 358

C Photon polarisation **359**
C.1 Polarisation states . . . 359
C.2 Density matrix and Stokes parameters . . . 362
C.2.1 Rotations of the base vectors . . . 366
C.3 Polarisation analysers . . . 368
C.3.1 Real analysers . . . 370
C.4 Simulation and polarisation states . . . 371

D Electron/positron transport in electromagnetic fields **375**
D.1 Tracking particles in vacuum . . . 376
D.1.1 Uniform electric fields . . . 379
D.1.2 Uniform magnetic fields . . . 379
D.2 Tracking particles in matter with an EM field . . . 381

Bibliography **385**

Preface

Radiation transport in matter has been a subject of interest since the beginning of the 20th century. 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 produced[1]. 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 (Eds.), 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; Chetty *et al.*, 2007).

The study of radiation transport problems was initially attempted on the basis of the linear 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 (*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. The evolution of an electron-photon shower is of a random nature and so it 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 linear transport equation (apart from the inherent statistical uncertainties).

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

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, a photon history terminates after a single photoelectric or pair-production interaction or after a few Compton interactions (say, of the order of 10). With present-day computational resources, 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 (interaction by interaction) is feasible only when the average number of collisions per track is not too large (say, up to a few hundred). Experimental 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, 1988a,b,c), ITS3 (Halbleib *et al.*, 1992), EGS4 (Nelson *et al.*, 1985), GEANT3 (Brun *et al.*, 1987), 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.*, 2006)] 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 mean free path for elastic collisions (see Fernández-Varea *et al.*, 1993a) 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.

The present report describes the 2018 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 fairly 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 PENELOPE 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 charged particles may take longer times).

PENELOPE in its current form is the result of continued evolution from the first version, which was released in 1996. The present version incorporates various changes and additions, which aim to improve both reliability and flexibility of the code system. The database of photoabsorption cross sections has been recalculated with the normalization screening correction included. Because that correction is controversial, the distribution package contains an alternative database that was calculated without the normalization screening correction. The angular distributions of bremmstrahlung photons are determined from a larger set of shape functions, which were calculated with the partial-wave program BREMS of Poškus (2018). The angular distribution of photoelectrons now includes the effect of photon polarisation, and the sampling of atomic relaxation accounts for the effect of having multiple vacancies in the intermediate and final subshells. The example main program `penmain` has been extended to include the option of radioactive sources with a single radionuclide, which are simulated using evaluated decay data from the NUCLEIDE Table of Isotopes (`www.nucleide.org/DDEP_WG/DDEPdata.htm`). `penmain` also allows producing output files with the particle tracks of a number of showers, which can be used for visualisation. Minor bugs and inconsistencies in the code

system have also been corrected.

The present document 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. The first chapters of the report deal with the fundamentals of Monte Carlo simulation and physical interaction models. 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 associated sampling techniques, are presented in Chapters 2 and 3[2]. Chapter 4 is devoted to mixed simulation methods for electron and positron transport. Additional information on physical interaction models for electron and photon transport used in PENELOPE and other Monte Carlo codes, with references to relevant publications and data resources, can be found in the review article by Salvat and Fernández-Varea (2009). In Chapter 5 we introduce concepts and quantities of common use in radiation dosimetry, and we describe the Monte Carlo calculation of intermediate dosimetric quantities (linear energy absorption for electrons, photons and positrons). The latter are useful, *e.g.*, to implement track length estimators for the absorbed dose. A relatively simple, but very effective, method to handle simulation in quadric geometries is presented in Chapter 6. The Fortran simulation package PENELOPE, two example main programs, and other complementary tools are described in Chapter 7, which also provides instructions for their operation. Information on relativistic kinematics and numerical methods is given in Appendices A and B, respectively. Appendix C contains a detailed study of photon polarisation, described in terms of the density matrix and the Stokes parameters. Finally, Appendix D is devoted to simulation of electron/positron transport under external, static electric and magnetic fields.

The PENELOPE package includes the Fortran source files, the database, various complementary tools, and code documentation. It is distributed by the NEA Data Bank[3] and the RSICC[4]. The code is also available from the author, but I would appreciate it if users did try to get the code from these institutions.

Acknowledgements

A great deal of the physics of PENELOPE, the program structure, and various parts of the present manual were developed over decades of close and friendly collaboration with José M. Fernández-Varea and Josep Sempau. Without their contributions and inspiration, PENELOPE would be a far less reliable and complete tool. With the maturity of the code, collaboration has become less intense, but they keep providing useful ideas and advice on code development, as well as correcting my errors.

In the course of the Monte Carlo research, I have had the good fortune of obtaining

[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; 46, Quai Alphonse Le Gallo; 92100 Boulogne-Billancourt; France. e-mail: programs@oecd-nea.org; `www.oecd-nea.org`

[4]Radiation Safety Information Computational Center. PO Box 2008, Oak Ridge, TN 37831-6362, USA. e-mail: pdc@ornl.gov; `www-rsicc.ornl.gov`

much help from numerous colleagues and friends. Collaboration with David Liljequist gave shape to our first algorithm for simulation of electrons and positrons. Stimulating collaboration with Alex F. Bielajew led to substantial improvements in electron transport mechanics and in code organisation. I am deeply indebted to John H. Hubbell and Dermott E. Cullen for kindly providing us with information on photon interaction and atomic relaxation data. Thanks are also due to Stephen M. Seltzer for sending us his bremsstrahlung energy-loss database. I am especially indebted to Pedro 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 Antonio Lallena. François Tola and Michel Moreau went deep into programming details, found bugs and suggested many practical improvements.

Alberto Sánchez-Reyes and Eduardo 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. Doctoral students and postdocs helped to improve parts of the code and in applications: Josep Baró, Eduardo Acosta, Loic Sorbier, Cristiani S. Campos, Lorenzo Brualla, Andreu Badal, Gloria González-Muñoz, and Julio Almansa. Alberto Riveros guided us into the field of electron-probe microanalysis and x-ray simulation, where Xavier Llovet has found interesting applications of the code. Needs in this field led to the calculation of reliable databases for electron- and positron-impact ionisation by David Bote.

Finally, I would like to thank the staff of the NEA Data Bank, particularly Juan Manuel Galán, Cristina Lebunetelle, Catherine Rocher-Thromas, and Alice Dufresne for their help in distributing the code and for kindly organising the training courses on PENELOPE.

Continued financial support from the Spanish Ministerio de Economía y Competitividad and ERDF (projects nos. FPA2013-44549-P and FPA2016-77689-C2-2R) and from the Generalitat de Catalunya (grant 2014 SGR 846) is gratefully acknowledged.

Francesc Salvat
Barcelona, February 2019.

List of acronyms

The following acronyms and abbreviations occur repeatedly in the text.

ASCII: American Standard Code for Information Interchange.
CSDA: continuous slowing-down approximation.
DCS: differential cross section.
DWBA: distorted-wave Born approximation.
EADL : Evaluated Atomic Data Library (Perkins *et al.*, 1991).
EM: electromagnetic field or force.
EPDL: Evaluated Photon Data Library (Cullen *et al.*, 1997).
GOS: generalized oscillator strength.
ICRU: International Commission on Radiation Units and Measurements.
LEA: linear energy absorption.
LET: linear energy transfer.
MEA: mass energy absorption.
MW: modified Wentzel model.
NIST: National Institute of Standards and Technology, US Department of Commerce.
OOS: optical oscillator strength.
PDF: probability distribution function.
PWBA: plane-wave Born approximation.
RITA: rational inverse transform with aliasing.

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, 2008), 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 (2008) 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. Sections 1.5 and 1.6 are devoted, respectively, to the evaluation of statistical uncertainties and to 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 $\mathrm{P}\{x|a < x < b\}$, 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 $N \to \infty$. The probability of obtaining x in a differential interval of length $\mathrm{d}x$ about x_1 can be expressed as

$$\mathrm{P}\{x|x_1 < x < x_1 + \mathrm{d}x\} = p(x_1)\,\mathrm{d}x, \tag{1.1}$$

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 the interval $(x_{\min},x_{\max})$, the PDF must be

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

$$p(x) \geq 0 \qquad \text{and} \qquad \int_{x_{\min}}^{x_{\max}} p(x)\,\mathrm{d}x = 1. \tag{1.2}$$

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

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

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

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

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

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

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 $x = x_1, x_2, \ldots$ with point probabilities $p_1, p_2, \ldots$ can be expressed as a mixture of delta distributions,

$$p(x) = \sum_i p_i\,\delta(x - x_i). \tag{1.5}$$

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

$$\mathcal{P}(x) \equiv \int_{x_{\min}}^{x} p(x')\,\mathrm{d}x'. \tag{1.6}$$

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 $\mathrm{P}\{x|a < x < b\}$ of having x in the interval (a,b) is

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

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

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

$$\langle x^n \rangle \equiv \int_{x_{\min}}^{x_{\max}} x^n \, p(x) \, \mathrm{d}x. \tag{1.8}$$

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,

$$p_{\mathrm{L}}(x) \equiv \frac{1}{\pi} \frac{\gamma}{\gamma^2 + x^2}, \qquad -\infty < x < \infty. \tag{1.9}$$

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.*,

$$\langle x \rangle_{\mathrm{L}} = \lim_{a \to \infty} \int_{-a}^{+a} x \, \frac{1}{\pi} \frac{\gamma}{\gamma^2 + x^2} \, \mathrm{d}x = 0. \tag{1.10}$$

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,

$$\langle x \rangle = \int x \, p(x) \, \mathrm{d}x. \tag{1.11}$$

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

$$\langle f(x) \rangle \equiv \int f(x) \, p(x) \, \mathrm{d}x. \tag{1.12}$$

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 $\mathrm{d}f$ is equal to the probability of having x in the corresponding interval or intervals[1]. 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), $p(x) \, \mathrm{d}x = \pi(f) \, \mathrm{d}f$ and

$$\pi(f) = p(x) \left(\mathrm{d}f / \mathrm{d}x \right)^{-1}. \tag{1.13}$$

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

$$\langle f \rangle = \int f \, \pi(f) \, \mathrm{d}f = \int f(x) \, p(x) \, \mathrm{d}x,$$

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

$$\langle a_1 f_1(x) + a_2 f_2(x) \rangle = a_1 \langle f_1(x) \rangle + a_2 \langle f_2(x) \rangle, \tag{1.14}$$

[1] When $f(x)$ 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

$$\mathrm{var}(x) \equiv \langle (x - \langle x \rangle)^2 \rangle = \int (x - \langle x \rangle)^2 \, p(x) \, \mathrm{d}x = \langle x^2 \rangle - \langle x \rangle^2. \tag{1.15}$$

The square root of the variance, $\sigma \equiv [\mathrm{var}(x)]^{1/2}$, 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

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

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

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

$$p(x, y) \geq 0 \qquad \text{and} \qquad \int \mathrm{d}x \int \mathrm{d}y \, p(x, y) = 1. \tag{1.17}$$

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

$$q(x) \equiv \int p(x, y) \, \mathrm{d}y \qquad \text{and} \qquad q(y) \equiv \int p(x, y) \, \mathrm{d}x, \tag{1.18}$$

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

$$p(x, y) = q(x) \, p(y|x) = q(y) \, p(x|y), \tag{1.19}$$

where

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

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

$$\langle f(x, y) \rangle = \int \mathrm{d}x \int \mathrm{d}y \, f(x, y) \, p(x, y). \tag{1.21}$$

The moments of the PDF are defined by

$$\langle x^n y^m \rangle = \int \mathrm{d}x \int \mathrm{d}y \, x^n y^m \, p(x, y). \tag{1.22}$$

In particular,

$$\langle x^n \rangle = \int \mathrm{d}x \int \mathrm{d}y \, x^n \, p(x,y) = \int x^n q(x) \, \mathrm{d}x. \tag{1.23}$$

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

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

The variance of $x + y$ is

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

where

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

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

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

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

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

Moreover, for independent variables,

$$\mathrm{var}\{a_1 x + a_2 y\} = a_1^2 \,\mathrm{var}(x) + a_2^2 \,\mathrm{var}(y). \tag{1.29}$$

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 (2008).

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

Table 1.1: Fortran random-number generator.

```
C  ***************************************************************
C                        FUNCTION RAND
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
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER*4 (I-N)
      PARAMETER (USCALE=1.0D0/2.147483563D9)
      COMMON/RSEED/ISEED1,ISEED2
C
      I1=ISEED1/53668
      ISEED1=40014*(ISEED1-I1*53668)-I1*12211
      IF(ISEED1.LT.0) ISEED1=ISEED1+2147483563
C
      I2=ISEED2/52774
      ISEED2=40692*(ISEED2-I2*52774)-I2*3791
      IF(ISEED2.LT.0) ISEED2=ISEED2+2147483399
C
      IZ=ISEED1-ISEED2
      IF(IZ.LT.1) IZ=IZ+2147483562
      RAND=IZ*USCALE
C
      RETURN
      END
```

on the computer (see Kalos and Whitlock, 2008; James, 1990; Hellekalek, 1998). Among the "good" random-number generators currently available, the simplest ones are the so-called multiplicative congruential generators (Press and Teukolski, 1992). A popular example of this kind of generator is the following,

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

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 Teukolski, 1992). However, the sequence is periodic, with a period of the order of 10^9. With present-day computers, 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.

In our simulations we use the generator RANECU (L'Ecuyer, 1988), which was coded in Fortran by James (1990). In our programs we use the function `RAND` (Table 1.1), which differs from the one given by James in that it generates and delivers a single random number at each call. `RAND` 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. The internal state of the generator is completely characterised by the two integers `ISEED1` and `ISEED2`, the seeds, whose initial values determine the sequence of random numbers utilised during a simulation. Our reasons for adopting the `RAND` generator, aside from its good random properties, are the smallness of the state definition, just two integers, and the repeatability of the sequence of generated random numbers.

In a typical simulation, `RAND` is initialised with seed values entered by the user. It is convenient to structure the simulation program in such a way that 1) each history (usually, a sequence of multiple random events) is completely determined by the values of the seeds at the beginning of the history, and 2) at the end of a simulation run, the program writes the current values of the seeds (*i.e.*, the ones that would start a new history) in an output file. This simple strategy allows a simulation that ended with insufficient statistics to be resumed. It is also useful for identifying program errors that may occur after large simulation times, a situation where interactive debugging tools are of little help. A practical solution consists of modifying the source code to write the `RAND` seeds at regular intervals, and running the program again until the error occurs. The last pair of seeds in the output list can then be entered to start the program close to the first occurrence of the error, which may then be investigated using available debugging tools.

In parallel calculations we have to make sure that different processors produce truly independent (*i.e.*, uncorrelated) sequences of random numbers. This can be accomplished by feeding each processor with initial seeds that belong to a single long sequence generated by `RAND`, but are far enough from each other to avoid overlap of the subsequences generated by the different processors. A convenient property of the `RAND` generator is that the values of the seeds after a number N of calls can be obtained directly from the initial seeds, without calculating the intermediate stages (L'Ecuyer, 1988). Badal and Sempau (2006) have written a Fortran program that, starting from initial seeds specified by the user, calculates the seed values after an arbitrary number of calls. Our random sampling routines include a pre-computed list of seed pairs that belong to a long sequence (initiated with the pair 1,1) and whose relative separation is 10^{14} calls. That is, if we start with the n-th seed pair, after 10^{14} calls to `RAND` we obtain the $(n+1)$-th pair. The n-th pair of seeds is loaded by calling subroutine `RAND0` with its argument equal to n.

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 $p_\xi(\xi)\,\mathrm{d}\xi = p(x)\,\mathrm{d}x$. Hence,

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

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

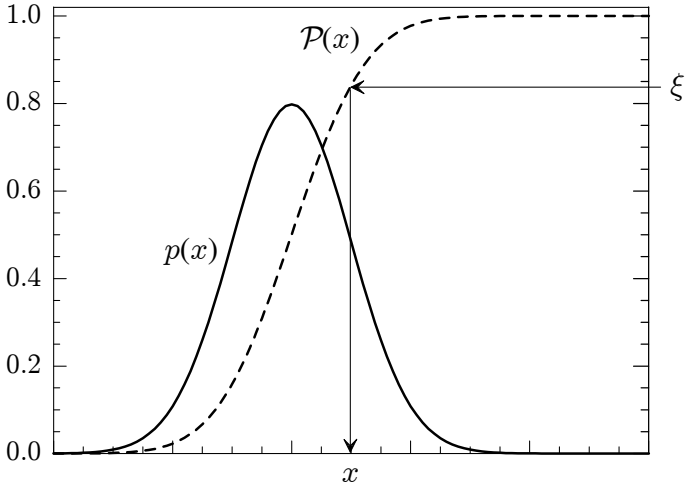

Figure 1.1: Random sampling from a distribution $p(x)$ using the inverse-transform method.

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

$$\xi = \int_{x_{\min}}^{x} p(x')\,\mathrm{d}x', \tag{1.32}$$

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 (ξ_i,x_i) table, 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

$$\xi = \frac{x-a}{b-a}, \tag{1.33}$$

which leads to the well-known sampling formula

$$x = a + \xi(b-a). \tag{1.34}$$

• **Exponential distribution.** The exponential distribution,

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

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

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

The last equality ($\stackrel{\sim}{=}$) 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

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

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

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

and yields the following sampling formula,

$$x = \frac{A\xi}{A+1-\xi}\,. \tag{1.39}$$

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 $x = 1, \ldots, N$ with point probabilities $p_1, \ldots, p_N$, respectively. The corresponding PDF can be expressed as

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

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

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

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

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

We can define the quantities

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

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

$$\mathcal{P}_i < \xi \le \mathcal{P}_{i+1}. \tag{1.44}$$

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 $j = N + 1$.

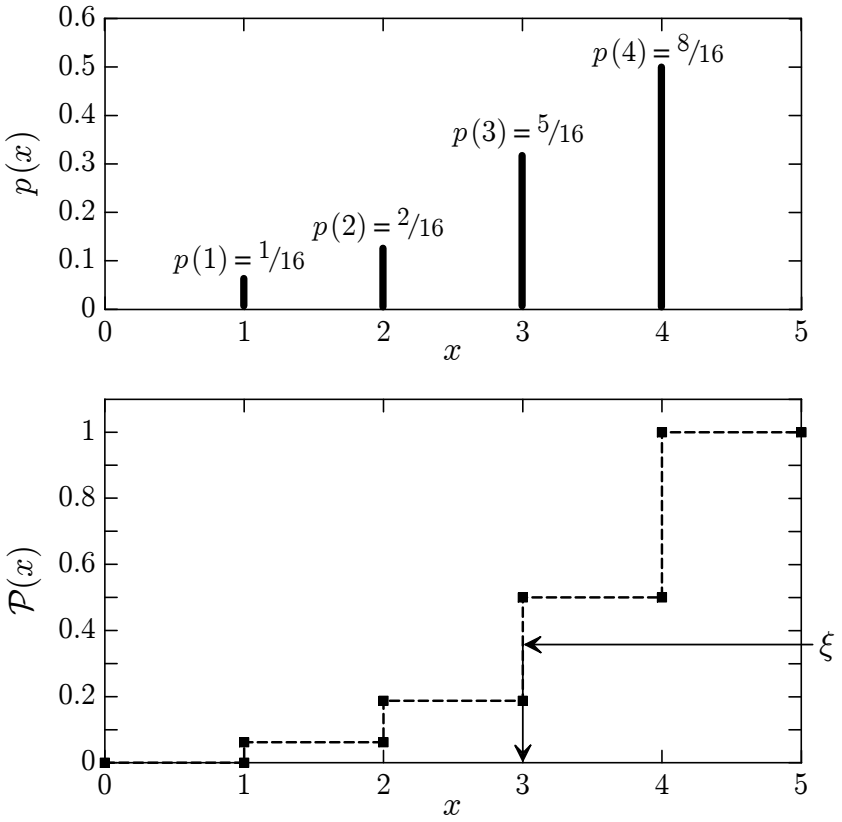

Figure 1.2: Random sampling from a discrete PDF using the inverse-transform method. The random variable can take the values $i = 1$, 2, 3 and 4 with relative probabilities 1, 2, 5 and 8, respectively.

(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

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

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 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 $\mathcal{P}_i$).

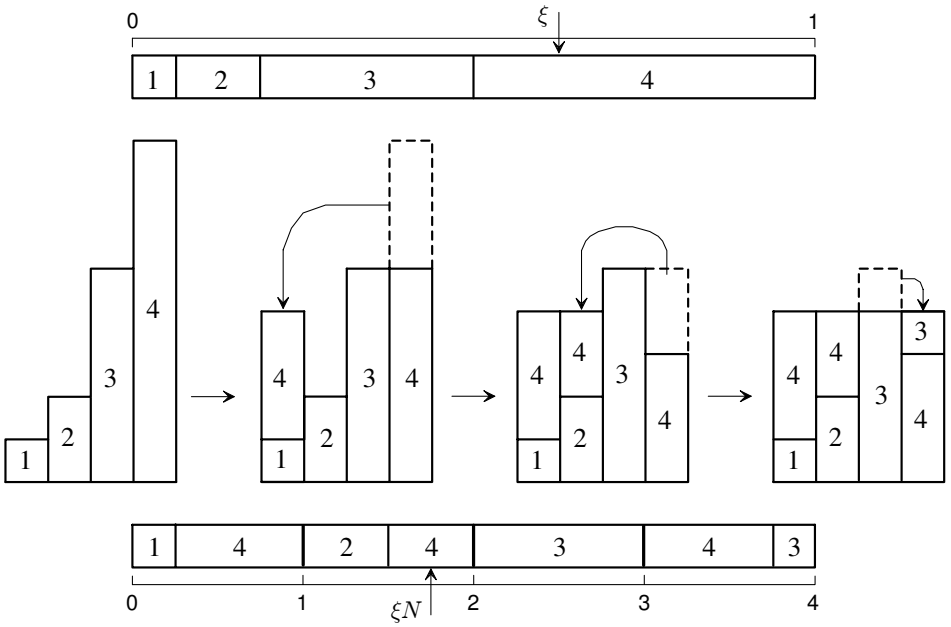

Figure 1.3: Graphical representation of the inverse-transform method (top) and Walker's aliasing method (bottom) for random sampling from the discrete distribution shown in Fig. 1.2. (Adapted from Salvat, 1987).

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 in Appendix B), 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 $(x_{\min}, x_{\max})$,

$$x_1 = x_{\min} < x_2 < \ldots < x_{N-1} < x_N = x_{\max}. \tag{1.46}$$

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 $(0, 1)$,

$$\xi_1 = 0 < \xi_2 < \ldots < \xi_{N-1} < \xi_N = 1. \tag{1.47}$$

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 $\mathcal{P}^{-1}(\xi)$,

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

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 type[2]

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

where

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

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

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

The parameters a_i and b_i are determined by requiring that

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

This implies

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

and it follows that

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

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

[2]We denote by $\widetilde{\mathcal{P}}^{-1}(\xi)$ 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

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

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

$$\widetilde{p}(x) = \left(\frac{\mathrm{d}\widetilde{\mathcal{P}}^{-1}(\xi)}{\mathrm{d}\xi}\right)^{-1} = \frac{(1 + a_i\eta + b_i\eta^2)^2}{(1 + a_i + b_i)(1 - b_i\eta^2)}\,\frac{\xi_{i+1} - \xi_i}{x_{i+1} - x_i} \qquad \text{if } x_i \le x < x_{i+1}. \tag{1.55}$$

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.*, $\widetilde{p}(x)$ is positive, as required for a proper PDF. To calculate $\widetilde{p}(x)$ 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

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

It is worth noting that the distributions $p(x)$ and $\widetilde{p}(x)$ 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.*, $\widetilde{p}(x)$ 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. Furthermore, 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 ($i = 1, \ldots, N$), 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

$$\epsilon_i = \int_{x_i}^{x_{i+1}} |p(x) - \widetilde{p}(x)|\,\mathrm{d}x, \tag{1.57}$$

where the integral is evaluated numerically using the trapezoidal rule. 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 $\widetilde{p}(x)$ vary smoothly within each grid interval (x_i, x_{i+1}). Integrals of those functions over grid intervals can then be calculated accurately by using simple quadrature formulas. In our implementation of the algorithm, we use the extended Simpson rule with 51 equally-spaced points,

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

where $h = (x_{i+1} - x_i)/50$, $f_k = f(x_i + kh)$, and $f^{(\mathrm{iv})}(x^*)$ 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,

$$x_i,\ \xi_i = \mathcal{P}(x_i),\ a_i,\ b_i \qquad (i = 1, \ldots, N), \tag{1.59}$$

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,

$$\xi_i \le \xi < \xi_{i+1}, \tag{1.60}$$

using the binary-search method.

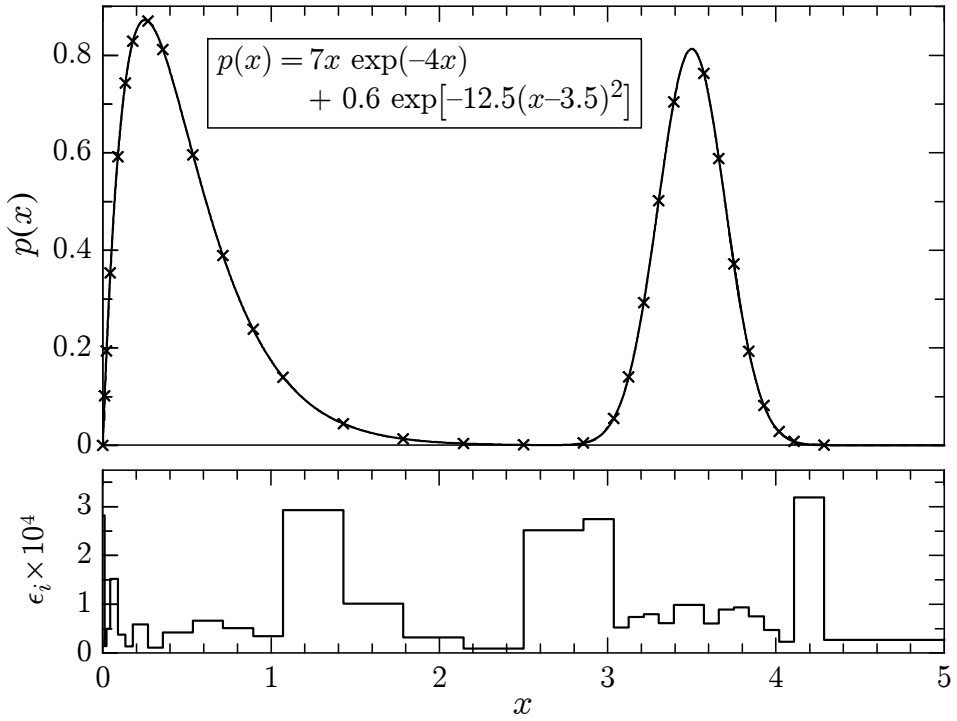

Figure 1.4: Rational interpolation of the continuous PDF defined by the analytical expression indicated in the inset and restricted to the interval (0,5). The crosses are grid points determined as described in the text with $N = 32$. The rational interpolating function given by Eq. (1.55) is represented 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

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

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 grid 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 $k = [\xi N]$) 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 2.2).

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.

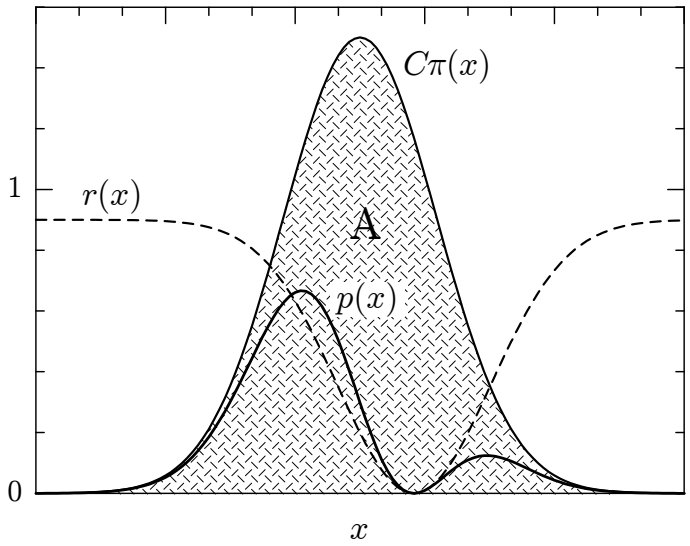

Figure 1.5: Random sampling from a distribution $p(x)$ using a rejection method.

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

$$p(x) = C\pi(x)r(x), \tag{1.62}$$

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 $(x, x + \mathrm{d}x)$ with probability $\pi(x)\,\mathrm{d}x$, 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 $(x, x + \mathrm{d}x)$ is equal to $p(x)\,\mathrm{d}x$ 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

$$\epsilon = \int_a^b r(x)\pi(x)\,\mathrm{d}x = \frac{1}{C}. \tag{1.63}$$

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)],

$$q(y) \equiv \int p(x, y)\, \mathrm{d}x, \qquad p(x|y) = \frac{p(x, y)}{q(y)},$$

the bivariate distribution can be expressed as

$$p(x, y) = q(y)\, p(x|y). \tag{1.64}$$

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,

$$q(x) \equiv \int p(x, y)\, \mathrm{d}y = \int q(y)\, p(x|y)\, \mathrm{d}y, \tag{1.65}$$

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

$$p(x) = \int w(y)\, p_y(x)\, \mathrm{d}y, \tag{1.66}$$

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

$$p(x) = \sum_i w_i\, p_i(x). \tag{1.67}$$

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

$$p_{\mathrm{G}}(x) = \frac{1}{\sqrt{2\pi}} \exp(-x^2/2). \tag{1.68}$$

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

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

Introducing the polar coordinates r and ϕ,

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

the PDF can be expressed as

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

We see that r and ϕ are independent random variables. The angle ϕ is distributed uniformly on (0,2π) 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

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

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 $\mathrm{var}(x) = 1$. The linear transformation

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

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

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

The PDF of X is

$$p(X) = p_{\mathrm{G}}(x)\,\frac{\mathrm{d}x}{\mathrm{d}X} = \frac{1}{\sigma\sqrt{2\pi}}\exp\left[-\frac{(X-m)^2}{2\sigma^2}\right], \tag{1.72}$$

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

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

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,

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

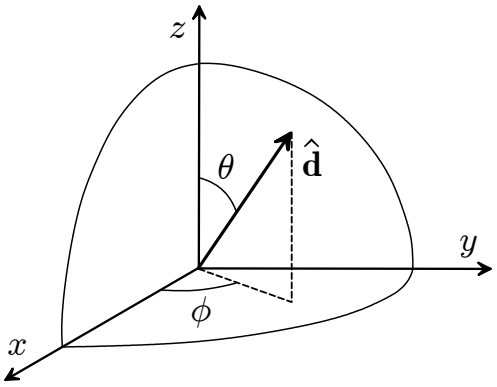

Figure 1.6: Polar and azimuthal angles of a direction vector.

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

$$\mu = (1 - \cos\theta)/2, \tag{1.76}$$

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

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

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

$$I = \int_a^b F(x)\,\mathrm{d}x, \tag{1.78}$$

which we recast in the form of an expectation value,

$$I = \int f(x)\,p(x)\,\mathrm{d}x \equiv \langle f \rangle, \tag{1.79}$$

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

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

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

$$\overline{f} \to I \quad \text{(in probability)}. \tag{1.81}$$

In statistical terminology, this means that $\overline{f}$, 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

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

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

$$\mathrm{var}\{f(x)\} = \int f^2(x)\, p(x)\, \mathrm{d}x - \langle f \rangle^2, \tag{1.83}$$

we obtain

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

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 $\mathrm{var}\{f(x)\}$ 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 $\overline{f}$. 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 $\overline{f}$ as a random variable, the PDF of which is, in principle, unknown. Its mean and variance are given by

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

and

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

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

$$\sigma_f \equiv \sqrt{\mathrm{var}(\overline{f})} = \sqrt{\frac{\mathrm{var}\{f(x)\}}{N}}, \tag{1.87}$$

gives a measure of the statistical uncertainty of the Monte Carlo estimate $\overline{f}$. 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 (James, 1980), which establishes that, in the limit $N \to \infty$, the PDF of $\overline{f}$ is a normal (Gaussian) distribution with mean $\langle f \rangle$ and standard deviation σ_f,

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

It follows that, for sufficiently large values of N, for which the theorem is applicable, the interval $\overline{f} \pm n\sigma_f$ 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 $\overline{f}$ 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,

$$\mu_3 \equiv \int [f(x) - \langle f \rangle]^3 \, p(x) \, \mathrm{d}x, \tag{1.89}$$

exists, the theorem is essentially satisfied when

$$|\mu_3| \ll \sigma_f^3 \sqrt{N}. \tag{1.90}$$

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 uncertainties 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 $\overline{f} \pm 3\sigma_f$. 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 $\mathrm{var}\{f(x)\}$, Eq. (1.83),

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

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

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

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 $\mathrm{var}\{f(x)\} = 0$. 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 Carlo method with that of conventional numerical quadrature. Let us thus consider the calculation of an integral over the D-dimensional unit cube,

$$I = \int_0^1 \mathrm{d}u_1 \int_0^1 \mathrm{d}u_2 \ldots \int_0^1 \mathrm{d}u_D \, F(u_1, u_2, \ldots, u_D), \tag{1.93}$$

where the integrand $F(u_1, u_2, \ldots, u_D)$ 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 subintervals of length $h = 1/n$; the centre of the i-th subinterval ($i = 1, \ldots, n$) is at

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

$$I = \sum_{i_1=1}^{n} \int_{x_{i_1}-h/2}^{x_{i_1}+h/2} \mathrm{d}u_1 \ldots \sum_{i_D=1}^{n} \int_{x_{i_D}-h/2}^{x_{i_D}+h/2} \mathrm{d}u_D \, F(u_1, \ldots, u_D). \tag{1.94}$$

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

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

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:

$$I_{\mathrm{num}} = h^D \sum_{i_1=1}^{n} \ldots \sum_{i_D=1}^{n} F(x_{i_1}, \ldots, x_{i_D}). \tag{1.96}$$

This formula involves only the N values of the function $F(u_1, \ldots, u_D)$ at the centres of the N cells; it is essentially equivalent to the trapezoidal rule (*e.g.*, Abramowitz and Stegun, 1972). The error, $(\Delta I)_{\mathrm{num}} = I - I_{\mathrm{num}}$, 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

$$\begin{aligned} (\Delta I)_{\mathrm{num}} \;=\; & \sum_{i_1=1}^{n} \int_{x_{i_1}-h/2}^{x_{i_1}+h/2} \mathrm{d}u_1 \ldots \sum_{i_D=1}^{n} \int_{x_{i_D}-h/2}^{x_{i_D}+h/2} \mathrm{d}u_D \, \frac{1}{2} \sum_{j=1}^{D} (u_j - x_{i_j})^2 \left[\partial^2 F / \partial u_j^2 \right]_{\{u_\ell = x_{i_\ell}\}} \\ \;=\; & h^{D+2} \frac{1}{24} \sum_{i_1=1}^{n} \ldots \sum_{i_D=1}^{n} \sum_{j=1}^{D} \left[\partial^2 F / \partial u_j^2 \right]_{\{u_\ell = x_{i_\ell}\}}. \end{aligned} \tag{1.97}$$

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

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

with

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

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 $p(u_1, \ldots, u_D) = 1$. 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)]

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

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. Furthermore, 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 (*e.g.*, Abramowitz and Stegun, 1972), 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 $N^{-4/D}$,

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

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 $F(u_1, \ldots, u_D)$ 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

methods stems not only from the convenient properties of statistical uncertainties for high-dimensionality problems, but also from the ease with which simulation 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 $A_{\mathrm{M}} = \Sigma n_i A_i$, where A_i is the atomic weight (mean relative atomic mass) of the i-th element. The number of molecules per unit volume is given by

$$\mathcal{N} = N_{\mathrm{A}} \frac{\rho}{A_{\mathrm{M}}}, \tag{1.102}$$

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. It is worth recalling that because photons are radiation quanta, in a Compton event the incident photon is absorbed and a "scattered" photon is emitted, usually with lower energy and in a direction different from that of the incident photon. In simulation codes, however, it is customary to consider the primary and scattered photons as the same particle.

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 $\hat{\mathbf{d}}$ parallel to the z axis, 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 $\mathbf{J}_{\rm inc}$[3]. We assume that particles interact only 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 $\mathrm{d}\Omega$, detects and counts all particles that enter its sensitive volume with energy in the interval $(E - W - \mathrm{d}W, E - W)$ (*i.e.*, particles that have lost energy between W and $W + \mathrm{d}W$). We consider that the detector is effectively screened from the incident beam so that only particles that have undergone an interaction are counted. Let $\dot{N}_{\rm count}$ denote the number of counts per unit time. The double-differential DCS (per unit solid angle and unit energy loss) is defined as

$$\frac{\mathrm{d}^2\sigma}{\mathrm{d}\Omega\,\mathrm{d}W} \equiv \frac{\dot{N}_{\rm count}}{|\mathbf{J}_{\rm inc}|\,\mathrm{d}\Omega\,\mathrm{d}W}. \tag{1.103}$$

The DCS has the dimensions of area/(solid angle × energy); the product $[\mathrm{d}^2\sigma/(\mathrm{d}\Omega\,\mathrm{d}W)] \times \mathrm{d}\Omega\,\mathrm{d}W$ 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 $\mathrm{d}\Omega$ with energy loss between W and $W + \mathrm{d}W$.

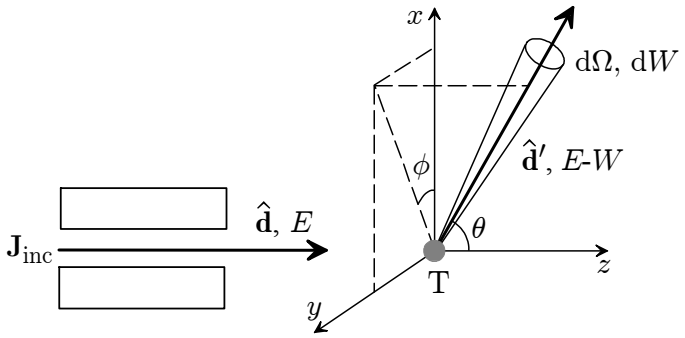

Figure 1.7: Schematic diagram of an experiment to measure the DCS. Incident particles move in the direction of the z axis; θ and ϕ are the polar and azimuthal scattering angles, respectively (cf. Fig. 1.6).

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

$$\frac{\mathrm{d}\sigma}{\mathrm{d}W} \equiv \int \frac{\mathrm{d}^2\sigma}{\mathrm{d}\Omega\,\mathrm{d}W}\,\mathrm{d}\Omega. \tag{1.104}$$

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

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

$$\sigma \equiv \int_0^E \frac{\mathrm{d}\sigma}{\mathrm{d}W}\,\mathrm{d}W = \int_0^E \left(\int \frac{\mathrm{d}^2\sigma}{\mathrm{d}\Omega\,\mathrm{d}W}\,\mathrm{d}\Omega \right) \mathrm{d}W. \tag{1.105}$$

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.

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 $\mathrm{d}s$ (see Fig. 1.8). What the incident particles see directly ahead of their path is a uniform distribution of $\mathcal{N}\,\mathrm{d}s$ 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 $J - \mathrm{d}J$, where $\mathrm{d}J = J\mathcal{N}\sigma\,\mathrm{d}s$ is the number of particles that undergo interactions per unit time and unit surface of the foil (note that $\mathcal{N}\sigma\,\mathrm{d}s$ is the fractional area covered by the spheres). Therefore, the interaction probability per unit path length is

$$\frac{\mathrm{d}J}{J}\,\frac{1}{\mathrm{d}s} = \mathcal{N}\sigma. \tag{1.106}$$

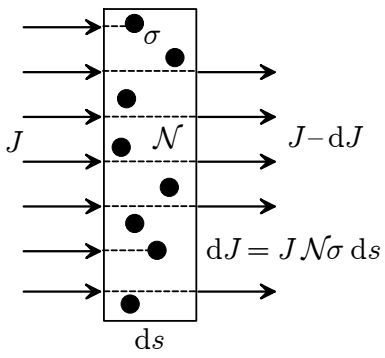

Figure 1.8: Attenuation of a beam through a thin material foil.

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

$$\mathcal{F}(s) = \int_s^\infty p(s')\,\mathrm{d}s'. \tag{1.107}$$

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

$$p(s) = \mathcal{N}\sigma \int_s^\infty p(s')\,\mathrm{d}s'. \tag{1.108}$$

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

$$p(s) = \mathcal{N}\sigma\, \exp\left[-s\,(\mathcal{N}\sigma)\right]. \tag{1.109}$$

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

$$\lambda \equiv \langle s \rangle = \int_0^\infty s\, p(s)\,\mathrm{d}s = \frac{1}{\mathcal{N}\sigma}. \tag{1.110}$$

Its inverse,

$$\lambda^{-1} = \mathcal{N}\sigma, \tag{1.111}$$

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

$$\frac{\mathrm{d}^2\sigma_\mathrm{A}(E;\theta,W)}{\mathrm{d}\Omega\,\mathrm{d}W} \qquad \text{and} \qquad \frac{\mathrm{d}^2\sigma_\mathrm{B}(E;\theta,W)}{\mathrm{d}\Omega\,\mathrm{d}W}, \tag{1.112}$$

where $\mathrm{d}\Omega$ 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

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

The total interaction cross section is

$$\sigma_{\mathrm{T}}(E) = \sigma_{\mathrm{A}}(E) + \sigma_{\mathrm{B}}(E). \tag{1.114}$$

The interaction probability per unit path length is

$$\lambda_{\mathrm{T}}^{-1} = \mathcal{N}\sigma_{\mathrm{T}}. \tag{1.115}$$

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

$$\lambda_{\mathrm{T}}^{-1} = \lambda_{\mathrm{A}}^{-1} + \lambda_{\mathrm{B}}^{-1}. \tag{1.116}$$

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

$$p(s) = \lambda_{\mathrm{T}}^{-1} \exp\left(-s/\lambda_{\mathrm{T}}\right). \tag{1.117}$$

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

$$\langle s \rangle = \lambda_{\mathrm{T}} = \left(\mathcal{N}\sigma_{\mathrm{T}}\right)^{-1}. \tag{1.118}$$

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

$$p_{\mathrm{A}} = \sigma_{\mathrm{A}}/\sigma_{\mathrm{T}} \qquad \text{and} \qquad p_{\mathrm{B}} = \sigma_{\mathrm{B}}/\sigma_{\mathrm{T}}. \tag{1.119}$$

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

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

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

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

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

$$\frac{\mathrm{d}^2\lambda_{\mathrm{A}}^{-1}(E;\theta,W)}{\mathrm{d}\Omega\,\mathrm{d}W} = \mathcal{N}\,\frac{\mathrm{d}^2\sigma_{\mathrm{A}}(E;\theta,W)}{\mathrm{d}\Omega\,\mathrm{d}W}\,. \tag{1.122}$$

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

$$\lambda_{\mathrm{A}}^{-1} = \int \mathrm{d}W \int 2\pi\sin\theta\,\mathrm{d}\theta\,\frac{\mathrm{d}^2\lambda_{\mathrm{A}}^{-1}(E;\theta,W)}{\mathrm{d}\Omega\,\mathrm{d}W} = \mathcal{N}\sigma_{\mathrm{A}}. \tag{1.123}$$

In the literature, the product $\mathcal{N}\sigma_{\mathrm{A}}$ 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 tracks

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)]

$$s = -\lambda_{\mathrm{T}}\,\ln\xi. \tag{1.124}$$

The following interaction occurs at the position

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

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 $p_{\mathrm{A,B}}(E;\theta,W)$, 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 $\phi = 2\pi\xi$.

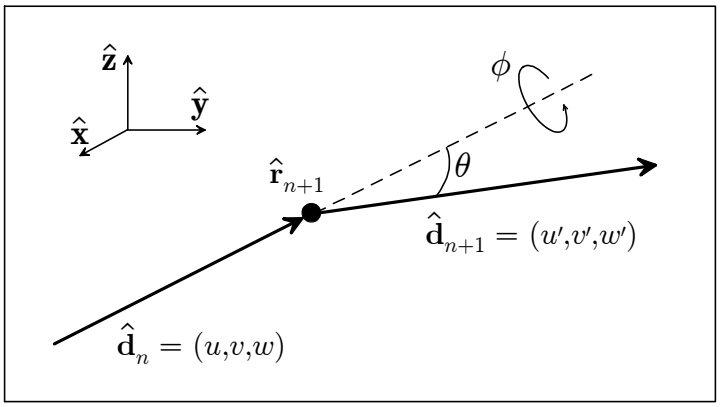

Figure 1.9: Angular deflections in single-scattering events.

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 is[4]

$$\begin{pmatrix} \sin\theta\cos\phi \\ \sin\theta\sin\phi \\ \cos\theta \end{pmatrix} = \mathcal{R}(\phi\hat{\mathbf{z}})\mathcal{R}(\theta\hat{\mathbf{y}}) \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \tag{1.126}$$

where

$$\mathcal{R}(\theta\hat{\mathbf{y}}) = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \qquad \text{and} \qquad \mathcal{R}(\phi\hat{\mathbf{z}}) = \begin{pmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{1.127}$$

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

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

the rotation $\mathcal{R}(-\vartheta\hat{\mathbf{y}})\mathcal{R}(-\varphi\hat{\mathbf{z}})$ 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

[4]Vectors 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.

of rotations of the initial direction vector: 1) $\mathcal{R}(-\vartheta\hat{\mathbf{y}})\mathcal{R}(-\varphi\hat{\mathbf{z}})$, which transforms $\hat{\mathbf{d}}_n$ into $\hat{\mathbf{z}}$; 2) $\mathcal{R}(\phi\hat{\mathbf{z}})\mathcal{R}(\theta\hat{\mathbf{y}})$, which rotates $\hat{\mathbf{z}}$ according to the sampled polar and azimuthal scattering angles; and 3) $\mathcal{R}(\varphi\hat{\mathbf{z}})\mathcal{R}(\vartheta\hat{\mathbf{y}})$, which inverts the rotation of the first step. Hence

$$\mathcal{R}(\theta,\phi) = \mathcal{R}(\varphi\hat{\mathbf{z}})\mathcal{R}(\vartheta\hat{\mathbf{y}})\mathcal{R}(\phi\hat{\mathbf{z}})\mathcal{R}(\theta\hat{\mathbf{y}})\mathcal{R}(-\vartheta\hat{\mathbf{y}})\mathcal{R}(-\varphi\hat{\mathbf{z}}). \tag{1.129}$$

The final direction vector is

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

and its direction cosines are

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

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

$$u' = \pm\sin\theta\,\cos\phi, \quad v' = \pm\sin\theta\,\sin\phi, \quad w' = \pm\cos\theta. \tag{1.132}$$

In addition, 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 renormalising $\hat{\mathbf{d}}_{n+1}$. The change of direction expressed by Eqs. (1.131) and (1.132) is performed by the subroutine `DIRECT` (see the PENELOPE source file).

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 $E_{\rm abs}$, which is the energy where particles are assumed to be effectively stopped and absorbed in the medium.

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 process[5], *i.e.*, "future values of a random

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

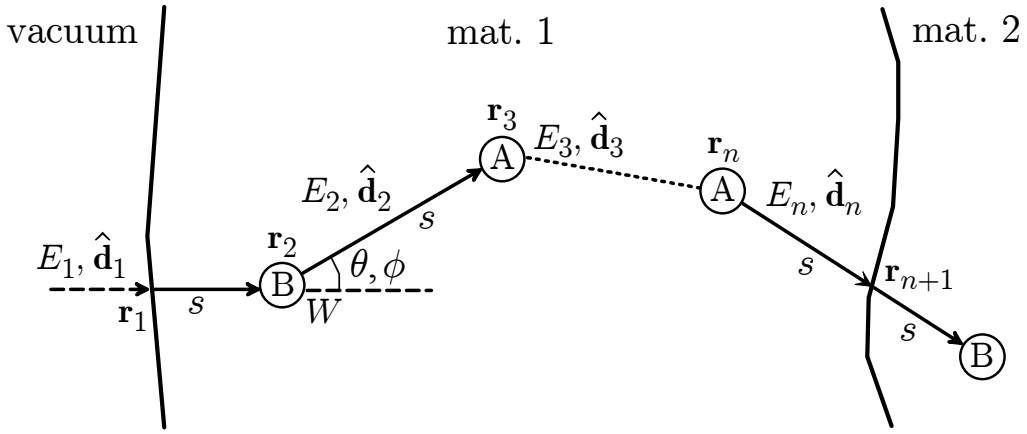

Figure 1.10: Generation of random trajectories using detailed simulation. A particle enters material 1 from the vacuum and, after multiple interactions, crosses the interface between materials 1 and 2.

variable (interaction event) are statistically determined by present events and depend 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 $\lambda_{\mathrm{T},1}$ and $\lambda_{\mathrm{T},2}$), 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 $\lambda_{\mathrm{T},1} + \lambda_{\mathrm{T},2}$, 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 $s_{\rm plane}$ of the track segment between the two real interactions immediately before and after the crossing. It is found that the average value of $s_{\rm plane}$ is $2\lambda_{\rm T}$, in spite of the fact that the free path length between consecutive collisions was sampled from an exponential PDF with the mean free path $\lambda_{\rm T}$ [yes, the scored values $s_{\rm plane}$ 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 $\lambda_{\rm T}$.

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 $s_{\rm max}$. To accomplish this, we still sample the free path length s to the next interaction from the exponential PDF (1.117), but when $s > s_{\rm max}$ we only let the particle advance a distance $s_{\rm max}$ 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 $s_{\rm max}$, a real interaction is simulated. After the interaction (either real or delta), we sample a new free path s, move the particle a distance $s' = \min(s, s_{\rm max})$, *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

$$p_\delta = \int_{s_{\rm max}}^{\infty} p(s)\,{\rm d}s = \exp(-s_{\rm max}/\lambda_{\rm T}). \tag{1.133}$$

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

$$p(s)\,{\rm d}s = p_\delta^n\ \lambda_{\rm T}^{-1}\exp(-s'/\lambda_{\rm T})\,{\rm d}s = \lambda_{\rm T}^{-1}\exp(-s/\lambda_{\rm T})\,{\rm d}s, \tag{1.134}$$

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),

$$Q = \int q(x)\, p(x)\, \mathrm{d}q, \tag{1.135}$$

where $p(x)$ represents the PDF of the set x of random variables which determine the score $q(x)$. Although in radiation transport problems this PDF is usually unknown, the simulation of individual showers provides a practical method to generate a set of random values of the variables x and the associated value of $q(x)$. The only difference to the case of Monte Carlo integration considered above is that now the PDF $p(x)$ describes a cascade of random interaction events, each with its characteristic PDF. The Monte Carlo estimate of Q is

$$\overline{Q} = \frac{1}{N} \sum_{i=1}^{N} q_i, \tag{1.136}$$

where q_i is the score of the i-th simulated shower. Thus, for instance, the average energy E_{dep} deposited within the water phantom per incident electron is obtained as

$$E_{\mathrm{dep}} = \frac{1}{N} \sum_{i=1}^{N} e_i, \tag{1.137}$$

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

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

As mentioned above, we shall usually express the simulation result in the form $\overline{Q} \pm 3\sigma_Q$, so that the interval $(\overline{Q} - 3\sigma_Q,\ \overline{Q} + 3\sigma_Q)$ 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,

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

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. $D(z)\,\mathrm{d}z$ is the average energy deposited at depths between z and $z+\mathrm{d}z$ 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

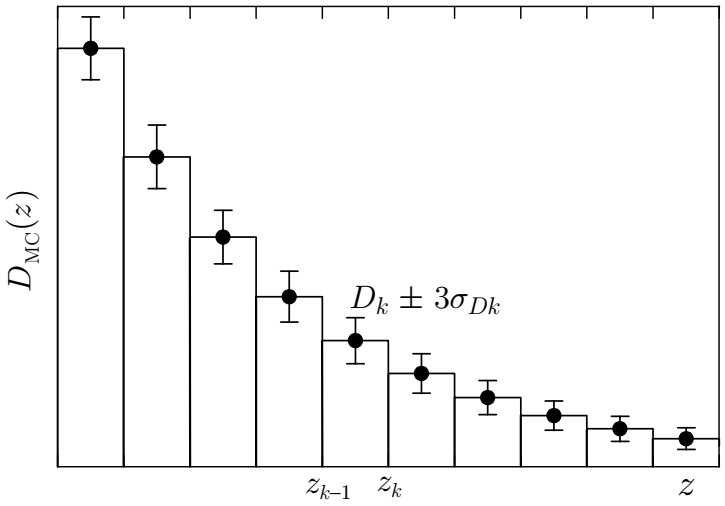

Figure 1.11: Typical continuous distribution generated by a Monte Carlo simulation, represented as a histogram. With the definition given by Eq. (1.143), the area of the histogram is independent of the bin size, except for statistical uncertainties.

backscattered radiation), $E_{\rm dep}$ is less than the kinetic energy $E_{\rm inc}$ of the incident electrons. We are interested in determining $D(z)$ in a limited depth interval, say from $z = z_{\rm min}$ to $z = z_{\rm max}$. The calculation proceeds as follows. First of all, we have to select a partition of the interval $(z_{\rm min}, z_{\rm max})$ into M different depth bins $(z_{k-1},\, z_k)$, with $z_{\rm min} = z_0 < z_1 < \cdots < z_M = z_{\rm max}$. 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

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

and is affected by a statistical uncertainty

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

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

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

with

$$D_k \equiv \frac{1}{z_k - z_{k-1}} E_k, \qquad \sigma_{Dk} \equiv \frac{1}{z_k - z_{k-1}} \sigma_{Ek}. \tag{1.143}$$

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, $D_{\rm MC}(z)$ 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.

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=q_ij
  LQ=i
ELSE
  QP=QP+q_ij
ENDIF
```

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

For some quantities (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, analogue[6] 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 uncertainty 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 practically oriented, review of variance-reduction methods in radiation transport has been given by Bielajew and Rogers (1988).

As we have seen, a Monte Carlo simulation is equivalent to the calculation of the integral defining the expectation value of the quantity of interest [Eq. (1.135)],

$$\overline{Q} \simeq \int q(x)\, p(x)\, dx. \tag{1.144}$$

Statistical uncertainties are evaluated from the variance of the sample, which is an unbiased estimator of the following integral [cf. Eq. (1.16)]

$$\mathrm{var}_Q \simeq \int \left[q(x) - \overline{Q}\right]^2 p(x)\, \mathrm{d}x = \int q^2(x)\, p(x)\, \mathrm{d}x - \overline{Q}^2. \tag{1.145}$$

The aim of variance-reduction techniques is to optimize the *efficiency* (1.92) of the simulation by re-arranging the integrand in Eq. (1.144). In radiation transport problems, this is frequently accomplished by assigning a weight w to each particle, typically set equal to unity for primary particles, which is suitably manipulated as the particle history progresses. We can write

$$\overline{Q} = \int wq(x)\, \frac{p(x)}{w}\, dx = \int q'(x)\, p'(x)\, dx, \tag{1.146}$$

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

$$\mathrm{var}'_Q = \int q'^2(x)\, p'(x)\, \mathrm{d}x - \overline{Q}^2 = \int wq^2(x)\, p(x)\, \mathrm{d}x - \overline{Q}^2. \tag{1.147}$$

Thus, with an adequate choice of weights, we could make $\mathrm{var}'_Q < \mathrm{var}_Q$ and, consequently, increase the efficiency of the simulation of the quantity of interest; the weighted simulation would then attain the desired accuracy with less computer time.

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

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 ($\sim$ 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 $\lambda_{\rm A}$ of the real process by a shorter one, $\lambda_{\rm A,f} = \lambda_{\rm A}/\mathcal{F}$ with $\mathcal{F} > 1$. This implies that between each pair of "real" A interactions we will have, on average, $\mathcal{F} - 1$ "forced" A interactions. 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 real interactions.

For the sake of programming simplicity, the length of the free jump to the next A interaction (real or forced) is sampled from the exponential distribution with the reduced mean free path $\lambda_{\rm A,f}$. 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 A interactions (real or forced) of a particle with weight w are given a weight $w_{\rm s} = w/\mathcal{F}$. Secondary particles generated in real interactions of types other than A are given a weight equal to that of their parent particle.

(ii) Interactions of type A (real and forced) are simulated to determine the energy loss and possible emission of secondary radiation, but the state variables of the interacting particle are altered only when the interaction is real. As the probability of having a real A interaction is $1/\mathcal{F}$, 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 (forced A interaction).

(iii) A weight $w_E = w/\mathcal{F}$ is given to the deposited energy (and to any other alteration of the medium such as, *e.g.*, charge deposition) that results from A interactions (real and forced) of a particle with weight w. For non-forced interactions of types other than A, $w_E = w$.

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.

Let w_{i1} and q_{i1} denote the weight and the contribution to the score of the i-th primary, and let w_{ij} and q_{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 Q obtained from the N simulated histories is

$$\overline{Q} = \frac{1}{N} \sum_{i,j} w_{ij} q_{ij}. \tag{1.148}$$

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

$$\sigma'_Q = \sqrt{\mathrm{var}'_Q} = \sqrt{\frac{1}{N}\left[\frac{1}{N}\sum_i \left(\sum_j w_{ij} q_{ij}\right)^2 - \overline{Q}^2\right]}. \tag{1.149}$$

Quantities directly related to forced interactions will have a reduced statistical uncertainty, 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.

Interaction forcing can effectively reduce the statistical uncertainties of some simulation results, particularly the emission of secondary radiation and the energy deposition in very thin volumes. However, this technique 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.

It should be noted that our implementation of interaction forcing introduces forced interactions randomly along the particle trajectory, independently of the geometry, and keeps the weight of the transported particle unaltered. This is at odds with the conventional technique of interaction forcing, where forced events occur with probability equal to unity and the scored quantities are weighted by factors that depend on the geometry (see Jenkins *et al.*, 1988, p. 411–412).

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, thus saving part of the numerical work that would be wasted tracking particles that do not contribute to the scores. 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. (1.148) and (1.149). The effectiveness of these techniques 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. It should also be noted that splitting and Russian roulette can be applied in conjunction with interaction forcing and other variance reduction techniques.

1.6.3 Delta scattering of photons

This technique, also known as the Woodcock method (Woodcock *et al.*, 1965; Coleman, 1968; Sempau *et al.*, 2000), is only applicable to photons. It takes advantage of the high penetration of photons (see Chapter 2) to simplify tracking of these particles through material systems with complex geometries. Photons are transported freely across the system using an augmented inverse mean free path, Λ^{-1}, which is larger than the actual total inverse mean free paths in all the materials crossed by a trajectory ray. The event at the end of each free flight may be either a real interaction or a delta interaction, which has no effect on the photon. Delta interactions occur with probability $1 - \lambda_{\mathrm{T}}^{-1}/\Lambda^{-1}$, where $\lambda_{\mathrm{T}}^{-1}$ is the actual total inverse mean free path in the current material. Thus, the probability of real interactions per unit path length, which is equal to $\lambda_{\mathrm{T}}^{-1}$, remains unaltered. This procedure avoids the need for computing intersections of particle rays with interfaces at the expense of having to determine which material is at the end of each free flight. Hence delta scattering will improve the efficiency only for those geometries where locating a particle (*i.e.*, finding the material at its current position) is faster than normal tracking.

1.6.4 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. For instance, when the system is symmetric under rotations about an axis, splitting can be made more effective if the position and direction of each of the $\mathcal{S}$ split particles are rotated around the symmetry axis by a random angle $\varphi = 2\pi\xi$. Thus, the $\mathcal{S}$ split particles are

assigned different positions and directions, and this gives a net information gain and an increase in efficiency (Bush *et al.*, 2007).

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 simulation of the track can be stopped. Range rejection is not adequate for photon transport simulation because 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 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 the photoelectric effect, coherent (Rayleigh) scattering, incoherent (Compton) scattering and electron-positron pair production, see Fig. 2.1. Other interactions, such as photonuclear absorption, occur with much smaller probability and can be disregarded for most practical purposes (see Hubbell *et al.*, 1980). We start by assuming that photons are unpolarised; the simulation of interactions of polarised photons is treated in Section 2.7, where arbitrary photon-polarisation states are described in terms of the Stokes parameters.

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 six 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 Section B.2, Appendix B. Furthermore, the random sampling from these DCSs can be done analytically and, hence, exactly. Only coherent scattering requires a simple preparatory numerical step.

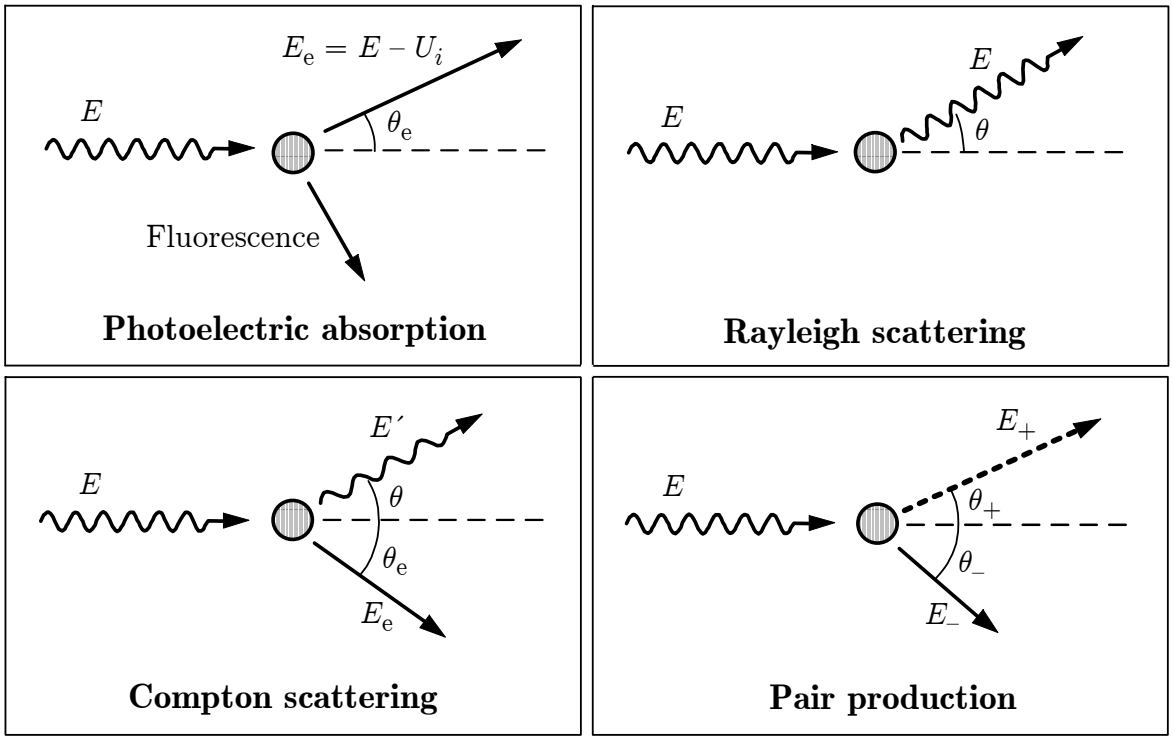

Figure 2.1: Basic interactions of photons with matter.

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.*,

$$\kappa \equiv \frac{E}{\mathrm{m_e}c^2}. \tag{2.1}$$

2.1 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 observed[1]. To represent the atomic states, we can adopt an independent-electron model, such as the Dirac–Hartree–Fock–Slater (DHFS) self-consistent model

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

(Pratt *et al.*, 1973a; Salvat and Fernández-Varea, 2019), in which each electron occupies a single-particle orbital, $\psi_{n\ell jm}$, where n is the principal quantum number, ℓ is the orbital angular momentum quantum number (which determines the parity of the orbital), j is the total angular momentum quantum number, and m is the magnetic quantum number. The set of orbitals with the same values of the quantum numbers n, ℓ and j have the same ionisation energy and constitute a subshell[2]. That is, a subshell Si ($n_i l_i j_i$) can accommodate up to $2j_i - 1$ electrons (in orbitals with different magnetic quantum numbers m_i) with characteristic ionisation energy U_i. Notice that the subshell ionisation energies are positive; the quantity $-U_i$ represents the "binding" energy of an electron. Figure 2.2 (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 ionisation energies recommended by Carlson (1975), which were obtained from a combination of experimental data and theoretical calculations. Figure 2.3 displays the subshell ionisation energies of neutral atoms, $Z = 1$ to 99, that are larger than 100 eV.

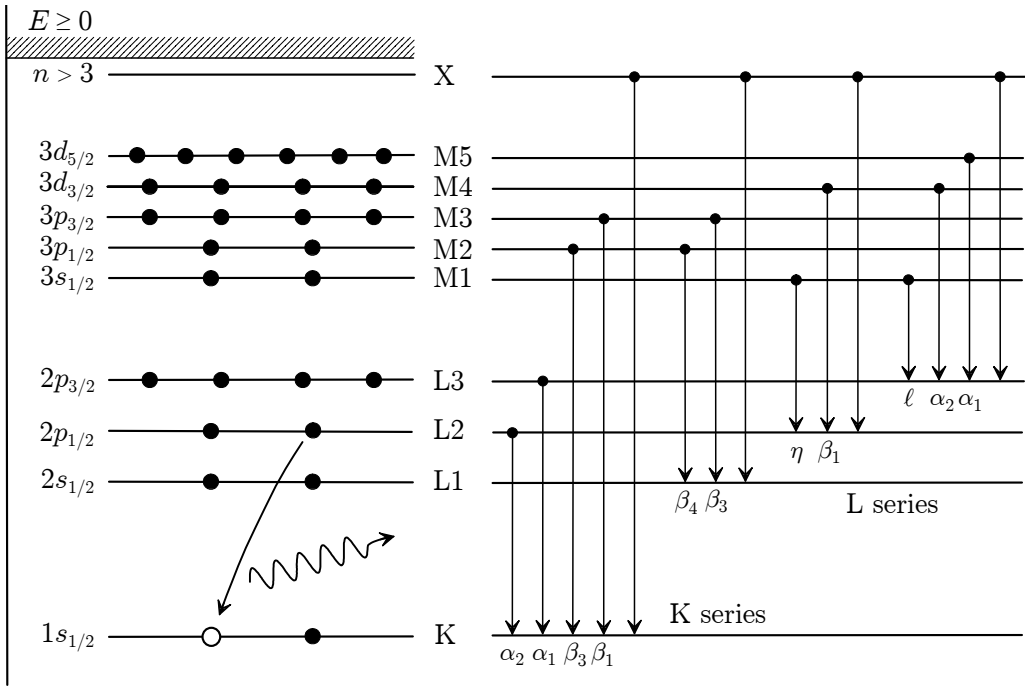

Figure 2.2: Various notations for inner atomic electron subshells (left) and allowed radiative transitions (right) to these subshells. Transitions that are different from the ones indicated in the diagram (*e.g.*, K-M4) are also possible, but their transition probabilities are extremely small.

[2] The set of orbitals with the same principal quantum number n constitute a shell. Shells with $n > 1$ consist of various subshells with $\ell = 0, 1, \ldots, n-1$. For each value of ℓ there are two possible values of j, $j = \ell \pm 1/2$, except for $\ell = 0$ and $\ell = n - 1$ which admit only one value, $j = 1/2$ and $j = n - 1/2$, respectively.

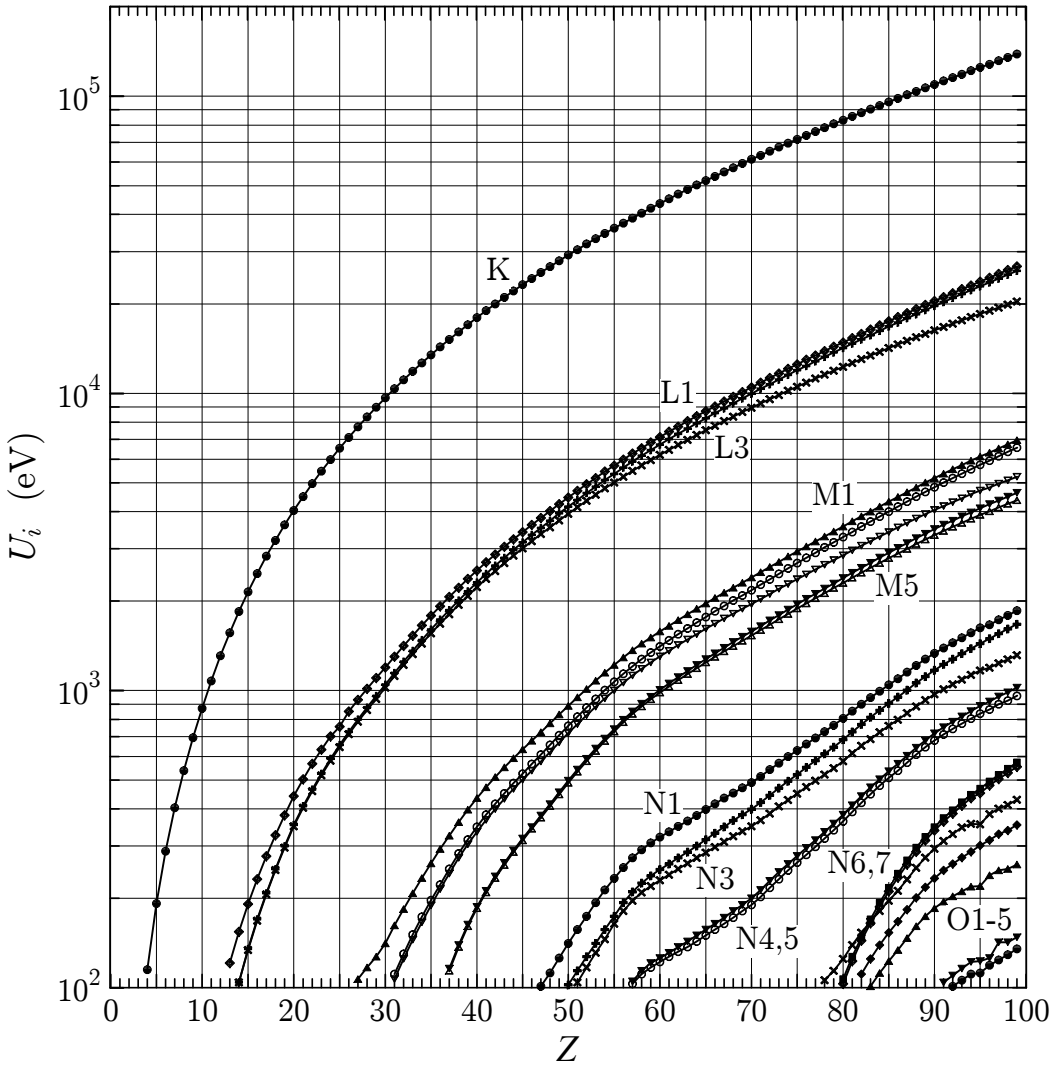

Figure 2.3: Shell ionisation energies of neutral atoms, U_i, in eV. Symbols represent values given by (Carlson, 1975), joined by straight segments for visual aid. (Adapted from Salvat and Fernández-Varea, 2009).

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" subshell i, which leaves the parent atom with kinetic energy $E_\mathrm{e} = E - U_i$. Evidently, photoionisation of a given subshell 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. 2.4).

The photoelectric cross sections used in previous versions of PENELOPE were obtained by interpolation from a numerical table extracted from the LLNL Evaluated Photon Data Library (EPDL; Cullen *et al.*, 1997). The present version of the code

uses a new database that has been calculated with the Fortran program PHOTACS of Sabbatucci and Salvat (2016). This program uses the same theory as in the original calculations by Scofield (1973) of the EPDL tables, but implements more accurate numerical algorithms. Our cross sections practically coincide with those in the EPDL, although they are tabulated in a denser grid of energies to describe the structure of the cross section near absorption edges. Because of numerical difficulties, the program PHOTACS is able to compute the partial cross section of a given subshell up to a certain maximum energy. For higher energies the partial cross section was obtained by extrapolating the numerical table using an analytical formula derived by Pratt (1960a), which is different from the extrapolation scheme adopted in the EPDL. The database was generated with the screening normalisation correction (Pratt, 1960b; Pratt *et al.*, 1973a,b; Saloman *et al.*, 1988; Sabbatucci and Salvat, 2016) included. This correction seems to improve agreement with dose measurements (see Büermann *et al.*, 2006), although its consistency with the theory underlying the calculations of the photoeffect is debatable. An alternative database of photoelectric cross sections generated without the screening normalization correction is included in the distribution package (see Section 7.5).

The present database for photoelectric absorption consists of tables of the total atomic cross section $\sigma_{\rm ph}(E)$ and the cross sections for the K shell, and the L, M and N subshells with binding energies higher than about 40 eV, $\sigma_{{\rm ph},i}(E)$ (i = K, L1 to L3, M1 to M5, and N1 to N7) for the elements $Z = 1 - 99$, which span the energy range from 50 eV to 1 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 keV $< E <$ 1 keV, 100–200% for 0.1 keV $< E <$ 0.5 keV, and 1000% for $E <$ 100 eV. Notice that these cross sections were obtained from 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.

For compound materials (and also for mixtures) the molecular cross section $\sigma_{\rm ph}(E)$ 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. 2.4). In PENELOPE, the molecular cross section is defined by means of a table of numerical values $\sigma_{\rm ph}(E_i)$ for a grid of photon energies E_i, suitably spaced to account for the fast variation of the cross section near absorption edges, which is stored in memory. Photon mean free paths are determined by linear log-log interpolation from 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).

2.1.1 Simulation of photoelectron emission

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

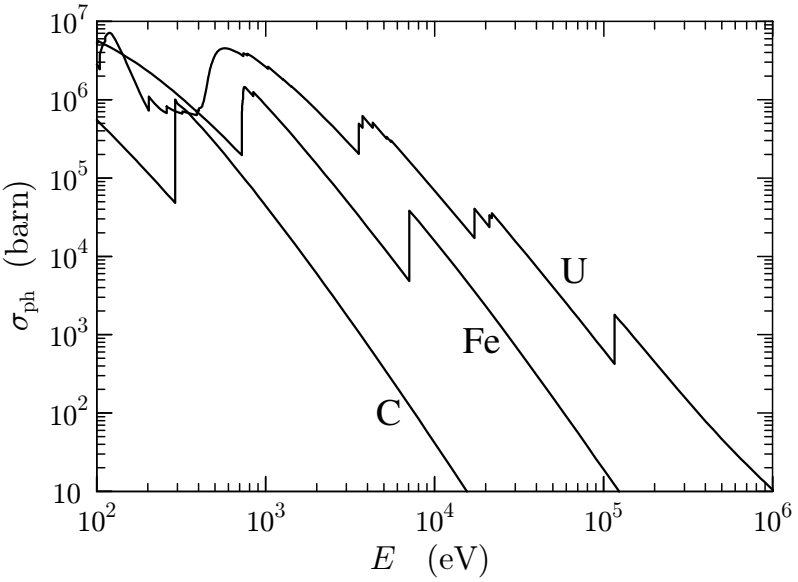

Figure 2.4: Atomic photoelectric cross sections for carbon, iron and uranium as functions of the photon energy E.

PDF

$$p_i = \sigma_{{\rm ph},i}(Z,E)/\sigma_{\rm ph}(Z,E), \tag{2.2}$$

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

$$p_{\rm outer} = 1 - p_{\rm K} - p_{\rm L1} - \ldots - p_{\rm N7}. \tag{2.3}$$

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

2.1.1.1 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 $\theta_{\rm e}$ (Fig. 2.1) and $\phi_{\rm e}$. We consider that the incident photon is not polarised and, hence, the angular distribution of photoelectrons is independent of $\phi_{\rm e}$, 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 the first-order plane-wave Born approximation with K-shell hydrogenic electron wave functions. The Sauter DCS (per electron) can be written as

$$\frac{\mathrm{d}\sigma_\mathrm{ph}}{\mathrm{d}\Omega_\mathrm{e}} = \alpha^4 r_\mathrm{e}^2 \left(\frac{Z}{\kappa}\right)^5 \frac{\beta^3}{\gamma} \frac{\sin^2\theta_\mathrm{e}}{(1-\beta\cos\theta_\mathrm{e})^4} \left[1 + \frac{1}{2}\gamma(\gamma-1)(\gamma-2)(1-\beta\cos\theta_\mathrm{e})\right], \quad (2.4)$$

where α is the fine-structure constant, r_e is the classical electron radius, $\kappa = E/(\mathrm{m_e}c^2)$ and

$$\gamma = 1 + E_\mathrm{e}/(\mathrm{m_e}c^2), \qquad \beta = \frac{\sqrt{E_\mathrm{e}(E_\mathrm{e}+2\mathrm{m_e}c^2)}}{E_\mathrm{e}+\mathrm{m_e}c^2}. \quad (2.5)$$

Strictly speaking, the DCS (2.4) is adequate only for ionisation of the K shells 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 subshell 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 $\nu = 1 - \cos\theta_\mathrm{e}$, the angular distribution of photoelectrons can be expressed in the form

$$p(\nu) = (2-\nu)\left[\frac{1}{A+\nu} + \frac{1}{2}\beta\gamma(\gamma-1)(\gamma-2)\right]\frac{\nu}{(A+\nu)^3}, \qquad A = \frac{1}{\beta} - 1, \quad (2.6)$$

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

$$p(\nu) = g(\nu)\pi(\nu) \quad (2.7)$$

with

$$g(\nu) = (2-\nu)\left[\frac{1}{A+\nu} + \frac{1}{2}\beta\gamma(\gamma-1)(\gamma-2)\right] \quad (2.8)$$

and

$$\pi(\nu) = \frac{A(A+2)^2}{2}\frac{\nu}{(A+\nu)^3}. \quad (2.9)$$

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)

$$\int_0^\nu \pi(\nu')\,\mathrm{d}\nu' = \xi, \quad (2.10)$$

which can be solved analytically to give

$$\nu = \frac{2A}{(A+2)^2 - 4\xi}\left[2\xi + (A+2)\xi^{1/2}\right]. \tag{2.11}$$

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. (2.11).

(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.

2.2 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 can be calculated using non-relativistic perturbation theory (Sakurai, 1967; Baym, 1974). For photons with energy E much higher than the ionisation energy of the K shell (high-frequency limit), the DCS is given approximately by (Born, 1969)

$$\frac{\mathrm{d}\sigma_{\mathrm{Ra}}}{\mathrm{d}\Omega} = \frac{\mathrm{d}\sigma_{\mathrm{T}}}{\mathrm{d}\Omega}\left[F(q,Z)\right]^2, \tag{2.12}$$

where

$$\frac{\mathrm{d}\sigma_{\mathrm{T}}(\theta)}{\mathrm{d}\Omega} = r_e^2\frac{1+\cos^2\theta}{2} \tag{2.13}$$

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

$$q = |\mathbf{p} - \mathbf{p}'| = 2(E/c)\sin(\theta/2) = (E/c)\left[2(1-\cos\theta)\right]^{1/2}. \tag{2.14}$$

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

$$x \equiv \frac{q\ 10^{-8}\mathrm{cm}}{4\pi\hbar} = 20.6074\ \frac{q}{\mathrm{m_e}c} \tag{2.15}$$

is normally used instead of q.

The atomic form factor is the Fourier transform of the atomic electron density $\rho(\mathbf{r})$ which, for a spherically symmetrical atom, is given by

$$F(q,Z) = 4\pi \int_0^\infty \rho(r) \frac{\sin(qr/\hbar)}{qr/\hbar} r^2\, \mathrm{d}r. \tag{2.16}$$

$F(q,Z)$ is a monotonically decreasing function of q that varies from $F(0,Z) = Z$ to $F(\infty,Z) = 0$ (see Fig. 2.5). 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 from the EPDL (Cullen *et al.*, 1997), which were calculated by Hubbell *et al.* (1975, 1977). 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).

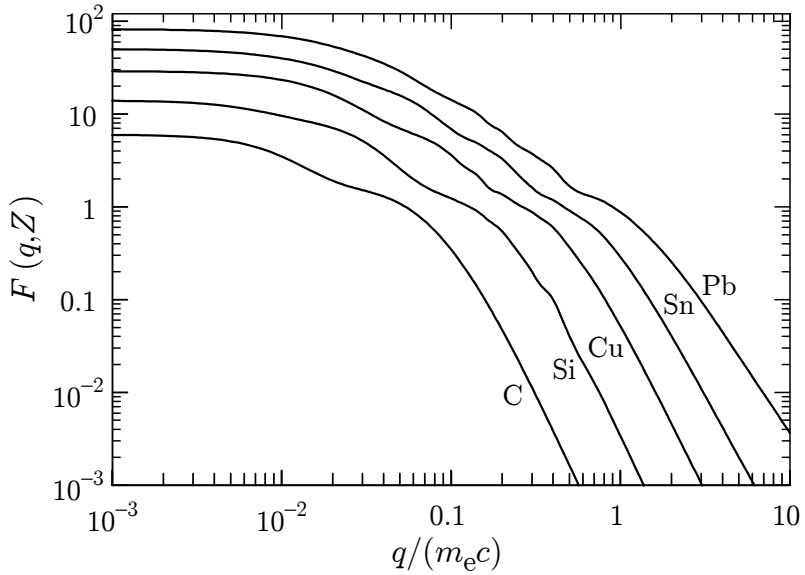

Figure 2.5: Atomic form factors of neutral atoms of the indicated elements, taken from the EPDL (Cullen *et al.*, 1997).

The total atomic cross section for coherent scattering is

$$\sigma_{\mathrm{Ra}} = \int \frac{\mathrm{d}\sigma_{\mathrm{Ra}}}{\mathrm{d}\Omega}\,\mathrm{d}\Omega = \pi r_{\mathrm{e}}^2 \int_{-1}^{1} \left(1+\cos^2\theta\right) \left[F(q,Z)\right]^2\, \mathrm{d}(\cos\theta). \tag{2.17}$$

Introducing q, Eq. (2.14), 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

$$\sigma_{\mathrm{Ra}} \simeq \frac{8}{3}\pi r_{\mathrm{e}}^2 \, Z^2. \tag{2.18}$$

In the high-energy limit, we get

$$\sigma_{\mathrm{Ra}} \propto E^{-2}. \tag{2.19}$$

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

Strictly speaking, expression (2.12) is approximately valid only for photons with energy well above the K absorption edge. More elaborate calculations (*e.g.*, Cromer and Liberman, 1970) show that the atomic DCS for coherent scattering of unpolarised photons can be expressed as

$$\frac{\mathrm{d}\sigma_{\mathrm{Ra}}}{\mathrm{d}\Omega} = r_{\mathrm{e}}^2 \frac{1+\cos^2\theta}{2} \left| F(q, Z) + f' + \mathrm{i}f'' \right|^2, \tag{2.20}$$

where the complex quantity $f' + \mathrm{i}f''$ is known as the anomalous scattering factor or as the dispersion correction to the form factor. The adjective "anomalous" refers to the fast variation of this quantity for photon energies around the absorption edge. Elaborate calculations show that the anomalous scattering factor is practically independent of the scattering angle θ (see Kissel *et al.*, 1995, and references therein). The total atomic cross sections σ_{Ra} used in PENELOPE are from the EPDL (Cullen *et al.*, 1997), and were calculated from the DCS given by Eq. (2.20), *i.e.*, including the anomalous scattering factor. Figure 2.6 displays these cross sections for aluminium, silver and gold atoms.

Anomalous scattering factors are available for all elements (Cullen *et al.*, 1997) but not for arbitrary compounds. On the other hand, the molecular cross section obtained from the additivity approximation has a complicated structure that is not suited for analytical sampling methods. For the sake of simplicity, the polar angle θ is sampled from the DCS given by Eq. (2.12), *i.e.*, disregarding anomalous scattering effects. This approximate method is sufficient for most applications because, at the energies where anomalous scattering effects become significant, coherent scattering is much less probable than photoelectric absorption (see Fig. 2.12 below).

2.2.1 Simulation of coherent scattering events

The PDF of the angular deflection, $\cos\theta$, can be written as [see Eqs. (2.12) and (2.13); normalisation is irrelevant here]

$$p_{\mathrm{Ra}}(\cos\theta) = \frac{1+\cos^2\theta}{2} \left[F(q, Z) \right]^2, \tag{2.21}$$

where the momentum transfer q can take values in the interval from 0 to

$$q_{\max} = 2(E/c) = 2\mathrm{m_e}c\,\kappa. \tag{2.22}$$

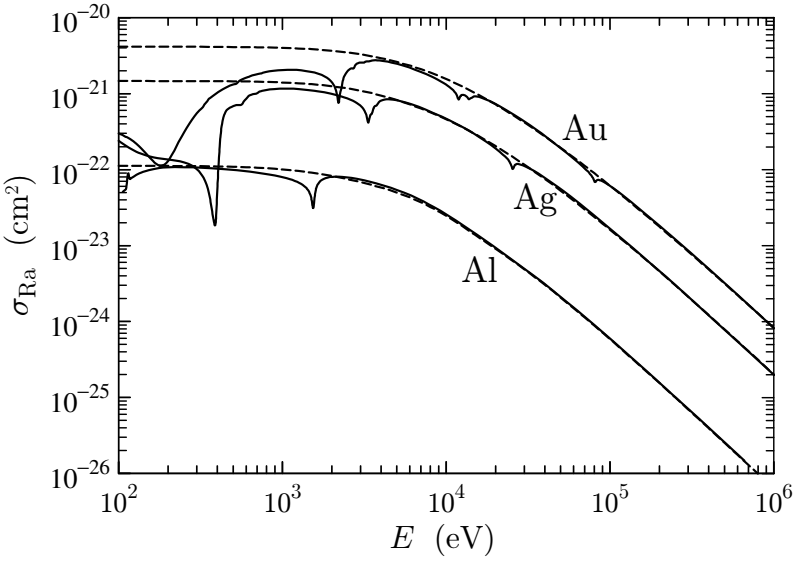

Figure 2.6: Total cross sections for Rayleigh scattering of photons by neutral atoms of aluminium, silver and gold, as functions of the energy E of the incident photon. The solid curves represent values taken from the EPDL (Cullen *et al.*, 1997). Dashed curves are cross sections calculated by numerical integration of the approximate DCS given by Eq. (2.12).

This PDF can be factorised in the form

$$p_{\mathrm{Ra}}(\cos\theta) = g(\cos\theta)\pi(q^2) \tag{2.23}$$

with

$$g(\cos\theta) \equiv \frac{1+\cos^2\theta}{2} \qquad \text{and} \qquad \pi(q^2) \equiv [F(q,Z)]^2\,. \tag{2.24}$$

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

The function $\pi(q^2)$ can be considered as the (unnormalised) PDF of the variable q^2, which takes values in the interval $(0, q^2_{\mathrm{max}})$. In PENELOPE, random values of q^2 distributed according to this PDF are generated by using the RITA algorithm (see Section 1.2.4) with 150 grid points.

The angular deflection $\cos\theta$ can then 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 q^2 from the distribution $\pi(q^2)$, restricted to the interval $(0, q^2_{\mathrm{max}})$.

(ii) Set

$$\cos\theta = 1 - \frac{1}{2}\,\frac{q^2}{\kappa^2}. \tag{2.25}$$

(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.

2.3 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 subshells are considered separately. After a Compton interaction with the i-th subshell, the active target electron is ejected to a free state with kinetic energy $E_\mathrm{e} = E - E' - U_i > 0$, where U_i is the ionisation energy of the considered subshell, and the residual atom is left in an excited state with a vacancy in the i-th subshell.

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)]

$$E' \equiv \frac{E}{1+\kappa(1-\cos\theta)} \equiv E_\mathrm{C}, \tag{2.26}$$

where $\kappa = E/\mathrm{m_e}c^2$, as before. The DCS for Compton scattering by a free electron at rest is given by the familiar Klein–Nishina formula (Heitler, 1954),

$$\frac{\mathrm{d}\sigma_\mathrm{Co}^\mathrm{KN}}{\mathrm{d}\Omega} = \frac{r_\mathrm{e}^2}{2}\left(\frac{E_\mathrm{C}}{E}\right)^2\left(\frac{E_\mathrm{C}}{E}+\frac{E}{E_\mathrm{C}}-\sin^2\theta\right). \tag{2.27}$$

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. Furthermore, transitions of bound electrons are allowed only if the energy transfer $E-E'$ is larger than the ionisation energy U_i of the active subshell (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 subshell 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 subshell 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 subshell can be expressed as (Eq. 21 in Brusa *et al.*, 1996)

$$\frac{\mathrm{d}^2\sigma_{\mathrm{Co},i}}{\mathrm{d}E'\mathrm{d}\Omega} = \frac{r_\mathrm{e}^2}{2}\left(\frac{E_\mathrm{C}}{E}\right)^2\left(\frac{E_\mathrm{C}}{E}+\frac{E}{E_\mathrm{C}}-\sin^2\theta\right)F(p_z)\,J_i(p_z)\,\frac{\mathrm{d}p_z}{\mathrm{d}E'}, \tag{2.28}$$

where r_e is the classical electron radius. E_C is the energy of the Compton line, defined by Eq. (2.26), *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

$$q = \frac{1}{c}\sqrt{E^2 + E'^2 - 2EE'\cos\theta}. \tag{2.29}$$

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 by[3]

$$p_z \equiv -\frac{\mathbf{p}\cdot\mathbf{q}}{q} = \frac{EE'(1-\cos\theta) - \mathrm{m_e}c^2(E-E')}{c^2 q} \tag{2.30}$$

or, equivalently,

$$\frac{p_z}{\mathrm{m_e}c} = \frac{E(E'-E_\mathrm{C})}{E_\mathrm{C}\,cq}. \tag{2.31}$$

Notice that $p_z = 0$ for $E' = E_\mathrm{C}$. In addition,

$$\frac{\mathrm{d}p_z}{\mathrm{d}E'} = \frac{\mathrm{m_e}c}{cq}\left(\frac{E}{E_\mathrm{C}} + \frac{E\cos\theta - E'}{cq}\,\frac{p_z}{\mathrm{m_e}c}\right). \tag{2.32}$$

The function $J_i(p_z)$ in Eq. (2.28) is the one-electron Compton profile of the active subshell, which is defined as

$$J_i(p_z) \equiv \iint \rho_i(\mathbf{p})\,\mathrm{d}p_x\,\mathrm{d}p_y, \tag{2.33}$$

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

$$\int_{-\infty}^{\infty} J_i(p_z)\,\mathrm{d}p_z = 1 \tag{2.34}$$

[3]The expression (2.30) contains an approximation; the exact relation is obtained by replacing the electron rest energy $\mathrm{m_e}c^2$ in the numerator by the electron initial total energy, $\sqrt{(\mathrm{m_e}c^2)^2 + (cp)^2}$.

is assumed. In the Hartree–Fock approximation for closed-subshell configurations, the momentum distribution of the electrons in an atomic subshell, obtained by adding the contributions of the orbitals in that subshell, is isotropic. For an isotropic distribution, expression (2.33) simplifies to

$$J_i(p_z) = 2\pi \int_{|p_z|}^{\infty} p\, \rho_i(p)\, \mathrm{d}p. \tag{2.35}$$

The atomic Compton profile is given by

$$J(p_z) = \sum_i f_i\, J_i(p_z), \tag{2.36}$$

where f_i is the number of electrons in the i-th subshell and $J_i(p_z)$ is the one-electron profile of this subshell. The functions $J(p_z)$ and $J_i(p_z)$ are both bell-shaped and symmetrical about $p_z = 0$ (see Fig. 2.7). 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 subshells. 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 (Cooper, 1985)

$$J_i^{\mathrm{feg}}(p_z) = \frac{3}{4p_\mathrm{F}} \left(1 - \frac{p_z^2}{p_\mathrm{F}^2}\right) \Theta(p_\mathrm{F} - |p_z|), \qquad J_i^{\mathrm{feg}}(0) = \frac{3}{4p_\mathrm{F}}, \tag{2.37}$$

where $p_\mathrm{F} \equiv \hbar(3\pi^2\rho_\mathrm{e})^{1/3}$ 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. (2.28) is approximately given by

$$F(p_z) \simeq 1 + \frac{cq_\mathrm{C}}{E} \left(1 + \frac{E_\mathrm{C}(E_\mathrm{C} - E\cos\theta)}{(cq_\mathrm{C})^2}\right) \frac{p_z}{\mathrm{m_e}c}, \tag{2.38}$$

where q_C is the momentum transfer associated with the energy $E' = E_\mathrm{C}$ of the Compton line,

$$q_\mathrm{C} \equiv \frac{1}{c}\sqrt{E^2 + E_\mathrm{C}^2 - 2EE_\mathrm{C}\cos\theta}. \tag{2.39}$$

Expression (2.38) 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 (2.38) only for $|p_z| < 0.2\,\mathrm{m_e}c$ and take $F(\pm|p_z|) = F(\pm 0.2\,\mathrm{m_e}c)$ for $|p_z| > 0.2\,\mathrm{m_e}c$. 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 $E - E'$

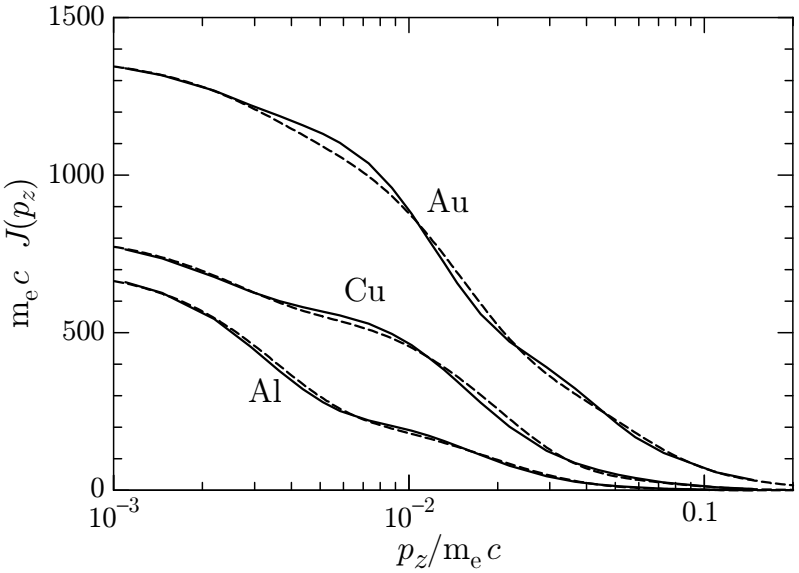

Figure 2.7: Atomic Compton profiles ($p_z > 0$) for aluminium, copper and gold. The continuous curves are numerical Hartree–Fock profiles tabulated by Biggs *et al.* (1975). The dashed curves represent the analytical profiles defined by Eq. (2.54). (Adapted from Brusa *et al.*, 1996).

is larger than the ionisation energy U_i of the active subshell. Therefore the atomic DCS, including Doppler broadening and binding effects, is given by

$$\frac{\mathrm{d}^2\sigma_{\mathrm{Co}}}{\mathrm{d}E'\mathrm{d}\Omega} = \frac{r_\mathrm{e}^2}{2}\left(\frac{E_\mathrm{C}}{E}\right)^2\left(\frac{E_\mathrm{C}}{E}+\frac{E}{E_\mathrm{C}}-\sin^2\theta\right) \times F(p_z)\left(\sum_i f_i\, J_i(p_z)\,\Theta(E-E'-U_i)\right)\frac{\mathrm{d}p_z}{\mathrm{d}E'}, \tag{2.40}$$

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 Carlson (1975), Fig. 2.3. The DCS for scattering of 10 keV photons by aluminium atoms is displayed in Fig. 2.8, 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 $E' = E_\mathrm{C}$; 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). Furthermore, from Eq. (2.31) $E' = E_\mathrm{C}$, so that photons scattered through an angle θ have energy E_C. Integration of the DCS, Eq. (2.40), over E' then yields the Klein–Nishina cross section,

$$\frac{\mathrm{d}\sigma_{\mathrm{Co}}^{\mathrm{KN}}}{\mathrm{d}\Omega} = Z\,\frac{r_\mathrm{e}^2}{2}\left(\frac{E_\mathrm{C}}{E}\right)^2\left(\frac{E_\mathrm{C}}{E}+\frac{E}{E_\mathrm{C}}-\sin^2\theta\right), \tag{2.41}$$

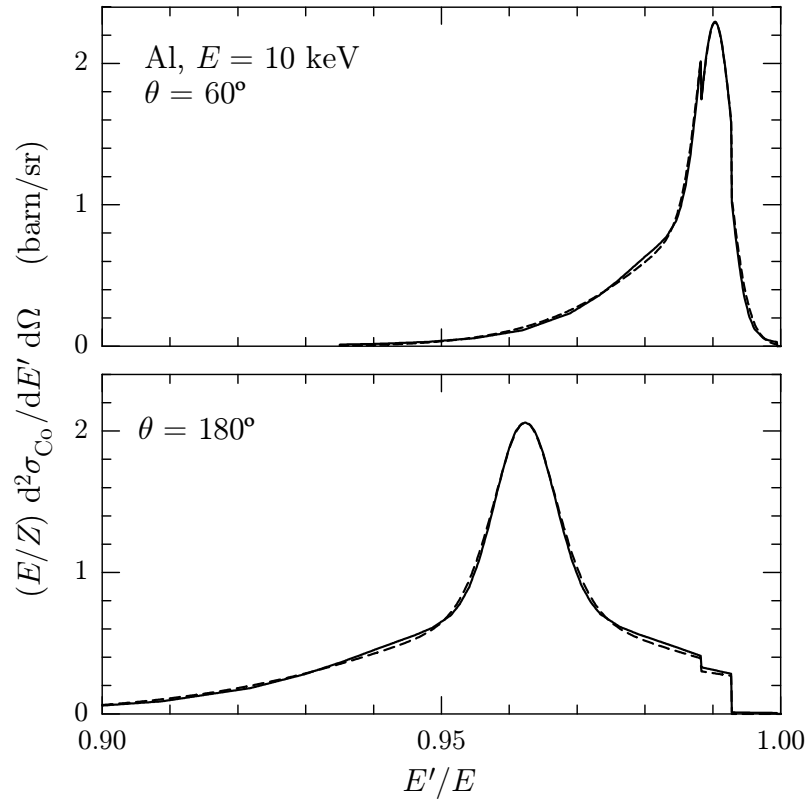

Figure 2.8: DCS for Compton scattering of 10 keV photons by aluminium atoms at the indicated scattering angles. The continuous curves represent the DCS (2.40) calculated using the Hartree–Fock Compton profile (Biggs *et al.*, 1975). The dashed curves are results from Eq. (2.40) with the analytical profiles given by Eq. (2.54). (Adapted from Brusa *et al.*, 1996).

for the Z atomic electrons [cf. Eq. (2.27)]. 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,

$$\frac{\mathrm{d}\sigma_{\mathrm{Co}}}{\mathrm{d}\Omega} = \int \frac{\mathrm{d}^2\sigma_{\mathrm{Co}}}{\mathrm{d}E'\mathrm{d}\Omega}\,\mathrm{d}E' = \frac{r_{\mathrm{e}}^2}{2}\left(\frac{E_{\mathrm{C}}}{E}\right)^2\left(\frac{E_{\mathrm{C}}}{E}+\frac{E}{E_{\mathrm{C}}}-\sin^2\theta\right) \times \sum_i f_i\,\Theta(E-U_i)\int_{-\infty}^{p_{i,\max}} F(p_z)J_i(p_z)\,\mathrm{d}p_z, \tag{2.42}$$

where $p_{i,\max}$ is the highest p_z-value for which an electron in the i-th subshell can be excited. It is obtained from Eq. (2.30) by setting $E' = E - U_i$,

$$p_{i,\max}(E,\theta) = \frac{E(E-U_i)(1-\cos\theta) - \mathrm{m_e}c^2U_i}{c\sqrt{2E(E-U_i)(1-\cos\theta)+U_i^2}}. \tag{2.43}$$

Except for energies just above the subshell 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.*,

$$\int_{-\infty}^{p_{i,\max}} F(p_z) J_i(p_z)\,\mathrm{d}p_z \simeq n_i(p_{i,\max}), \tag{2.44}$$

where

$$n_i(p_z) \equiv \int_{-\infty}^{p_z} J_i(p'_z)\,\mathrm{d}p'_z. \tag{2.45}$$

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 subshell that can be effectively excited in a Compton interaction. We can then write

$$\frac{\mathrm{d}\sigma_{\mathrm{Co}}}{\mathrm{d}\Omega} \simeq \frac{r_\mathrm{e}^2}{2}\left(\frac{E_\mathrm{C}}{E}\right)^2\left(\frac{E_\mathrm{C}}{E} + \frac{E}{E_\mathrm{C}} - \sin^2\theta\right) S(E,\theta). \tag{2.46}$$

The function

$$S(E,\theta) = \sum_i f_i\,\Theta(E - U_i)\,n_i(p_{i,\max}) \tag{2.47}$$

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

$$\sigma_{\mathrm{Co}} \simeq 2\pi \int_{-1}^{1} \frac{\mathrm{d}\sigma_{\mathrm{Co}}}{\mathrm{d}\Omega}\,\mathrm{d}(\cos\theta). \tag{2.48}$$

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,

$$\frac{\mathrm{d}\sigma_{\mathrm{Co}}}{\mathrm{d}E'} \equiv \int \frac{\mathrm{d}^2\sigma_{\mathrm{Co}}}{\mathrm{d}E'\mathrm{d}\Omega}\,\mathrm{d}\Omega. \tag{2.49}$$

In the case of scattering by free electrons at rest, $E' = E_\mathrm{C}$ and the Klein–Nishina formula (2.41) gives the following expression for the energy DCS,

$$\begin{aligned}\frac{\mathrm{d}\sigma_{\mathrm{Co}}^{\mathrm{KN}}}{\mathrm{d}E'} &= 2\pi\,\frac{\mathrm{d}\sigma_{\mathrm{Co}}^{\mathrm{KN}}}{\mathrm{d}\Omega}\,\frac{\mathrm{d}(\cos\theta)}{\mathrm{d}E_\mathrm{C}} \\ &= \frac{\pi r_\mathrm{e}^2}{E}\,\kappa^{-3}\left(\frac{E^2}{E'^2} + \frac{(\kappa^2 - 2\kappa - 2)E}{E'} + (2\kappa + 1) + \frac{\kappa^2 E'}{E}\right).\end{aligned} \tag{2.50}$$

The EPDL (Cullen *et al.*, 1997), the XCOM program (Berger and Hubbell, 1987; Berger *et al.*, 2005) and several Monte Carlo codes describe Compton scattering by means of the theory of Waller and Hartree (1929), which accounts for binding effects but disregards Doppler broadening, *i.e.*, photons scattered through an angle θ are assumed to have energy $E' = E_\mathrm{C}$. This theory leads to the DCS

$$\frac{\mathrm{d}\sigma_{\mathrm{Co}}^{\mathrm{WH}}}{\mathrm{d}E'} = \frac{\mathrm{d}\sigma_{\mathrm{Co}}^{\mathrm{KN}}}{\mathrm{d}E'}\,S_{\mathrm{WH}}(q_\mathrm{C}), \tag{2.51}$$

where q_C is the momentum transfer for the Compton line, given by Eq. (2.39), and $S_{WH}(q_C)$ is the Waller–Hartree incoherent scattering function, which can be calculated in terms of the atomic ground-state wave function. It is a monotonically increasing function of q_C that takes values from $S_{WH}(0) = 0$ to $S_{WH}(\infty) = Z$. Tables of the Waller–Hartree incoherent scattering function for all elements from $Z = 1$ to 100 have been compiled by Hubbell *et al.* (1975, 1977); accurate analytical approximations for $S_{WH}(q_C)$ are given by Baró *et al.* (1994a).

Figure 2.9 displays energy DCSs for Compton scattering by aluminium and gold atoms obtained from the impulse approximation, the Klein–Nishina formula and the Waller–Hartree theory [Eqs. (2.49), (2.50) and (2.51), respectively]. These results clearly show the differences between the physics of the impulse approximation and the cruder Waller–Hartree and free-electron approximations. The Klein–Nishina and Waller–Hartree DCSs have a threshold at

$$E'_{th} = E/(1 + 2\kappa). \tag{2.52}$$

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. 2.4). The Waller–Hartree DCS decreases smoothly when the energy of the scattered photon approaches E, in qualitative agreement with the behaviour of the DCS from the impulse approximation.

The left panel of Fig. 2.10 shows total cross sections for Compton scattering by aluminium and gold atoms as functions of the energy E of the incident photon. While total cross sections obtained from the impulse approximation and from the Waller–Hartree theory tend to zero for small E, the Klein–Nishina total cross section goes to a finite value for $E \to 0$. For energies larger than a few tens of keV, the total cross section obtained from the Waller–Hartree theory does not differ significantly from the result of the impulse approximation, reflecting the fact that the total cross section is quite insensitive to Doppler broadening, *i.e.*, the areas under the impulse approximation and Waller–Hartree curves in Figs. 2.9 are practically the same.

A quantity of interest in dosimetry is the energy-deposition cross section,

$$\sigma_{Co,E} \equiv \int_0 \frac{d\sigma_{Co}}{dE} (E - E') \, dE. \tag{2.53}$$

Energy-deposition cross sections for gold, computed using the impulse approximation, the Waller–Hartree and the Klein–Nishina DCSs, are displayed on the right-hand panel of Fig. 2.10. As pointed out by Ribberfors (1983), values of $\sigma_{Co,E}$ obtained from the Waller–Hartree theory are systematically smaller than those derived from the impulse approximation. Clearly, this is a consequence of Doppler broadening, which makes

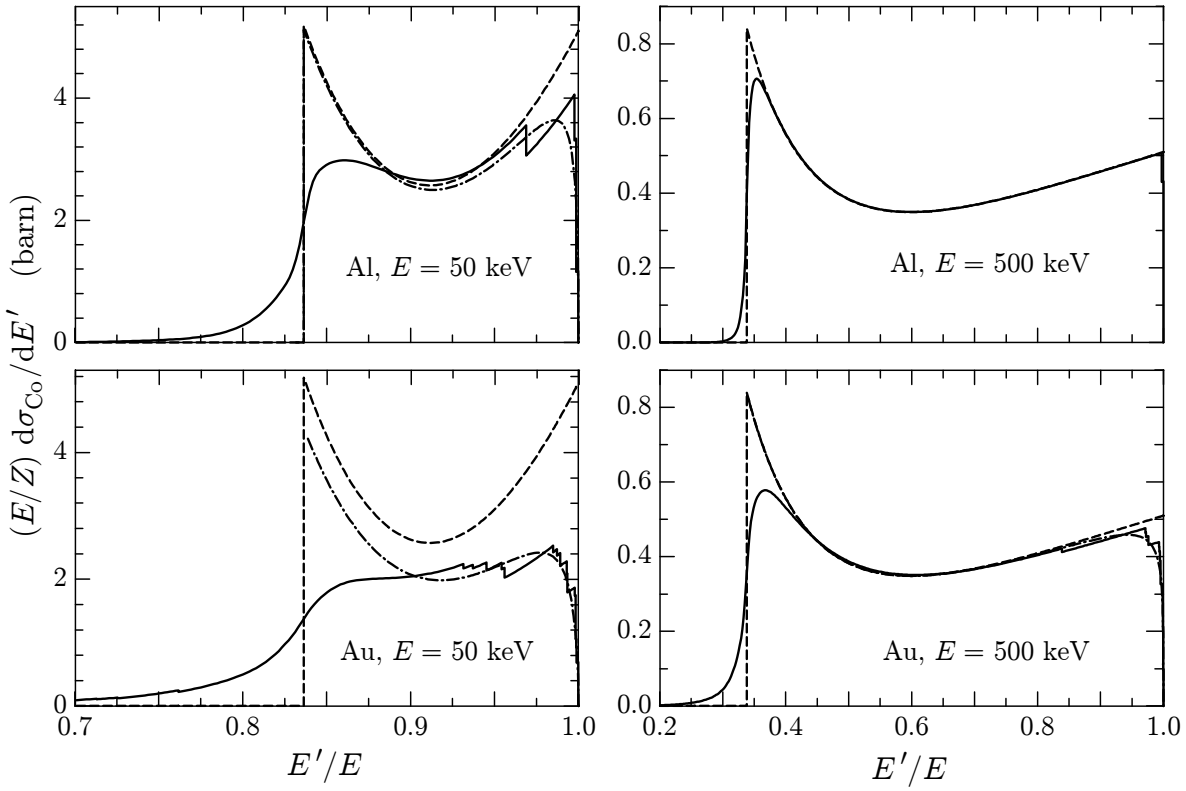

Figure 2.9: Energy DCSs for Compton scattering of 50 and 500 keV photons by aluminium and gold atoms. The continuous curves represent the DCS (2.49), computed using the analytical Compton profiles (2.54). The dashed curves are obtained from the Klein–Nishina formula (2.50), *i.e.*, assuming that the atomic electrons are free and at rest. Dot-dashed curves represent results from the Waller–Hartree theory, Eq. (2.51). (Adapted from Brusa *et al.*, 1996).

possible energy transfers larger than the maximum value, $E - E_{\mathrm{C}}$, allowed in the Waller–Hartree theory (see Figs. 2.9). Interestingly, in the low-energy domain, the simpler Klein–Nishina DCS yields values of the energy-deposition cross section $\sigma_{\mathrm{Co},E}$ that are closer to the result of the impulse approximation.

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

$$J_i^{\mathrm{A}}(p_z) = J_{i,0}\,\frac{n d_2}{2}\left(d_1 + d_2 J_{i,0}\,|p_z|\right)^{n-1} \exp\left[d_1^n - \left(d_1 + d_2 J_{i,0}\,|p_z|\right)^n\right] \tag{2.54}$$

with

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

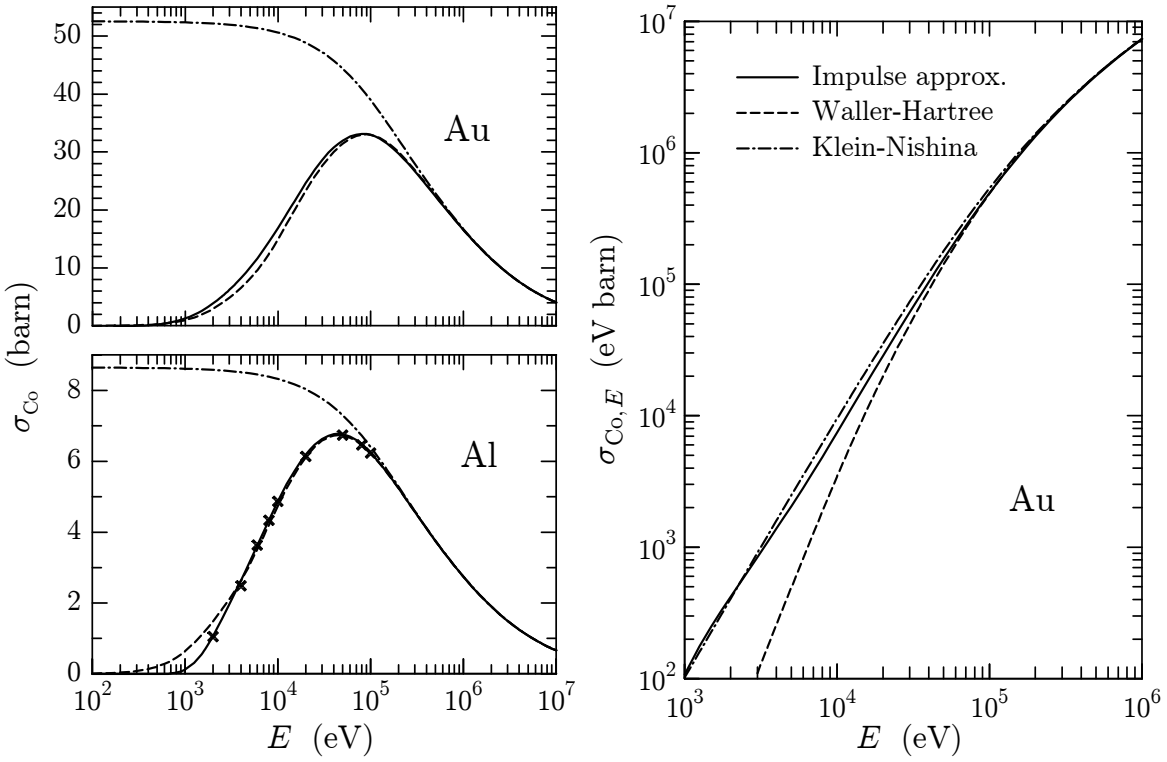

Figure 2.10: Total cross sections for Compton scattering of photons by aluminium and gold atoms as functions of the photon energy E (left panel). Continuous curves are results obtained from the impulse approximation with the analytical profiles (2.54). The dashed and dot-dashed curves represent results from the Waller–Hartree theory, Eq. (2.51), and from the Klein–Nishina formula, Eq. (2.50), respectively. Crosses are total cross sections given by Ribberfors (1983) for aluminium, calculated from the impulse approximation using numerical non-relativistic Hartree–Fock Compton profiles. The right plot shows energy-deposition cross sections, Eq. (2.53), for Compton scattering by gold atoms, calculated from the theoretical models indicated in the legend. (Adapted from Brusa *et al.*, 1996).

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. In previous versions of PENELOPE, this quantity was extracted from the tables of Biggs *et al.* (1975). In the present version of the code we use more accurate $J_i(0)$ values, which were calculated numerically from the radial wave functions given by the MCDF program of Desclaux (1975, 1977). A list of these values for all subshells of the elements $Z = 1$ to 99 is included in the file `pdatconf.p14`. Notice that $J_i^{\rm A}(p_z)$ is normalised according to Eq. (2.34). With the profiles (2.54),

$$n_i^{\rm A}(p_z) \equiv \int_{-\infty}^{p_z} J_i^{\rm A}(p'_z)\,{\rm d}p'_z = \begin{cases} \frac{1}{2}\,\exp\left[d_1^2 - \Big(d_1 - d_2 J_{i,0}\,p_z\Big)^2\right] & \text{if } p_z < 0, \\ 1 - \frac{1}{2}\,\exp\left[d_1^2 - \Big(d_1 + d_2 J_{i,0}\,p_z\Big)^2\right] & \text{if } p_z > 0. \end{cases} \tag{2.55}$$

Thus, the incoherent scattering function (2.47) can be expressed analytically and the integral (2.48) evaluated very quickly with the aid of function `SUMGA` (Section B.2 in Appendix B). On the other hand, the sampling equation $n_i^{\mathrm{A}}(p_z) \equiv \xi n_i^{\mathrm{A}}(p_{i,\max})$ (see Section 1.2.2) can be solved analytically,

$$p_z = \begin{cases} \dfrac{1}{d_2 J_{i,0}} \left[d_1 - \left({d_1}^2 - \ln 2A \right)^{1/2} \right] & \text{if } A < \frac{1}{2}, \\ \dfrac{1}{d_2 J_{i,0}} \left[\left({d_1}^2 - \ln 2(1-A) \right)^{1/2} - d_1 \right] & \text{if } A > \frac{1}{2}, \end{cases} \tag{2.56}$$

where $A \equiv \xi n_i^{\mathrm{A}}(p_{i,\max})$. Atomic Compton profiles obtained from the approximation given by Eq. (2.54) are accurate for small p_z and oscillate about the Hartree–Fock values for intermediate momenta (see Fig. 2.7). 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. 2.8). For most applications (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 (2.54) then provides a conveniently simple method to introduce Doppler broadening and binding effects in the simulation of Compton scattering.

For photons with energies E lower than, and of the order of, the ionisation energy of the K shell, the total cross section obtained from Eq. (2.48) (with $F(p_z) = 1$) may differ by a few percent from the result of integrating the DCS (2.40) numerically. In the simulation code, we use a more accurate calculation scheme in which the contribution of the second term in the expression (2.38) of $F(p_z)$ is estimated by using the Compton profile of a free-electron gas, Eq. (2.37), with a Fermi momentum such that $J_i^{\mathrm{feg}} = J_{i,0}$. This reduces the error of the calculated total atomic cross sections to less than about 1%.

In PENELOPE, the maximum number of electron subshells for each material is limited. For heavy elements, and also for compounds, the number of subshells may be fairly large. In this case, outer subshells with similar ionisation energies are grouped together and replaced by a single subshell with a $J_{i,0}$ value and an effective ionisation energy equal to the corresponding averages of the grouped subshells. 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. (2.40) with the analytical Compton profiles (2.54). 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)

$$P_{\mathrm{Co}}(\cos\theta, E') = \left(\frac{E_{\mathrm{C}}}{E}\right)^2 \left(\frac{E_{\mathrm{C}}}{E} + \frac{E}{E_{\mathrm{C}}} - \sin^2\theta\right) \times F(p_z) \left(\sum_i f_i\, J_i(p_z)\, \Theta(E - E' - U_i)\right) \frac{\mathrm{d}p_z}{\mathrm{d}E'}. \quad (2.57)$$

Integration of expression (2.57) over E', using the approximation (2.44), yields the PDF of the polar deflection

$$P_\theta(\cos\theta) = \left(\frac{E_{\mathrm{C}}}{E}\right)^2 \left(\frac{E_{\mathrm{C}}}{E} + \frac{E}{E_{\mathrm{C}}} - \sin^2\theta\right) S(E,\theta), \quad (2.58)$$

where $S(E,\theta)$ is the incoherent scattering function, Eq. (2.47).

Random values of $\cos\theta$ from the PDF (2.58) can be generated by using the following algorithm (Baró *et al.*, 1994a). Let us introduce the quantity

$$\tau \equiv \frac{E_{\mathrm{C}}}{E} = \frac{1}{1 + \kappa(1 - \cos\theta)}. \quad (2.59)$$

The minimum and maximum values of τ are

$$\tau_{\min} = \frac{1}{1+2\kappa} \qquad \text{and} \qquad \tau_{\max} = 1, \quad (2.60)$$

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

$$P_\tau(\tau) = P_\theta(\cos\theta)\frac{\mathrm{d}(\cos\theta)}{\mathrm{d}\tau} = \left(\frac{1}{\tau^2} + \frac{\kappa^2 - 2\kappa - 2}{\tau} + (2\kappa+1) + \kappa^2\tau\right) S(E,\theta). \quad (2.61)$$

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

$$P_\tau(\tau) = [a_1\, P_1(\tau) + a_2\, P_2(\tau)]\, T(\cos\theta), \quad (2.62)$$

where

$$a_1 = \ln(1+2\kappa), \qquad a_2 = \frac{2\kappa(1+\kappa)}{(1+2\kappa)^2}, \quad (2.63)$$

$$P_1(\tau) = \frac{1}{\ln(1+2\kappa)}\frac{1}{\tau}, \qquad P_2(\tau) = \frac{(1+2\kappa)^2}{2\kappa(1+\kappa)}\tau \quad (2.64)$$

and

$$T(\cos\theta) = \frac{1 + (\kappa^2 - 2\kappa - 2)\tau + (2\kappa+1)\tau^2 + \kappa^2\tau^3}{\kappa^2\tau(1+\tau^2)} \frac{S(E,\theta)}{S(E,\theta=\pi)} = \left\{1 - \frac{(1-\tau)\,[(2\kappa+1)\tau - 1]}{\kappa^2\tau(1+\tau^2)}\right\} \frac{S(E,\theta)}{S(E,\theta=\pi)}. \quad (2.65)$$

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. In addition, 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. (2.61) 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

$$\pi(1) = \frac{a_1}{a_1 + a_2} \qquad \text{and} \qquad \pi(2) = \frac{a_2}{a_1 + a_2}. \tag{2.66}$$

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

$$\tau = \begin{cases} \tau_{\min}^{\xi} & \text{if } i = 1, \\ \left[\tau_{\min}^2 + \xi\,(1 - \tau_{\min}^2)\right]^{1/2} & \text{if } i = 2, \end{cases} \tag{2.67}$$

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

(iii) Determine $\cos\theta$ using Eq. (2.59),

$$\cos\theta = 1 - \frac{1 - \tau}{\kappa\tau}, \tag{2.68}$$

and compute the quantities $p_{i,\max}(E, \theta)$, Eq. (2.43), and

$$S(E, \theta) = \sum_i f_i\,\Theta(E - U_i)\,n_i^{\mathrm{A}}(p_{i,\max}). \tag{2.69}$$

(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 subshell i is selected with relative probability equal to $Z_i\,\Theta(E - U_i)\,n_i^{\mathrm{A}}(p_{i,\max}(E, \theta))$. A random value of p_z is generated from the analytical Compton profile (2.54) using the sampling formula (2.56). If p_z is less than $-\mathrm{m_e}c$, it is rejected and a new subshell and a p_z-value are sampled[4]. Finally, the factor $F(p_z)$ in the PDF (2.40) is accounted for by means of

[4]Notice that, due to the approximation introduced in Eq. (2.38), a value $p_z < -\mathrm{m_e}c$ would yield a negative energy for the scattered photon.

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 (2.38) must be used. Let $F_{\max}$ denote the maximum value of $F(p_z)$, which occurs at $p_z = 0.2\mathrm{m_e}c$ or $-0.2\mathrm{m_e}c$; 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 subshell and a p_z-value is reinitiated. The energy E' of the emerging photon is then calculated from Eq. (2.30), which gives

$$E' = E\,\frac{\tau}{1-t\tau^2}\left[(1-t\tau\cos\theta) + \mathrm{sign}(p_z)\,\sqrt{(1-t\tau\cos\theta)^2 - (1-t\tau^2)(1-t)}\right], \quad (2.70)$$

where

$$t \equiv (p_z/\mathrm{m_e}c)^2 \qquad \text{and} \qquad \mathrm{sign}(p_z) \equiv p_z/|p_z|. \quad (2.71)$$

For photons with energy larger than 5 MeV, for which Doppler broadening is negligible, we set $E' = E_{\mathrm{C}}$ (which amounts to assuming that $p_z = 0$). In this case, the active electron subshell 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 $E_{\mathrm{e}} = E - E' - U_i$ in the direction of the momentum transfer vector $\mathbf{q} = \hbar\mathbf{k} - \hbar\mathbf{k}'$, with polar angle θ_{e} and azimuthal angle $\phi_{\mathrm{e}} = \phi + \pi$, relative to the direction of the incident photon. $\cos\theta_{\mathrm{e}}$ is given by

$$\cos\theta_{\mathrm{e}} = \frac{E - E'\cos\theta}{\sqrt{E^2 + E'^2 - 2EE'\cos\theta}}. \quad (2.72)$$

When $E' = E_{\mathrm{C}}$, this expression simplifies to

$$\cos\theta_{\mathrm{e}} = \frac{E + \mathrm{m_e}c^2}{E}\left(\frac{E - E_{\mathrm{C}}}{2\mathrm{m_e}c^2 + E - E_{\mathrm{C}}}\right)^{1/2}, \quad (2.73)$$

which coincides with the result (A.20). Since the active electron subshell 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

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. 2.12 below).

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 Brusa *et al.* (1996).

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

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 (1934) from the Born approximation (see also Motz *et al.*, 1969; Tsai, 1974, 1977). 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)

$$\frac{d\sigma_{pp}^{(BH)}}{d\epsilon} = r_e^2 \alpha Z[Z+\eta]\left\{\left[\epsilon^2+(1-\epsilon)^2\right](\Phi_1 - 4f_C) + \frac{2}{3}\epsilon(1-\epsilon)(\Phi_2 - 4f_C)\right\}. \quad (2.74)$$

Notice that the "reduced energy" $\epsilon = (E_- + m_e c^2)/E$ is the fraction of the photon energy that is taken away by the electron; the kinetic energy of the positron is $E_+ = E - E_- - 2m_e c^2 = (1-\epsilon)E - m_e c^2$. 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 2.2). 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, 1951, 1968; Tsai, 1974), *i.e.*, the electrostatic potential of the atom is assumed to be (Wentzel model)

$$\varphi_W(r) = \frac{Ze}{r}\exp(-r/R), \quad (2.75)$$

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

$$\rho_{\mathrm{W}}(r) = \frac{1}{4\pi e}\nabla^2\varphi(r) = \frac{1}{4\pi e}\frac{1}{r}\frac{\mathrm{d}^2}{\mathrm{d}r^2}\left[r\varphi(r)\right] = \frac{Z}{4\pi R^2 r}\exp(-r/R), \tag{2.76}$$

and the atomic form factor is

$$F_{\mathrm{W}}(q,Z) = 4\pi\int_0^\infty \rho_{\mathrm{W}}(r)\frac{\sin(qr/\hbar)}{qr/\hbar}r^2\,\mathrm{d}r = \frac{Z}{1+(Rq/\hbar)^2}. \tag{2.77}$$

The screening functions for this particular form factor take the following analytical expressions (Tsai, 1974, 1977)

$$\begin{aligned}\Phi_1 &= 2 - 2\ln(1+b^2) - 4b\arctan(b^{-1}) + 4\ln(R\mathrm{m_e}c/\hbar)\\ \Phi_2 &= \frac{4}{3} - 2\ln(1+b^2) + 2b^2\left[4 - 4b\arctan(b^{-1}) - 3\ln(1+b^{-2})\right]\\ &\quad + 4\ln(R\mathrm{m_e}c/\hbar),\end{aligned} \tag{2.78}$$

where

$$b = \frac{R\mathrm{m_e}c}{\hbar}\frac{1}{2\kappa}\frac{1}{\epsilon(1-\epsilon)}. \tag{2.79}$$

The quantity η in Eq. (2.74) 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 reduced energy, ϵ, is assumed to be the same as that of the pair cross section. The function f_{C} in (2.74) is the high-energy Coulomb correction of Davies *et al.* (1954) given by

$$\begin{aligned}f_{\mathrm{C}}(Z) &= a^2\left[(1+a^2)^{-1} + 0.202059 - 0.03693a^2 + 0.00835a^4\right.\\ &\quad \left. - 0.00201a^6 + 0.00049a^8 - 0.00012a^{10} + 0.00003a^{12}\right],\end{aligned} \tag{2.80}$$

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

$$\sigma_{\mathrm{pp}}^{(\mathrm{BH})} = \int_{\epsilon_{\min}}^{\epsilon_{\max}} \frac{\mathrm{d}\sigma_{\mathrm{pp}}^{(\mathrm{BH})}}{\mathrm{d}\epsilon}\,\mathrm{d}\epsilon, \tag{2.81}$$

where

$$\epsilon_{\min} = \mathrm{m_e}c^2/E = \kappa^{-1} \qquad \text{and} \qquad \epsilon_{\max} = 1 - \mathrm{m_e}c^2/E = 1 - \kappa^{-1}. \tag{2.82}$$

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. (2.74) with $\eta = 0$ exactly reproduces the

Table 2.2: Reduced screening radius, $R\mathrm{m_e}c/\hbar$, 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 $\hbar/\mathrm{m_e}c = 3.8616 \times 10^{-13}$ m is the Compton wavelength of the electron (Baró *et al.*, 1994a).

Z	$R\mathrm{m_e}c/\hbar$	η_∞	Z	$R\mathrm{m_e}c/\hbar$	η_∞	Z	$R\mathrm{m_e}c/\hbar$	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

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 $E \simeq 4\mathrm{m_e}c^2$ and reaches a saturation value, η_∞, at high energies. It can be obtained, for all elements and energies up to 10^5 MeV, as

$$\eta(E) = Z\,\sigma^{\mathrm{HGO}}_{\mathrm{triplet}}(E)/\sigma^{\mathrm{HGO}}_{\mathrm{pair}}(E), \tag{2.83}$$

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

$$\eta_\infty \simeq Z\,\sigma^{\mathrm{HGO}}_{\mathrm{triplet}}(10^5\,\mathrm{MeV})/\sigma^{\mathrm{HGO}}_{\mathrm{pair}}(10^5\,\mathrm{MeV}). \tag{2.84}$$

The values of η_∞ for the elements Z = 1–99 are given in Table 2.2.

The approximation given by Eq. (2.74) 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 random sampling, the Bethe-Heitler DCS, Eq. (2.74), including this low-energy correction and a high-energy radiative correction, is written in the form

$$\frac{\mathrm{d}\sigma_{\mathrm{pp}}}{\mathrm{d}\epsilon} = r_\mathrm{e}^2\alpha Z[Z+\eta]\,C_\mathrm{r}\,\frac{2}{3}\left[2\left(\frac{1}{2}-\epsilon\right)^2\phi_1(\epsilon)+\phi_2(\epsilon)\right], \tag{2.85}$$

where

$$\begin{aligned}\phi_1(\epsilon) &= g_1(b) + g_0(\kappa),\\ \phi_2(\epsilon) &= g_2(b) + g_0(\kappa),\end{aligned} \tag{2.86}$$

with

$$\begin{aligned}g_1(b) &= \frac{1}{2}(3\Phi_1-\Phi_2) - 4\ln(R\mathrm{m_e}c/\hbar) = \frac{7}{3} - 2\ln(1+b^2) - 6b\arctan(b^{-1})\\ &\quad - b^2\left[4 - 4b\arctan(b^{-1}) - 3\ln(1+b^{-2})\right],\\ g_2(b) &= \frac{1}{4}(3\Phi_1+\Phi_2) - 4\ln(R\mathrm{m_e}c/\hbar) = \frac{11}{6} - 2\ln(1+b^2) - 3b\arctan(b^{-1})\\ &\quad + \frac{1}{2}b^2\left[4 - 4b\arctan(b^{-1}) - 3\ln(1+b^{-2})\right],\\ g_0(\kappa) &= 4\ln(R\mathrm{m_e}c/\hbar) - 4f_\mathrm{C}(Z) + F_0(\kappa, Z).\end{aligned} \tag{2.87}$$

$C_\mathrm{r} = 1.0093$ 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. (2.85) (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

$$\begin{aligned} F_0(\kappa, Z) \;=\; & (-1.774 - 12.10a + 11.18a^2)(2/\kappa)^{1/2} \\ & + (8.523 + 73.26a - 44.41a^2)(2/\kappa) \\ & - (13.52 + 121.1a - 96.41a^2)(2/\kappa)^{3/2} \\ & + (8.946 + 62.05a - 63.41a^2)(2/\kappa)^2. \end{aligned} \tag{2.88}$$

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. (2.82), 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. (2.85) and the total cross sections tabulated by Hubbell *et al.* (1980) are appreciable near the threshold [actually, (2.85) 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, in PENELOPE we use a database of atomic cross sections for pair and triplet production, $\sigma_{\rm pair}(E)$ and $\sigma_{\rm triplet}(E)$, generated with the XCOM program (Berger *et al.*, 2005) for a dense grid of photon energies. The total cross section,

$$\sigma_{\rm pp}(E) = \sigma_{\rm pair}(E) + \sigma_{\rm triplet}(E)\,, \tag{2.89}$$

and the probability of triplet production, $p_{\rm triplet} = \sigma_{\rm triplet}/\sigma_{\rm pp}$, are obtained by interpolation. 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 reduced energy ϵ, is symmetrical about $\epsilon = 1/2$ (see Fig. 2.11). This dependence on ϵ is reasonably accurate only for photon energies larger than $\sim$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 (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. (2.85), for energies down to the threshold.

2.4.1 Simulation of pair-production events

The Bethe-Heitler DCS, Eq. (2.85), only depends on the kinetic energy $E_- = \epsilon E - \mathrm{m_e}c^2$ of the produced electron, so that E_- can be directly sampled from Eq. (2.85); the kinetic energy of the positron is obtained as $E_+ = E - E_- - 2\mathrm{m_e}c^2$. 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 reduced energy ϵ is distributed in the interval $(\kappa^{-1},1-\kappa^{-1})$, see Eq. (2.82), according to the PDF given by Eq. (2.85) (normalisation is again irrelevant)

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

which is symmetrical about the point $\epsilon = 1/2$. Figure 2.11 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 $\epsilon = 1/2$.

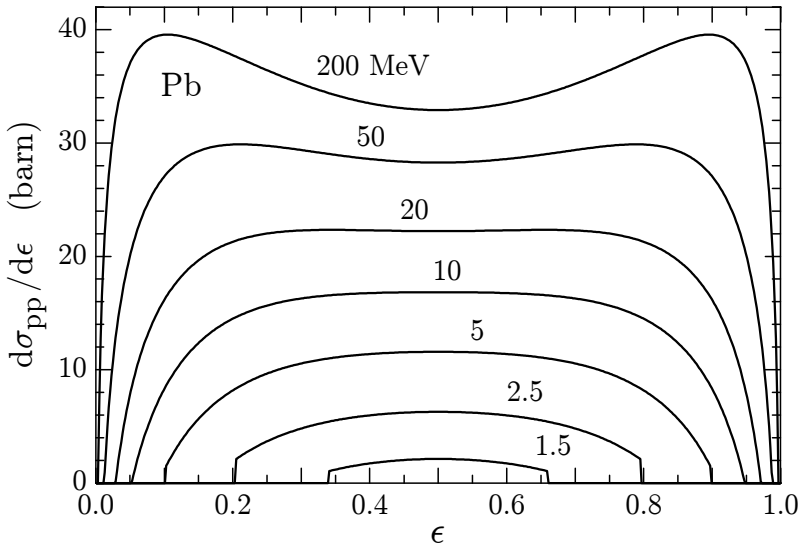

Figure 2.11: DCS for pair-production in lead as a function of the electron reduced energy, $\epsilon = (E_- + \mathrm{m_e}c^2)/E$. (Adapted from Baró *et al.*, 1994a).

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

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

with

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

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

and

$$U_1(\epsilon) = \phi_1(\epsilon)/\phi_1(1/2), \qquad U_2(\epsilon) = \phi_2(\epsilon)/\phi_2(1/2). \tag{2.94}$$

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

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

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

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

(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.

Triplet production occurs with probability, $p_{\text{triplet}} = \sigma_{\text{triplet}}/\sigma_{\text{pp}}$. Although we do not follow the recoiling target electron, we consider that triplet production may leave the target atom with a vacancy in an inner subshell. The active-electron subshell is selected randomly, by assuming that the Z atomic electrons have equal probabilities of being ejected.

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)

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

where a is a normalisation constant and

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

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

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

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. Furthermore, 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 $\mathrm{X_xY_y}$, in which the molecules consist of x atoms of element X and y atoms of element Y. The number of electrons per molecule is $Z_\mathrm{M} = \mathrm{x}Z(\mathrm{X})+\mathrm{y}Z(\mathrm{Y})$ and the molar mass is $A_\mathrm{M} = \mathrm{x}A_\mathrm{w}(\mathrm{X}) + \mathrm{y}A_\mathrm{w}(\mathrm{Y})$, where $Z(\mathrm{X})$ and $A_\mathrm{w}(\mathrm{X})$ stand for the atomic number and atomic weight 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 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 $Z_{\rm eq}$ and molar mass $A_{\rm eq}$ given by

$$Z_{\rm eq}A_{\rm M} = Z_{\rm M}A_{\rm eq} = {\rm x}Z({\rm X})A_{\rm w}({\rm X}) + {\rm y}Z({\rm Y})A_{\rm w}({\rm Y}), \tag{2.100}$$

i.e., its atomic number (molar mass) is the mass-average (Z-average) of the atomic numbers (atomic weights) of the constituent atoms. The reduced energy is sampled from the DCS of the element with the atomic number closest to $Z_{\rm eq}$. 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

$$\mu_{\rm ph} = \mathcal{N}\sigma_{\rm ph}, \tag{2.101}$$

where $\mathcal{N} = N_{\rm A}\rho/A_{\rm M}$ is the number of atoms or molecules per unit volume and $\sigma_{\rm ph}$ is the atomic or molecular photoelectric cross section. The photoelectric mass attenuation coefficient is defined as $\mu_{\rm ph}/\rho$ 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

$$\frac{\mu}{\rho} = \frac{N_{\rm A}}{A_{\rm M}}\left(\sigma_{\rm Ra} + \sigma_{\rm Co} + \sigma_{\rm ph} + \sigma_{\rm pp}\right). \tag{2.102}$$

As mentioned above, PENELOPE uses tables of total cross sections for photoelectric absorption calculated with the program PHOTACS (Sabbatucci and Salvat, 2016), and total cross sections for Rayleigh scattering extracted from the EPDL (Cullen *et al.*, 1997). Cross sections of these two processes for energies different from those in the tables are calculated by linear log-log interpolation. Total cross sections for pair production are obtained from tables generated with the program XCOM (Berger *et al.*, 2005). Values of $\sigma_{\rm pair}(E)$ and $\sigma_{\rm triplet}(E)$ for arbitrary energies are evaluated by cubic spline log-log interpolation of the functions $(1-2{\rm m_e}c^2/E)^{-3}\sigma_{\rm pair}(E)$ and $(1-4{\rm m_e}c^2/E)^{-3}\sigma_{\rm triplet}(E)$, which vary slowly with the photon energy.

Mean free paths for incoherent scattering are calculated from the DCSs given by the impulse approximation, as described in Section 2.3. For E greater than ~ 50 keV, the resulting values are very close to those given by the XCOM program, which were calculated from the Waller–Hartree theory, Eq. (2.51), with the incoherent scattering functions of Hubbell *et al.* (1975, 1977). At lower energies, our mean free paths for Compton scattering deviate from those given by XCOM, because the Waller–Hartree theory neglects Doppler broadening. The evaluation of the total atomic cross section for Compton scattering [see Eq. (2.48)] involves a numerical quadrature, which is performed by using the function `SUMGA` (SectionB.2 in Appendix B). Partial and total mass attenuation coefficients for water, aluminium, iodine and lead are displayed in Fig. 2.12.

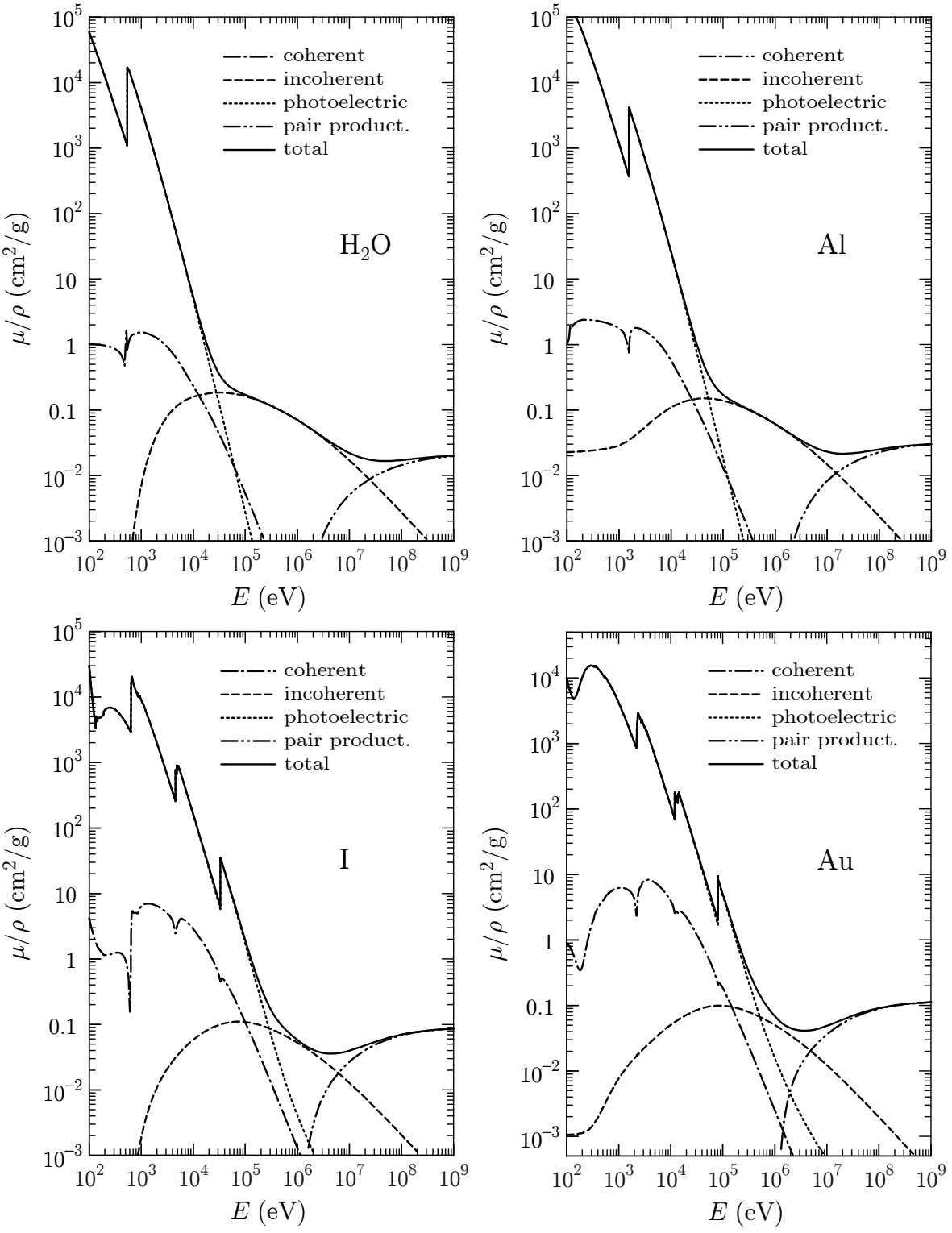

Figure 2.12: Partial and total mass attenuation coefficients of water, aluminium, iodine and gold as functions of the photon energy. Notice the different low-E behaviour of $\mu_{\rm Co}/\rho$ for insulators (water and iodine) and conductors (aluminium and gold). (Adapted from Salvat and Fernández-Varea, 2009).

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 ($i = 1, \ldots, 200$), which spans the complete energy range considered in the simulation, and values of these quantities at intermediate energies E are obtained by linear log-log interpolation. This scheme works well for pair production and for Compton scattering, whose partial attenuation coefficients vary smoothly with energy. However, it is not applicable to photoelectric absorption and Rayleigh scattering because $\mu_{\rm ph}(E)$ is not continuous and $\mu_{\rm Ra}(E)$ varies very rapidly in the vicinity of absorption edges (see Fig. 2.12). To ensure accuracy, the grid of energies where $\mu_{\rm ph}(E)$ is tabulated must include the absorption edges and, therefore, cannot be the same for all materials. On the other hand, $\mu_{\rm Ra}(E)$ must be tabulated densely in the regions where anomalous scattering occurs. Consequently, interpolation of the attenuation coefficients for photoelectric absorption and Rayleigh scattering is lengthier than for other interactions.

Many interpolations of $\mu_{\rm ph}(E)$ and $\mu_{\rm Ra}(E)$ can be avoided by using the following simple method, which involves delta interactions (see Section 1.4.5). Let $\mu_{\rm ph,max}$ be the maximum value of $\mu_{\rm ph}(E)$ in the grid interval $(E_i,\ E_{i+1})$. Transported photons are allowed to undergo delta interactions, and we consider photoelectric absorption and delta interactions as a combined process ("ph+δ") with partial attenuation coefficient equal to $\mu_{\rm ph,max}$, which is constant within each grid interval. That is, the attenuation coefficient for delta interactions is defined by

$$\mu_{{\rm ph},\delta}(E) = \mu_{\rm ph,max} - \mu_{\rm ph}(E). \tag{2.103}$$

Similarly, Rayleigh scattering is combined with delta interactions, and the attenuation coefficient of the latter is set equal to

$$\mu_{{\rm Ra},\delta}(E) = \mu_{\rm Ra,max} - \mu_{\rm Ra}(E), \tag{2.104}$$

where $\mu_{\rm Ra,max}$ is the maximum value of $\mu_{\rm Ra}(E)$ in the grid interval $(E_i,\ E_{i+1})$. To sample the length of the free flight to the next interaction, we use the augmented total attenuation coefficient

$$\mu' = \mu_{\rm ph,max} + \mu_{\rm Ra,max} + \mu_{\rm Co} + \mu_{\rm pp}. \tag{2.105}$$

Note that interpolation of $\mu_{\rm Ra}$ and $\mu_{\rm ph}$ 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+δ", "ph+δ", "Co" and "pp"), which are proportional to the corresponding partial attenuation coefficients,

$$\begin{aligned} p_{\rm Ra+\delta} &= \frac{\mu_{\rm Ra,max}}{\mu'} = \frac{\mu_{{\rm Ra},\delta} + \mu_{\rm Ra}}{\mu'}, \qquad p_{\rm ph+\delta} = \frac{\mu_{\rm ph,max}}{\mu'} = \frac{\mu_{{\rm ph},\delta} + \mu_{\rm ph}}{\mu'} \\ p_{\rm Co} &= \frac{\mu_{\rm Co}}{\mu'}, \qquad p_{\rm pp} = \frac{\mu_{\rm pp}}{\mu'}. \end{aligned} \tag{2.106}$$

If the result of the sampling is, *e.g.*, "ph+δ", the interaction may be either true photoelectric absorption or a delta interaction. The kind of interaction that takes place is determined by random sampling from the corresponding point probabilities $\pi_{\rm ph} = \mu_{\rm ph}(E)/\mu_{\rm ph,max}$ and $\pi_\delta = 1 - \pi_{\rm ph}$. Thus, only when the outcome is "ph+δ" ("Ra+δ"), do we need to evaluate the partial attenuation coefficient for photoelectric absorption (Rayleigh scattering) to determine whether the interaction is photoelectric absorption (scattering) or a delta interaction. The costly interpolation of $\mu_{\rm ph}(E)$ [or $\mu_{\rm Ra}(E)$] is performed only when the probability of photoabsorption [or Rayleigh scattering] 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 single atom. 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 subshell is ionised is nearly proportional to the number of electrons in the subshell. 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 subshells, 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 subshell 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 subshell 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 subshell that is further out (Auger effect). Each non-radiative transition generates an additional vacancy that, in turn, migrates "outwards". The production of vacancies in inner subshells 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 can simulate the emission of characteristic x rays and Auger electrons that result from vacancies produced in K, L, M and N subshells by photoelectric absorption, Compton scattering, triplet production, and electron/positron impact (see Chapter 3). In practical simulations, we introduce a suitable cutoff energy, $E_{\rm c}$ [see Eq. (2.112) below], and consider the subshells with ionisation energies U_i larger than $E_{\rm c}$ as inner subshells. Shells with ionisation energies less than $E_{\rm c}$, or beyond the N7 subshell, are regarded as outer subshells. Although the partial cross sections and relaxation data of subshells N6 and N7 are included in the database and considered in the calculations, the results for these two subshells may not be reliable when their ionisation energies alternate with

those of the O1-5 subshells (see Fig. 2.3). The relaxation of vacancies in ionized atoms is followed until all inner subshells are filled up, *i.e.*, until the vacancies have migrated to outer subshells. Vacancies in these outer subshells originate secondary radiation with energy lower than $E_{\rm c}$, whose main effect is to spread out the excitation energy of the ion in the surrounding material. To obtain a reliable description of the spatial distribution of absorbed dose and other macroscopic transport characteristics, we only have to follow energetic 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 ionisation processes of outer subshells (*i.e.*, photoelectric absorption, Compton scattering, and electron/positron impact), we simply assume that, when ionisation occurs in an outer subshell, a secondary (delta) electron is emitted from the parent ion with a kinetic energy $E_{\rm s}$ equal to the energy deposited by the primary particle,

$$E_{\rm dep} = \begin{cases} E - E' & \text{in Compton scattering,} \\ E & \text{in photoelectric absorption,} \\ W & \text{in electron/positron impact (see Chapter 3).} \end{cases} \tag{2.107}$$

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 values given by Eq. (2.107) 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 farther 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 subshell i (*i.e.*, a subshell with $U_i > E_{\rm c}$), we consider that the electron is ejected with kinetic energy

$$E_{\rm s} = E_{\rm dep} - U_i, \tag{2.108}$$

and that the target atom is left with a vacancy in subshell i. As mentioned above, we consider only characteristic x rays and Auger electrons emitted in the first stages of the relaxation process, until all inner subshells are filled up. This secondary radiation is 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 subshell fills the vacancy in the S0 subshell, leaving a hole in the S1 subshell). The considered radiative transitions (for elements with $Z > 18$ with the M-shell filled) are shown in Fig. 2.2.

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

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, M or N subshell. These data are entered through the material definition file. The transition probabilities are extracted from the LLNL Evaluated Atomic Data Library (EADL; Perkins *et al.*, 1991). Figure 2.13 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 all non-radiative transitions. We see that for low-Z elements, 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 (Bransden and Joachain, 1983); radiative transitions K-S1 are strictly forbidden (to first order) within the dipole approximation.

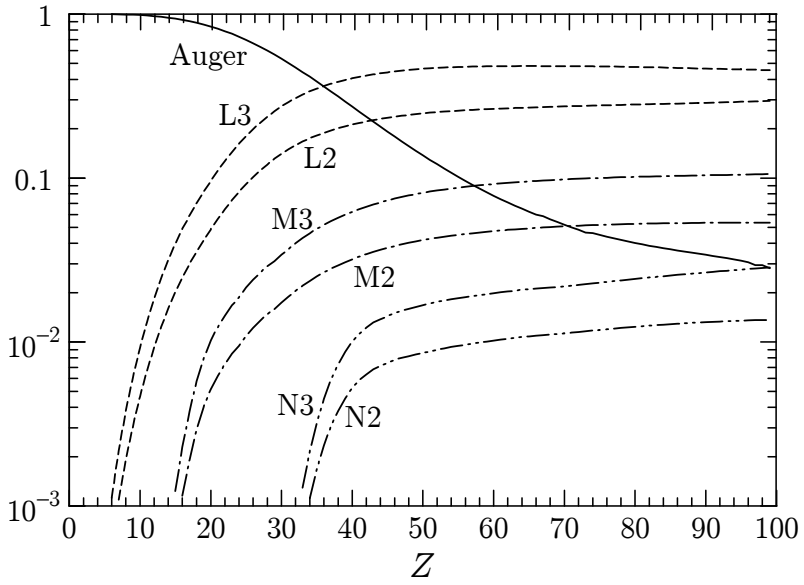

Figure 2.13: Relative probabilities for radiative and non-radiative (Auger) transitions that fill a vacancy in the K-shell of atoms.

The energies of x rays emitted in radiative transitions resulting from single vacancies in the K-shell and L subshells were taken from the 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

$$E_{\rm x} = U'_{\rm S0} - U'_{\rm S1}, \tag{2.109}$$

where $U'_{\mathrm{S}i}$ is the ionisation energy of an electron in the subshell Si of the neutral atom, which was taken from the LLNL Evaluated Atomic Data Library (Perkins *et al.*, 1991). The ionisation energies in this library are the negatives of the eigenvalues of the DHFS self-consistent equations for neutral atoms in their ground states. For tightly bound subshells, with $U'_{\mathrm{S}_i} \gtrsim 200$ eV, the theoretical values $U'_{\mathrm{S}i}$ agree closely with Carlson's (1975) recommended subshell ionisation energies (Salvat and Fernández-Varea, 2009). The energy of the electron emitted in the non-radiative transition S0-S1-S2 is set equal to

$$E_{\mathrm{e}} = U'_{\mathrm{S0}} - U'_{\mathrm{S1}} - U'_{\mathrm{S2}}. \tag{2.110}$$

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 subshells, which is an approximation. It should be mentioned 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 possible relaxation of the ion (see 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% (Hubbell, 1989; Perkins *et al.*, 1991).

The simulation of the relaxation cascade is performed by subroutine `RELAX`, which starts by considering that there is a single vacancy in a subshell Sa. 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 the remaining vacancies. When the decaying ion has vacancies in several different subshells, we assume that those in the innermost subshell migrate first, *i.e.*, we give preference to transitions that fill the innermost open subshell more rapidly. This assumption determines the active subshell Sa of the next transition, and the transition that effectively occurs is sampled randomly from the adopted transition probabilities. The simulation process ends either when the K-shell, and the L, M and N subshells have been filled up or when there is not enough energy to produce "effective" fluorescence radiation (with energy larger than the cutoff energy, E_{c}). The excitation energy of the residual ion is assumed to be deposited locally.

Unfortunately, the EADL gives transition probabilities only for ions with a single vacancy. Because calculations for multiply-ionized atoms are not generally available, we adopt the following simple strategy to correct partially for the possible occurrence of multiple vacancies during the de-excitation cascade. As each vacancy in the active sub-

shell S0 is explicitly filled by subroutine `RELAX` through a S0-S1 or S0-S1-S2 transition, we only need to account for vacancies existing in the intermediate and final subshells, S1 and S2. We assume that transition probabilities are essentially proportional to the number of electrons in these subshells. Let $v_{\mathrm{S}k}$ denote the number of vacancies in subshell Sk before the transition. The probabilities of non-radiative and radiative transitions filling a S0 vacancy of an ion with vacancies in the S1 or S2 subshells are approximated as

$$P^{\mathrm{ion}}_{\mathrm{S0\text{-}S1}} = P_{\mathrm{S0\text{-}S1}} \left[\frac{q_{\mathrm{S1}} - v_{\mathrm{S1}}}{q_{\mathrm{S1}}} \right] \tag{2.111a}$$

and

$$P^{\mathrm{ion}}_{\mathrm{S0\text{-}S1\text{-}S2}} = P_{\mathrm{S0\text{-}S1\text{-}S2}} \left[\frac{q_{\mathrm{S1}} - v_{\mathrm{S1}}}{q_{\mathrm{S1}}} \, \frac{q_{\mathrm{S2}} - v_{\mathrm{S2}}}{q_{\mathrm{S2}}} \right] , \tag{2.111b}$$

respectively. The factors in square brackets are less than unity and, hence, they can be accounted for by a rejection test: the transition is accepted only if a random number ξ is less than the corresponding factor, otherwise a new transition filling the vacancy in the S0 subshell is sampled from the EADL probabilities.

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. As indicated above, the database of interaction cross sections covers the energy range above ~ 50 eV and, consequently, only radiation with energy higher than this value can be followed. Additionally, we do not consider the relaxation of subshells with principal quantum number larger than 4 (O, P, and Q subshells). Hence, simulation results are expected to be reliable only for photons with energies higher than

$$E_{\mathrm{c}} = \max\{50 \text{ eV}, U_{\mathrm{max,out}}(Z_{\mathrm{m}})\} , \tag{2.112}$$

where $U_{\mathrm{max,out}}(Z_{\mathrm{m}})$ denotes the largest ionisation energy of the outer (*i.e.*, O, P, or Q) subshells of the heaviest element present, whose atomic number is Z_{m}. The value of E_{c} is 92 eV for barium, 114 eV for gold, and 329 eV for uranium.

2.6.1 Photon propagation time and atomic lifetimes

In experimental arrangements involving fast detection devices and/or coincidence techniques, the results of measurements may be affected by the time evolution of radiation showers. Conventional Monte Carlo simulations assume that possible interactions between transported particles are negligible and, consequently, their results correspond to a solution of the *linear* Boltzmann transport equation (Berger, 1963). This assumption implies that a shower is not affected by previously simulated showers. The time dependence of a shower can then be readily described by attaching to each particle a state variable that measures the time elapsed since the emission of the primary particle that originated the shower. For brevity, we will refer to this time variable as the *age* of the particle.

The flight time of a photon between two events separated a distance s is s/v, where v is the speed of electromagnetic radiation in the material. The ratio $n = c/v$ defines the refractive index of the medium, which is a function of the radiation frequency. For most materials, and for photons with energies larger than about 100 eV, the refractive index does not differ appreciably from unity (Palik, 1985, 1991, 1998). Therefore, in PENELOPE we consider that photons propagate with the speed of light in vacuum, $v = c$.

While scattering and ionization events occur instantaneously, the emission of x rays (and Auger electrons) after an ionisation event is delayed because of the finite lifetime of atomic energy levels (Sevier, 1972). The natural width Γ of an atomic energy level and the lifetime τ of the states of that level are related by the time-energy uncertainty relation,

$$\Gamma\tau \simeq \hbar. \tag{2.113}$$

As noted by Richtmyer *et al.* (1934), in the case of excitations which produce a single vacancy in an inner subshell i, the subsequent decay of the excited state involves the filling of that vacancy by electrons from nearest subshells, a process practically independent of the condition of the excited electron. Consequently, all excited levels with a vacancy in subshell i have approximately the same level width, Γ_i, the so-called *core-level width*, and the same mean lifetime $\tau_i = \hbar/\Gamma_i$. Calculated values of core-level widths of free atoms are given in the LLNL Evaluated Atomic Data Library (Perkins *et al.*, 1991). Campbell and Papp (2001) provide a set of recommended widths for K to N7 levels of atoms obtained from consideration of available experimental data. These corel-level widths are tabulated in the file `pdatconf.p14` of the PENELOPE database. Typically, the widths Γ_i increase with the binding energy of the subshell; they are of the order of 0.1 eV or less for weakly bound subshells, and reach values of the order of 100 eV for K shells of transuranic elements. The corresponding lifetimes have values of the order of 10^{-14} to 10^{-17} seconds.

Let us consider an atom with a vacancy in subshell i at $t = 0$. The probability of decay per unit time is constant and equal to τ_i^{-1} (Merzbacher, 1970), the reciprocal of the lifetime. That is, the probability that the filling of the vacancy occurs during the time interval between t and $t + \mathrm{d}t$ is $\tau^{-1}\mathrm{d}t$. It may be observed that atomic decay bears a similarity to the interaction process studied in Section 1.4, which is characterised by a constant interaction probability λ^{-1} per unit path length. The PDF $p(t)$ of the time t when the atom decays can be obtained as follows. By definition, the probability that the atom remains in the excited state at time t is

$$\mathcal{F}(t) = \int_t^\infty p(t')\,\mathrm{d}t'. \tag{2.114}$$

The probability that the vacancy will be filled in the time interval between t and $t + \mathrm{d}t$ can be expressed as $\mathcal{F}(t)\,\tau_i^{-1}\mathrm{d}t$. Hence, $p(t)$ satisfies the integral equation

$$p(t) = \tau_i^{-1} \int_t^\infty p(t')\,\mathrm{d}t', \tag{2.115}$$

with the boundary condition $p(\infty) = 0$. The solution is

$$p(t) = \tau_i^{-1}\,\exp(-t\,\tau_i^{-1}). \tag{2.116}$$

Evidently, τ_i is the expectation value of the vacancy lifetime.

Optionally, PENELOPE can follow the time evolution of showers by activating the particle age counter, which must be initialised in the main program and increased after each free flight. Subroutine `RELAX` accounts for the finite lifetime of atomic states by assuming that a vacancy in subshell i is filled after a random time t, which is sampled from the PDF (2.116),

$$t = -\tau_i \ln \xi \,. \tag{2.117}$$

The lifetimes τ_i are obtained from the level widths in the tables of Campbell and Papp (2001), which include data for elements with atomic number Z from 10 to 92, except for a few N subshells. For elements with $Z > 92$, and for N subshells that are missing in the tables of Campbell and Papp, the level widths are extracted from the LLNL Evaluated Atomic Data Library (Perkins *et al.*, 1991). It is worth mentioning that we simulate relaxation cascades using atomic lifetimes that correspond to excited states with a single vacancy. To account partially for the possible occurrence of multiple vacancies, when in the course of the de-excitation cascade we find $\nu > 1$ vacancies in the active subshell, the lifetime is set equal to τ_i/ν.

2.7 Scattering of polarised photons

The theory of scattering of polarised photons is treated in almost all textbooks on quantum mechanics and quantum electrodynamics. Nevertheless, elementary treatments found in textbooks do not suffice to formulate a consistent algorithm for Monte Carlo simulation of photons with *arbitrary* polarisations. In our simulation code we adopt the formal method described by Fano (1954a) and Fano *et al.* (1959), with polarisation states represented by means of the Stokes parameters (see Appendix C). It should be noted that these parameters are defined with respect to a given basis of linear-polarisation states which, in turn, depends on the direction of the photon wave vector $\mathbf{k}$. With the conventions adopted in Appendix C, the simulation algorithm takes a very simple form.

We start by considering the general case of scattering of photons in the mode $\mathbf{k}\boldsymbol{\zeta}$ (*i.e.*, photons that propagate in the direction $\hat{\mathbf{k}}$, with energy $E = \hbar\omega = c\hbar k$ and in the pure polarisation state $\boldsymbol{\zeta}$) by an atom in the energy level E_a; we assume that the target atom is "unpolarised" (*i.e.*, that all degenerate states Ψ_a of the level E_a are equally populated) and that final degenerate atomic states are not observed. That is, the DCS is obtained by summing over degenerate final states and averaging over initial states. The cross section for the transition to a final state with the atom in the energy level E_b and with the scattered photon in the mode $\mathbf{k}'\boldsymbol{\zeta}'$ is given by the Kramers-Heisenberg formula (see Sakurai, 1967), which can be expressed as

$$\frac{\mathrm{d}^2\sigma(b\boldsymbol{\zeta}', a\boldsymbol{\zeta})}{\mathrm{d}E'\,\mathrm{d}\Omega} = r_\mathrm{e}^2\,\frac{E}{E'}\,\left|\mathcal{J}(b,\mathbf{k}'\boldsymbol{\zeta}';a,\mathbf{k}\boldsymbol{\zeta})\right|^2 , \tag{2.118}$$

where $E' = \hbar\omega' = c\hbar k'$ is the energy of the scattered photon, $\mathrm{d}\Omega$ is a solid angle element in the direction $\hat{\mathbf{k}}'$, and $\mathcal{J}(b,\mathbf{k}'\boldsymbol{\zeta}';a,\mathbf{k}\boldsymbol{\zeta})$ is a sum of products of dimensionless matrix

elements. The interesting feature here is that the quantity $\mathcal{J}(b, \mathbf{k}'\boldsymbol{\zeta}'; a, \mathbf{k}\boldsymbol{\zeta})$ is linear in $\boldsymbol{\zeta}$ and anti-linear in $\boldsymbol{\zeta}'$. It is convenient to express all polarisation states in the "natural" basis $\{\hat{\boldsymbol{\epsilon}}_1(\mathbf{k}), \hat{\boldsymbol{\epsilon}}_2(\mathbf{k})\}$ of linear polarisation (see Section C.1 in Appendix C). In this basis, the states of the incident and the scattered photons are represented as

$$\boldsymbol{\zeta} = \sum_i d_i \hat{\boldsymbol{\epsilon}}_i(\hat{\mathbf{k}}), \qquad \boldsymbol{\zeta}' = \sum_j f_j \hat{\boldsymbol{\epsilon}}_j(\hat{\mathbf{k}}') \tag{2.119}$$

and $\mathcal{J}(b, \mathbf{k}'\boldsymbol{\zeta}'; a, \mathbf{k}\boldsymbol{\zeta})$ is given by

$$\mathcal{J}(b, \mathbf{k}'\boldsymbol{\zeta}'; a, \mathbf{k}\boldsymbol{\zeta}) \equiv \sum_{i,s} d_i f_s^* T_{is} \tag{2.120}$$

with

$$T_{is} \equiv \mathcal{J}(b, \mathbf{k}'\hat{\boldsymbol{\epsilon}}'_s; a, \mathbf{k}\hat{\boldsymbol{\epsilon}}_i). \tag{2.121}$$

The DCS (2.118) then takes the form

$$\begin{aligned} \frac{\mathrm{d}^2\sigma(b\boldsymbol{\zeta}', a\boldsymbol{\zeta})}{\mathrm{d}E'\,\mathrm{d}\Omega} &= r_\mathrm{e}^2 \frac{E}{E'} \sum_{rs,ij} f_r f_s^* \left(T_{is} T_{jr}^*\right) d_i d_j^* \\ &= \sum_{rs,ij} f_r f_s^* \; \mathcal{S}_{sr,ij} \; d_i d_j^*, \end{aligned} \tag{2.122}$$

where we have set (note the ordering of the indices)

$$\mathcal{S}_{sr,ij} \equiv r_\mathrm{e}^2 \frac{E}{E'} \; T_{is} T_{jr}^*. \tag{2.123}$$

In matrix form,

$$\mathcal{S} = \begin{array}{c} \\ 11 \\ 22 \\ 12 \\ 21 \end{array} \begin{array}{c} \begin{array}{cccc} 11 & 22 & 12 & 21 \end{array} \\ \begin{pmatrix} T_{11}T_{11}^* & T_{21}T_{21}^* & T_{11}T_{21}^* & T_{21}T_{11}^* \\ T_{12}T_{12}^* & T_{22}T_{22}^* & T_{12}T_{22}^* & T_{22}T_{12}^* \\ T_{11}T_{12}^* & T_{21}T_{22}^* & T_{11}T_{22}^* & T_{21}T_{12}^* \\ T_{12}T_{11}^* & T_{22}T_{21}^* & T_{12}T_{21}^* & T_{22}T_{11}^* \end{pmatrix} \end{array} r_\mathrm{e}^2 \frac{E}{E'}. \tag{2.124}$$

The products $d_i d_j^*$ and $f_r f_s^*$ are the elements of the polarisation density matrices of incident and scattered photons (see Section C.2 of Appendix C),

$$\rho = |\boldsymbol{\zeta}\rangle\langle\boldsymbol{\zeta}|, \qquad \rho' = |\boldsymbol{\zeta}'\rangle\langle\boldsymbol{\zeta}'|. \tag{2.125}$$

On the other hand, the quantities

$$\rho_{sr}^{(\mathrm{sc})} \equiv \sum_{ij} \mathcal{S}_{sr,ij} \; d_i d_j^* \tag{2.126}$$

can be regarded as the elements of the density matrix $\rho^{(\mathrm{sc})}$ of the scattered photons. We can then write

$$\frac{\mathrm{d}^2\sigma(b\boldsymbol{\zeta}', a\boldsymbol{\zeta})}{\mathrm{d}E'\,\mathrm{d}\Omega} = \sum_{rs} f_r f_s^*\ \rho_{sr}^{(\mathrm{sc})} = \mathrm{Tr}(\rho'\rho^{(\mathrm{sc})}), \tag{2.127}$$

where "Tr" indicates the trace (sum of diagonal elements) of the matrix. In addition, by virtue of the identity (C.49), the matrix ρ' can be interpreted as the operator $\boldsymbol{\mathcal{F}}$ of an ideal polariser that transmits photons with the polarisation $\hat{\boldsymbol{\epsilon}}'_\alpha$ of the scattered beam.

In the case of completely polarised photons, Eq. (2.127) does not contain any new physics. However, now we can interpret the expression on the right-hand side as follows. The interaction with the atom modifies the density matrix of the incident photon beam through the transformation $\rho \to \rho^{(\mathrm{sc})}$ and the DCS is simply the expectation value of a measurement with the polarising filter $\boldsymbol{\mathcal{F}} = \rho'$ on the beam that results from the interaction. When the polarisation of scattered photons is not observed, the operator $\boldsymbol{\mathcal{F}}$ is that of the transparent filter, $\boldsymbol{\mathcal{F}} = I_2$. We then obtain from Eq. (2.127)

$$\frac{\mathrm{d}^2\sigma(b, a\boldsymbol{\zeta})}{\mathrm{d}E'\,\mathrm{d}\Omega} = \mathrm{Tr}(\rho^{(\mathrm{sc})}). \tag{2.128}$$

The great advantage of expressions (2.127) and (2.128) is that they are also valid for photon beams with partial polarisation. We can readily verify this assertion for the familiar case in which the incident beam is unpolarised ($\rho = \frac{1}{2}I_2$), and the polarisation of the scattered beam is not observed ($\boldsymbol{\mathcal{F}} = I_2$). Formula (2.128) then gives

$$\begin{aligned}\frac{\mathrm{d}^2\sigma(b, a)}{\mathrm{d}E'\,\mathrm{d}\Omega} &= \frac{1}{2}\sum_s \left[\mathcal{S}_{ss,11} + \mathcal{S}_{ss,22}\right] = \frac{1}{2}\left[\mathcal{S}_{11,11} + \mathcal{S}_{11,22} + \mathcal{S}_{22,11} + \mathcal{S}_{22,22}\right] \\ &= r_\mathrm{e}^2\,\frac{E}{E'}\,\frac{1}{2}\left[T_{11}T_{11}^* + T_{21}T_{21}^* + T_{12}T_{12}^* + T_{22}T_{22}^*\right],\end{aligned} \tag{2.129}$$

which is the result obtained with the habitual method of summing over final polarisation states and averaging over initial states.

It is convenient to use the four-vector representation of polarisation states, Eq. (C.61), to describe the polarisation of photon beams. Let P and Q denote the polarisation four-vectors of the incident and scattered photons, corresponding to the polarisation vectors $\boldsymbol{\zeta}$ and $\boldsymbol{\zeta}'$, respectively. The DCS can then be expressed in the form,

$$\frac{\mathrm{d}^2\sigma(b\mathsf{Q}, a\mathsf{P})}{\mathrm{d}E'\,\mathrm{d}\Omega} = \mathsf{D}_{\mathbf{Q}}^{\mathrm{T}}\mathcal{Z}_{34}\mathcal{S}\mathsf{D}_{\mathbf{P}} = \frac{1}{2}\mathsf{Q}^{\mathrm{T}}(\mathcal{A}\mathsf{P}), \tag{2.130}$$

with

$$\mathcal{A} \equiv 2\,\mathcal{X}^{\mathrm{T}}\mathcal{Z}_{34}\mathcal{S}\mathcal{X}. \tag{2.131}$$

The factor $\frac{1}{2}\mathsf{Q}^{\mathrm{T}}$ represents the action of an ideal filter that selects the final polarisation state [see Eq. (C.65)]. The matrix $\mathcal{A}$ transforms the Stokes vector of the incident photons into that of the scattered radiation,

$$\mathsf{P}_{\mathrm{sc}} = \mathcal{A}\mathsf{P}, \tag{2.132}$$

which corresponds to the density matrix $\rho^{(\mathrm{sc})}$, Eq. (2.126).

In the following we shall determine the matrix $\mathcal{A}$ for the elementary processes of Rayleigh and Compton scattering in an isotropic medium. For the sake of simplicity, we assume that the incident photons propagate in the direction $\hat{\mathbf{z}}$, and take the polarisation vectors $\hat{\boldsymbol{\epsilon}}_1$ and $\hat{\boldsymbol{\epsilon}}_2$ in the directions of the x and y axes, respectively. That is,

$$\hat{\mathbf{k}} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \qquad \hat{\boldsymbol{\epsilon}}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \qquad \hat{\boldsymbol{\epsilon}}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}. \tag{2.133}$$

Photons are scattered into directions $\hat{\mathbf{k}}'$ defined by the polar and azimuthal scattering angles, θ and ϕ, respectively. To describe the polarisation of scattered photons, we use Stokes vectors with respect to the basis of linear polarisation states $\{\hat{\boldsymbol{\epsilon}}_1(\hat{\mathbf{k}}'), \hat{\boldsymbol{\epsilon}}_2(\hat{\mathbf{k}}')\}$ defined by Eqs. (C.6) in Appendix C (see also Fig. C.1). Then,

$$\hat{\mathbf{k}}' = \begin{pmatrix} \cos\phi\, \sin\theta \\ \sin\phi\, \sin\theta \\ \cos\theta \end{pmatrix}, \qquad \hat{\boldsymbol{\epsilon}}'_1 = \begin{pmatrix} \cos\phi\, \cos\theta \\ \sin\phi\, \cos\theta \\ -\sin\theta \end{pmatrix}, \qquad \hat{\boldsymbol{\epsilon}}'_2 = \begin{pmatrix} -\sin\phi \\ \cos\phi \\ 0 \end{pmatrix}. \tag{2.134}$$

For simulation purposes, this convention has the advantage that the inverse of the rotation $R(\theta, \phi)$, Eq. (C.5), transforms the vectors $\hat{\mathbf{k}}'$, $\hat{\boldsymbol{\epsilon}}'_1$ and $\hat{\boldsymbol{\epsilon}}'_2$ into the triad (2.133). Note that $\hat{\boldsymbol{\epsilon}}'_1$ lies on the scattering plane, defined by the directions $\hat{\mathbf{k}}$ and $\hat{\mathbf{k}}'$, and $\hat{\boldsymbol{\epsilon}}'_2$ is on the plane $z = 0$, see Fig. 2.14.

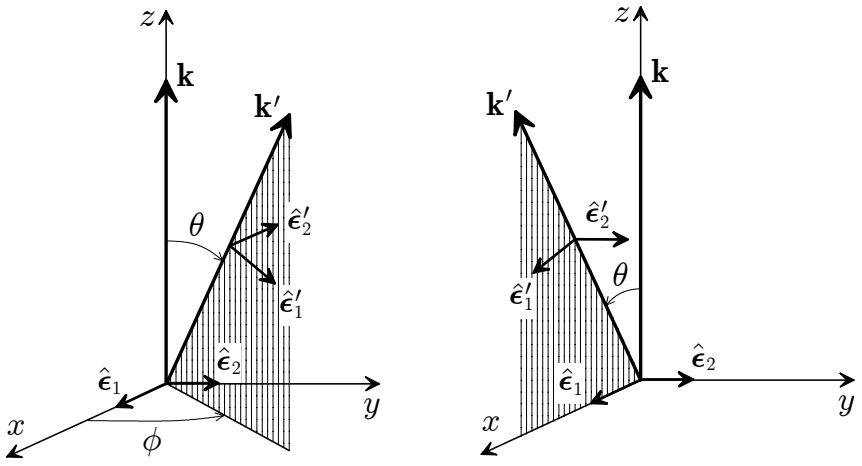

Figure 2.14: Linear polarisation vectors used in the description of scattering events. The left diagram corresponds to the general case of scattering with $\phi \neq 0$. The right diagram represents the case $\phi = 0$, *i.e.*, the scattering plane (hatched) coincides with the zero-azimuth plane; the polarisation vectors $\hat{\boldsymbol{\epsilon}}_1$ and $\hat{\boldsymbol{\epsilon}}'_1$ are both on this plane. This is the situation assumed in calculations of the matrix $\mathcal{A}^{(0)}$, Eq. (2.137).

Most formulas take simpler forms when the polarisation vectors $\hat{\boldsymbol{\epsilon}}_1$ and $\hat{\boldsymbol{\epsilon}}'_1$ are both on the plane of scattering. Hence, to calculate the matrix $\mathcal{A}$ it is advantageous to proceed in two stages. We first perform a rotation of angle ϕ of the polarisation vectors $\hat{\boldsymbol{\epsilon}}_1$ and $\hat{\boldsymbol{\epsilon}}_2$ about the $\hat{\mathbf{k}}$ axis, *i.e.*, a rotation of the zero-azimuth plane (see Fig. 2.14). The rotated zero-azimuth plane then coincides with the scattering plane and, therefore, the rotated vector $\hat{\boldsymbol{\epsilon}}_1$ also lies on that plane. The rotated polarisation vectors are (for simplicity we use the same symbols as for the original vectors)

$$\hat{\boldsymbol{\epsilon}}_1 = \begin{pmatrix} \cos\phi \\ \sin\phi \\ 0 \end{pmatrix}, \qquad \hat{\boldsymbol{\epsilon}}_2 = \begin{pmatrix} -\sin\phi \\ \cos\phi \\ 0 \end{pmatrix}. \tag{2.135}$$

We have

$$\hat{\boldsymbol{\epsilon}}_1 \cdot \hat{\boldsymbol{\epsilon}}'_1 = \cos\theta, \quad \hat{\boldsymbol{\epsilon}}_2 \cdot \hat{\boldsymbol{\epsilon}}'_2 = 1, \quad \hat{\boldsymbol{\epsilon}}_1 \cdot \hat{\boldsymbol{\epsilon}}'_2 = \hat{\boldsymbol{\epsilon}}_2 \cdot \hat{\boldsymbol{\epsilon}}'_1 = 0. \tag{2.136}$$

The effect of the rotation on the Stokes vector of the incident beam is described by the transformation (C.68), $\mathsf{P}' = \mathcal{R}(\phi)\mathsf{P}$. Hence the matrix $\mathcal{A}$ can be expressed as

$$\mathcal{A} = \mathcal{A}^{(0)}\mathcal{R}(\phi), \tag{2.137}$$

where the matrix $\mathcal{A}^{(0)}$ corresponds to the case of scattering in the zero-azimuth plane (with $\phi = 0$), where the polarisation vectors take the forms (2.134) and (2.135) and satisfy the orthogonality relations (2.136).

2.7.1 Rayleigh scattering

In the case of Rayleigh scattering ($E_b = E_a$, $E' = E$), the transition matrix elements T_{is} are

$$T_{is} = \mathcal{F}(q)\,(\hat{\boldsymbol{\epsilon}}_i \cdot \hat{\boldsymbol{\epsilon}}'_s), \tag{2.138}$$

with

$$\mathcal{F}(q) = F(q, Z) + f' + \mathrm{i}f'', \tag{2.139}$$

where $F(q, Z)$ is the atomic form factor, Eq. (2.16), which depends on the momentum transfer

$$q = 2k\sin(\theta/2), \tag{2.140}$$

and $f' + \mathrm{i}f''$ is the anomalous-scattering factor. As indicated in Section 2.2, the factor $f' + \mathrm{i}f''$ varies slowly with the scattering angle and here it is approximated by its value at $\theta = 0$.

The essential fact is that the matrix element (2.138) is proportional to the scalar product of the polarisation vectors of the incident and scattered beams. Therefore, in the case of scattering in the zero-azimuth plane [see Eqs. (2.134) and (2.135)] we have

$$T_{11} = \mathcal{F}(q)\cos\theta, \quad T_{22} = \mathcal{F}(q), \quad T_{12} = 0, \quad T_{21} = 0. \tag{2.141}$$

The corresponding $\mathcal{S}$ matrix, Eq. (2.124), takes the form

$$\mathcal{S}^{(0)} = \begin{pmatrix} \cos^2\theta & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & \cos\theta \end{pmatrix} r_\mathrm{e}^2 \, |\mathcal{F}(q)|^2. \tag{2.142}$$

Then

$$\begin{aligned} \mathcal{A}_\mathrm{Ra}^{(0)} &= 2\,\mathcal{X}^\mathrm{T}\mathcal{Z}_{34}\mathcal{S}^{(0)}\mathcal{X} \\ &= \frac{r_\mathrm{e}^2}{2}\,|\mathcal{F}(q)|^2 \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & -\mathrm{i} & \mathrm{i} \\ 1 & -1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \cos^2\theta & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & -\mathrm{i} & 0 \\ 0 & 1 & \mathrm{i} & 0 \end{pmatrix}. \end{aligned}$$

That is,

$$\mathcal{A}_\mathrm{Ra}^{(0)} = \frac{r_\mathrm{e}^2}{2}\,|\mathcal{F}(q)|^2 \begin{pmatrix} \cos^2\theta + 1 & 0 & 0 & \cos^2\theta - 1 \\ 0 & 2\cos\theta & 0 & 0 \\ 0 & 0 & 2\cos\theta & 0 \\ \cos^2\theta - 1 & 0 & 0 & \cos^2\theta + 1 \end{pmatrix}. \tag{2.143}$$

Finally, from Eq. (2.130), the DCS for Rayleigh scattering of polarised photons is

$$\frac{\mathrm{d}\sigma_\mathrm{Ra}(\mathsf{Q},\mathsf{P})}{\mathrm{d}\Omega} = \frac{1}{2}\,\mathsf{Q}^\mathrm{T}\{\mathcal{A}_\mathrm{Ra}^{(0)}\mathcal{R}(\phi)\mathsf{P}\}. \tag{2.144}$$

When polarisation of the scattered photons is not observed, the DCS is given by this expression with the Stokes vector $\mathsf{Q}_I = (2,0,0,0)$ of the transparent filter,

$$\begin{aligned} \frac{\mathrm{d}\sigma_\mathrm{Ra}(\mathsf{P})}{\mathrm{d}\Omega} &= (1,0,0,0)\{\mathcal{A}_\mathrm{Ra}^{(0)}\mathcal{R}(\phi)\mathsf{P}\} \\ &= \frac{r_\mathrm{e}^2}{2}\,|\mathcal{F}(q)|^2 \left(\cos^2\theta + 1, 0, 0, \cos^2\theta - 1\right) \begin{pmatrix} 1 \\ \cos(2\phi)P_1 - \sin(2\phi)P_3 \\ P_2 \\ \sin(2\phi)P_1 + \cos(2\phi)P_3 \end{pmatrix} \\ &= \frac{r_\mathrm{e}^2}{2}\,|\mathcal{F}(q)|^2 \left\{\cos^2\theta + 1 + (\cos^2\theta - 1)\left[\sin(2\phi)P_1 + \cos(2\phi)P_3\right]\right\}. \end{aligned} \tag{2.145}$$

Note that terms which depend on ϕ yield null contributions when integrated over this angle. We can then consider the cross section differential in only the polar deflection,

$$\frac{\mathrm{d}\sigma_\mathrm{Ra}(\mathsf{P})}{\mathrm{d}(\cos\theta)} = \int_0^{2\pi} \frac{\mathrm{d}\sigma_\mathrm{Ra}(\mathsf{P})}{\mathrm{d}\Omega}\,\mathrm{d}\phi = 2\pi r_\mathrm{e}^2\,|\mathcal{F}(q)|^2\,\frac{\cos^2\theta + 1}{2}, \tag{2.146}$$

which is independent of the polarisation of the incident photons. This feature, which is a consequence of the assumed isotropy of the target atoms, implies that the distribution of the polar angle θ for polarised photons is the same as for unpolarised radiation. Finally, the total cross section for Rayleigh scattering of incident photons with arbitrary polarisation P is

$$\sigma_{\mathrm{Ra}}(\mathsf{P}) = 2\pi \int_{-1}^{1} \frac{r_{\mathrm{e}}^2}{2} |\mathcal{F}(q)|^2 (\cos^2\theta + 1)\, \mathrm{d}(\cos\theta), \tag{2.147}$$

irrespective of the polarisation of the incident photons.

2.7.2 Compton scattering

Let us now study the effects of polarisation in Compton scattering ($E_b > E_a$, $E' < E$). For simplicity, we assume that these effects are not affected by the binding of the target electrons. We also assume that target electrons are unpolarised, and that the spin of the recoiling Compton electron is not observed. That is, we will limit our considerations to the theory of photon scattering by free electrons at rest, which leads to the Klein–Nishina formula, Eq. (2.27). The matrix elements $\mathcal{S}_{sr,ij}$, Eq. (2.123), for Compton scattering have been evaluated by Wightman (1948) using the methods described in Heitler's (1954) book. These matrix elements are given by the general expression

$$\begin{aligned}\mathcal{S}_{sr,ij} = \frac{r_{\mathrm{e}}^2}{4}\frac{E'^2}{E^2} \Bigg\{ &\left(\hat{\boldsymbol{\epsilon}}_i \cdot \hat{\boldsymbol{\epsilon}}_s'^*\right)\left(\hat{\boldsymbol{\epsilon}}_j^* \cdot \hat{\boldsymbol{\epsilon}}_r'\right)\left[4 + \frac{E-E'}{\mathrm{m_e}c^2}(1-\cos\theta)\right] \\ &+ \left[\left(\hat{\boldsymbol{\epsilon}}_i \cdot \hat{\boldsymbol{\epsilon}}_j^*\right)\left(\hat{\boldsymbol{\epsilon}}_r' \cdot \hat{\boldsymbol{\epsilon}}_s'^*\right) - \left(\hat{\boldsymbol{\epsilon}}_i \cdot \hat{\boldsymbol{\epsilon}}_r'\right)\left(\hat{\boldsymbol{\epsilon}}_j^* \cdot \hat{\boldsymbol{\epsilon}}_s'^*\right)\right] \frac{E-E'}{\mathrm{m_e}c^2}(1-\cos\theta) \Bigg\},\end{aligned} \tag{2.148}$$

which is valid even when the bases of polarisation vectors are complex. The expression in curly braces depends on the quantity

$$C = \frac{E-E'}{\mathrm{m_e}c^2}\,(1-\cos\theta), \tag{2.149}$$

which is always positive. Note that, with the target electron initially at rest, the energy of the scattered photon is that of the Compton line, Eq. (2.26),

$$E' = E_{\mathrm{C}} = \frac{E}{1 + (E/\mathrm{m_e}c^2)(1-\cos\theta)}.$$

Hence,

$$C = \frac{(E/\mathrm{m_e}c^2)^2(1-\cos\theta)^2}{1 + (E/\mathrm{m_e}c^2)(1-\cos\theta)}. \tag{2.150}$$

In the case of scattering on the zero-azimuth plane, using the polarisation vectors (2.134) and (2.135), and their orthogonality relations (2.136), we find that the matrix

$\mathcal{S}$ for Compton scattering takes the form

$$\mathcal{S}_{\mathrm{Co}}^{(0)} = \frac{r_{\mathrm{e}}^2}{4}\frac{E'^2}{E^2}\begin{pmatrix} 4\cos^2\theta + C & C & 0 & 0 \\ C & 4 + C & 0 & 0 \\ 0 & 0 & (4+C)\cos\theta & -C\cos\theta \\ 0 & 0 & -C\cos\theta & (4+C)\cos\theta \end{pmatrix}. \tag{2.151}$$

Then, from Eq. (2.131),

$$\begin{aligned}\mathcal{A}_{\mathrm{Co}}^{(0)} &= 2\,\mathcal{X}^{\mathrm{T}}\mathcal{Z}_{34}\mathcal{S}_{\mathrm{Co}}^{(0)}\mathcal{X} \\ &= \frac{r_{\mathrm{e}}^2}{2}\frac{E'^2}{E^2}\begin{pmatrix} (\cos^2\theta + 1) + C & 0 & 0 & \cos^2\theta - 1 \\ 0 & 2\cos\theta & 0 & 0 \\ 0 & 0 & (2+C)\cos\theta & 0 \\ \cos^2\theta - 1 & 0 & 0 & \cos^2\theta + 1 \end{pmatrix}. \end{aligned} \tag{2.152}$$

Now, the DCS for Compton scattering is given by Eq. (2.130),

$$\frac{\mathrm{d}\sigma_{\mathrm{Co}}(\mathsf{Q},\mathsf{P})}{\mathrm{d}\Omega} = \frac{1}{2}\,\mathsf{Q}^{\mathrm{T}}\{\mathcal{A}_{\mathrm{Co}}^{(0)}\,\mathcal{R}(\phi)\mathsf{P}\}, \tag{2.153}$$

where the quantity in braces is the Stokes vector of the photons scattered in the direction θ, ϕ.

When polarisation of the scattered photons is not observed, the DCS for scattering of incident photons with arbitrary polarisation P is given by Eq. (2.153) with the Stokes vector $\mathsf{Q}_I = (2,0,0,0)$ of the transparent filter,

$$\begin{aligned}\frac{\mathrm{d}\sigma_{\mathrm{Co}}(\mathsf{P})}{\mathrm{d}\Omega} &= (1,0,0,0)\{\mathcal{A}_{\mathrm{Co}}^{(0)}\mathcal{R}(\phi)\mathsf{P}\} \\ &= \frac{r_{\mathrm{e}}^2}{2}\frac{E'^2}{E^2}\left(\cos^2\theta + 1 + C, 0, 0, \cos^2\theta - 1\right)\begin{pmatrix} 1 \\ \cos(2\phi)P_1 - \sin(2\phi)P_3 \\ P_2 \\ \sin(2\phi)P_1 + \cos(2\phi)P_3 \end{pmatrix} \\ &= \frac{r_{\mathrm{e}}^2}{2}\frac{E'^2}{E^2}\left\{\cos^2\theta + 1 + C + (\cos^2\theta - 1)\left[\sin(2\phi)P_1 + \cos(2\phi)P_3\right]\right\}. \end{aligned} \tag{2.154}$$

Again, terms that depend on ϕ yield null contributions when integrated over this angle. The cross section differential in only the polar deflection is

$$\frac{\mathrm{d}\sigma_{\mathrm{Co}}(\mathsf{P})}{\mathrm{d}(\cos\theta)} = \int_0^{2\pi}\frac{\mathrm{d}\sigma_{\mathrm{Co}}(\mathsf{P})}{\mathrm{d}\Omega}\,\mathrm{d}\phi = \pi r_{\mathrm{e}}^2\frac{E'^2}{E^2}\,(\cos^2\theta + 1 + C). \tag{2.155}$$

Because of the symmetry of the interaction, this DCS is independent of the polarisation of the incident photons. Hence, the distribution of the polar angle θ for scattering of

polarised radiation is the same as for unpolarised photons. Inserting expression (2.150), we obtain

$$\frac{\mathrm{d}\sigma_{\mathrm{Co}}(\mathsf{P})}{\mathrm{d}(\cos\theta)} = \pi r_{\mathrm{e}}^2 \frac{E'^2}{E^2}\left(\frac{E'}{E} + \frac{E}{E'} - \sin^2\theta\right), \tag{2.156}$$

which is the Klein–Nishina DCS, Eq. (2.27), multiplied by the factor 2π that results from integration over ϕ.

From the foregoing discussion it is clear that photons scattered into the direction (θ, ϕ) have the polarisation defined by the Stokes vector

$$\mathsf{P}' = \mathcal{B}_{\mathrm{Co}}\mathsf{P}, \tag{2.157}$$

where the matrix $\mathcal{B}_{\mathrm{Co}}$ is equal to the product $\mathcal{A}^{(0)}_{\mathrm{Co}}\,\mathcal{R}(\phi)$. However, in our Monte Carlo simulations, individual photons are followed. Since the Stokes vector of a photon state has its zeroth component equal to unity [see Eq. (C.61)], the Stokes vector of a scattered photon will always be renormalised in this way. Hence, we can disregard constant factors in the matrix $\mathcal{B}_{\mathrm{Co}}$ and write

$$\mathcal{B}_{\mathrm{Co}} = \begin{pmatrix} (\cos^2\theta + 1) + C & 0 & 0 & \cos^2\theta - 1 \\ 0 & 2\cos\theta & 0 & 0 \\ 0 & 0 & (2 + C)\cos\theta & 0 \\ \cos^2\theta - 1 & 0 & 0 & \cos^2\theta + 1 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(2\phi) & 0 & -\sin(2\phi) \\ 0 & 0 & 1 & 0 \\ 0 & \sin(2\phi) & 0 & \cos(2\phi) \end{pmatrix}. \tag{2.158}$$

From the similarity between the matrices $\mathcal{A}^{(0)}_{\mathrm{Ra}}$ and $\mathcal{A}^{(0)}_{\mathrm{Co}}$, Eqs. (2.143) and (2.152), we see that polarisation changes in Rayleigh scattering are described by a matrix $\mathcal{B}_{\mathrm{Ra}}$ which is given by this expression with $C = 0$.

It is interesting to note that the transformation matrix (2.158) does not mix the component of circular polarisation P_2 with other components (the only non-zero element in the third column and in the third row is the one on the diagonal). This implies that, if polarisation of the incident photons has no circular component, $P_2 = 0$, the scattered photons also have $P'_2 = 0$. That is, scattering is not able to produce circular polarisation when it is not present in the incident beam. If we were interested in describing only the transport of photons with $P_2 = 0$, we could simply remove the component P_2 of the Stokes vector and of the transformation matrices. In fact, this is equivalent to the linear-polarisation approximation adopted in the EGS5 code (Namito *et al.*, 1993; Hirayama *et al.*, 2006).

2.8 Simulation of interactions of polarised photons

Simulation of polarised photons is largely simplified by the fact that, because of the assumed isotropy of the medium, the total cross sections (and also the attenuation coefficients) of the various interaction mechanisms are independent of the polarisation of the transported photons. This means that sampling of the free path and of the active mechanism in each interaction event can be performed by using the same methods as in conventional (scalar) Monte Carlo simulation of unpolarised photons. As shown above, polarisation of the transported photons has an appreciable effect on Rayleigh and Compton scattering. In the photoelectric absorption, the polarization of the photon does influence the angular distribution of the photoelectrons and may be relevant, *e.g.*, in x-ray photoelectron spectroscopy (*e.g.*, Kövér, 2010) and in the simulation of dose distributions from synchrotron photon beams. Polarisation is generally unimportant for electron-pair production, because this process occurs only for high-energy photons. Since the generated charged particles have ranges much smaller than the mean free path of the parent photon, the possible effect of the photon polarization is shadowed by the multiple scattering of these particles as soon as they travel a fraction of their range.

2.8.1 Rayleigh and Compton scattering

In the simulation of Rayleigh and Compton scattering of polarized photons we follow a scheme similar to the one adopted by Fernández *et al.* (1998) in their vector Monte Carlo code MCSHAPE for the simulation of polarised photons in simple planar geometries. As shown in previous Sections, the PDF of the polar deflection $\cos\theta$ is independent of the polarisation of the incident photons. Therefore, $\cos\theta$ can be sampled using the same algorithms as for unpolarized photons. The PDF of the azimuthal scattering angle ϕ (with $\cos\theta$ fixed) is given by the DCSs (2.145) or (2.154). We have

$$p(\phi) = N\left\{\cos^2\theta + 1 + C - \sin^2\theta\,[\sin(2\phi)P_1 + \cos(2\phi)P_3]\right\}, \tag{2.159}$$

where N is a normalisation constant. Note that the PDF for Rayleigh scattering, cf. Eq. (2.145), is obtained by setting $C = 0$. It is clear that polarisation of the incident photon only affects the distribution of azimuthal angles. In the case of circularly polarised photons, with $P_2 \neq 0$ and $P_1 = P_3 = 0$, this distribution is still uniform.

Random values of ϕ from the PDF (2.159) can be sampled by a rejection method (see Section 1.2.5). This PDF has extrema, maxima and minima, at angles $\phi_{\rm extr}$ such that

$$\tan(2\phi_{\rm extr}) = P_1/P_3. \tag{2.160}$$

It can be readily verified that the maximum value of $p(\phi)$ occurs at an angle $\phi_{\rm max}$ such that

$$\sin(2\phi_{\rm max}) = \frac{-P_1}{\sqrt{P_1^2 + P_3^2}} \qquad \text{and} \qquad \cos(2\phi_{\rm max}) = \frac{-P_3}{\sqrt{P_1^2 + P_3^2}}. \tag{2.161}$$

Let us consider the function

$$g(\phi) = \cos^2\theta + 1 + C - \sin^2\theta\,[\sin(2\phi)P_1 + \cos(2\phi)P_3]\,, \tag{2.162}$$

which attains its maximum value at $\phi_{\max}$,

$$g(\phi_{\max}) = \cos^2\theta + 1 + C + \sin^2\theta\sqrt{P_1^2 + P_3^2}\,. \tag{2.163}$$

To sample ϕ we adopt the following rejection algorithm:

(i) Sample a value ϕ from the uniform distribution $U_{0,2\pi}(\phi)$, *i.e.*, generate a random number ξ and set $\phi = 2\pi\xi$.

(ii) Generate another random number ξ.

(iii) If $\xi g(\phi_{\max}) > g(\phi)$, go to step (i).

(iv) Deliver ϕ.

Note that this sampling algorithm is exact. Furthermore, its efficiency improves when the value of C increases. The least favourable case is found when $C = 0$ (which occurs for Rayleigh scattering and for forward Compton scattering, with $\theta = 0$). Efficiency is higher than (or equal to) 50%, *i.e.*, on average we have to generate less than four random numbers to get an accepted value of ϕ.

The simulation of random histories is similar to that of unpolarised radiation (see Section 1.4.4). After the n-th interaction, a photon propagates in a direction $\hat{\mathbf{k}}_n$, with energy E and polarisation described by the Stokes vector P_n. The direction vector $\hat{\mathbf{k}}_n$ can be represented by means of its direction cosines, (u, v, w), or by its polar and azimuthal angles, (ϑ, φ),

$$\hat{\mathbf{k}} = \begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} \cos\varphi\sin\vartheta \\ \sin\varphi\sin\vartheta \\ \cos\vartheta \end{pmatrix}. \tag{2.164}$$

Note that[5]

$$\vartheta = \arccos w, \qquad \varphi = \texttt{ATAN2}(y, x). \tag{2.165}$$

In the Monte Carlo code, direction cosines are convenient for fast calculation of space displacements, while the angles (ϑ, φ) are better suited for defining rotations (see below). The Stokes vector P_n is referred to the "natural" basis of polarisation vectors, $\hat{\boldsymbol{\epsilon}}_1(\hat{\mathbf{k}}_n)$

[5]The Fortran intrinsic function `ATAN2`(y, x) gives the arc-tangent of y/x resolved into the correct quadrant; the result is in the range $(-\pi, \pi)$.

and $\hat{\boldsymbol{\epsilon}}_2(\hat{\mathbf{k}}_n)$, defined by Eqs. (C.6), and normalised with its zeroth component equal to unity. The rotation

$$\mathcal{R}(\vartheta,\varphi) \equiv \mathcal{R}(\varphi\hat{\mathbf{z}})\,\mathcal{R}(\vartheta\hat{\mathbf{y}}) = \begin{pmatrix} \cos\varphi\cos\vartheta & -\sin\varphi & \cos\varphi\sin\vartheta \\ \sin\varphi\cos\vartheta & \cos\varphi & \sin\varphi\sin\vartheta \\ -\sin\vartheta & 0 & \cos\vartheta \end{pmatrix} \tag{2.166}$$

transforms the unit vectors $\{\hat{\mathbf{z}},\ \hat{\mathbf{x}},\ \hat{\mathbf{y}}\}$ of the laboratory frame into the corresponding triad $\{\hat{\mathbf{k}}_n,\ \hat{\boldsymbol{\epsilon}}_1(\hat{\mathbf{k}}_n),\ \hat{\boldsymbol{\epsilon}}_2(\hat{\mathbf{k}}_n)\}$.

The polar and azimuthal scattering angles in the next, $(n+1)$-th interaction of the photon are sampled from the relevant PDFs as described above. To determine the direction $\hat{\mathbf{k}}_{n+1}$ and polarisation P_{n+1} of the scattered photon we perform the following sequence of rotations:
1) A rotation $\mathcal{R}^{-1}(\vartheta,\varphi) = \mathcal{R}(-\vartheta\hat{\mathbf{y}})\mathcal{R}(-\varphi\hat{\mathbf{z}})$, which transforms the vectors $\{\hat{\mathbf{k}}_n,\ \hat{\boldsymbol{\epsilon}}_1(\hat{\mathbf{k}}_n),\ \hat{\boldsymbol{\epsilon}}_2(\hat{\mathbf{k}}_n)\}$ into $\{\hat{\mathbf{z}},\ \hat{\mathbf{x}},\ \hat{\mathbf{y}}\}$. This rotation keeps the relative orientations of the unit base vectors unaltered and, therefore, it does not modify the Stokes vector P_n.
2) A rotation $\mathcal{R}(\phi\hat{\mathbf{z}})\mathcal{R}(\theta\hat{\mathbf{y}})$ that describes the effect of an interaction with the sampled scattering angles for photons incident in the direction of the z axis. This rotation transforms the vectors $\{\hat{\mathbf{z}},\ \hat{\mathbf{x}},\ \hat{\mathbf{y}}\}$ into

$$\hat{\mathbf{k}}' = \begin{pmatrix} \cos\phi\,\sin\theta \\ \sin\phi\,\sin\theta \\ \cos\theta \end{pmatrix}, \qquad \hat{\boldsymbol{\epsilon}}'_1 = \begin{pmatrix} \cos\phi\,\cos\theta \\ \sin\phi\,\cos\theta \\ -\sin\theta \end{pmatrix}, \qquad \hat{\boldsymbol{\epsilon}}'_2 = \begin{pmatrix} -\sin\phi \\ \cos\phi \\ 0 \end{pmatrix}. \tag{2.167}$$

Polarisation of the scattered photon, relative to the "natural" basis of polarisation vectors for the direction $\hat{\mathbf{k}}'$, $\hat{\boldsymbol{\epsilon}}_1(\hat{\mathbf{k}}')$ and $\hat{\boldsymbol{\epsilon}}_2(\hat{\mathbf{k}}')$, is obtained from Eq. (2.157),

$$\mathsf{P}' = \mathcal{B}_{\text{Ra or Co}}\mathsf{P}_n, \tag{2.168}$$

and is renormalised so that $P'_0 = 1$. That is, the transformed Stokes parameters are given by

$$\begin{aligned} P'_0 &= (\cos^2\theta + 1 + C) + (\cos^2\theta - 1)[\sin(2\phi)P_1 + \cos(2\phi)P_3], \\ P'_1 &= \{2\cos\theta[\cos(2\phi)P_1 - \sin(2\phi)P_3]\}\,/P'_0, \\ P'_2 &= \{(2+C)\cos\theta P_2\}\,/P'_0, \\ P'_3 &= \left\{(\cos^2\theta - 1) + (\cos^2\theta + 1)[\sin(2\phi)P_1 + \cos(2\phi)P_3]\right\}/P'_0. \end{aligned} \tag{2.169}$$

After computing the parameters P'_1 to P'_3, the Stokes four-vector P' is obtained by setting $P'_0 = 1$. In the case of Compton scattering, the constant C is given by Eq. (2.150). For Rayleigh scattering, $C = 0$.
3) A rotation $\mathcal{R}(\vartheta,\varphi) = \mathcal{R}(\varphi\hat{\mathbf{z}})\mathcal{R}(\vartheta\hat{\mathbf{y}})$, which inverts the rotation of the first step. Again,

this rotation leaves the Stokes vector unaltered so that $\mathsf{P}_{n+1} = \mathsf{P}'$. The final direction vector is

$$\hat{\mathbf{k}}_{n+1} = \mathcal{R}(\vartheta,\varphi)\hat{\mathbf{k}}' = \begin{pmatrix} \cos\varphi\cos\vartheta & -\sin\varphi & \cos\varphi\sin\vartheta \\ \sin\varphi\cos\vartheta & \cos\varphi & \sin\varphi\sin\vartheta \\ -\sin\vartheta & 0 & \cos\vartheta \end{pmatrix} \begin{pmatrix} \cos\phi\,\sin\theta \\ \sin\phi\,\sin\theta \\ \cos\theta \end{pmatrix}, \tag{2.170}$$

and its direction cosines are given by the same formulas as for unpolarised particles [see Eqs. (1.131)],

$$\begin{aligned} u' &= u\cos\theta + \frac{\sin\theta}{\sqrt{1-w^2}}\left[uw\cos\phi - v\sin\phi\right], \\ v' &= v\cos\theta + \frac{\sin\theta}{\sqrt{1-w^2}}\left[vw\cos\phi + u\sin\phi\right], \\ w' &= w\cos\theta - \sqrt{1-w^2}\sin\theta\cos\phi. \end{aligned} \tag{2.171}$$

2.8.2 Angular distribution of photoelectrons

The angular distribution of photoelectrons emitted by absorption of polarised photons is described by means the DCS calculated from the relativistic first-order plane-wave Born approximation with K-shell hydrogenic electron wave functions (Sauter, 1931). Assuming that photons are polarized along the direction of the x axis [*i.e.*, with Poincaré vector $\mathbf{P} = (0,0,1)$], and using the same notation as in Eq. (2.4), Sauter's result (for a single K-shell electron) can be expressed as (Sauter, 1931; Bethe and Salpeter, 1957)

$$\begin{aligned} \frac{\mathrm{d}\sigma^{\mathrm{ph}}(\hat{\mathbf{x}})}{\mathrm{d}\Omega_{\mathrm{e}}} &= \alpha^4 r_{\mathrm{e}}^2 \left(\frac{Z}{\kappa}\right)^5 \frac{\beta^3}{\gamma} \frac{\sin^2\theta_{\mathrm{e}}}{(1-\beta\cos\theta_{\mathrm{e}})^4} \left\{ \left[1 + \frac{1}{2}\gamma(\gamma-1)(\gamma-2)(1-\beta\cos\theta_{\mathrm{e}})\right] \right. \\ &\left. + \cos(2\phi_{\mathrm{e}}) \left[1 - \frac{1}{2}\gamma(\gamma-1)(1-\beta\cos\theta_{\mathrm{e}})\right] \right\}, \end{aligned} \tag{2.172}$$

where the quantities β and γ pertain to the emerging photoelectron [see Eq. (2.5)]. Comparison with the general result from the elementary theory of the photoeffect [formula (70) in (Sabbatucci and Salvat, 2016)] shows that the DCS for photons with arbitrary polarization $\mathbf{P} = (P_1, P_2, P_3)$ $(P \leq 1)$ is given by

$$\begin{aligned} \frac{\mathrm{d}\sigma^{\mathrm{ph}}(\mathbf{P})}{\mathrm{d}\Omega_{\mathrm{e}}} &= \alpha^4 r_{\mathrm{e}}^2 \left(\frac{Z}{\kappa}\right)^5 \frac{\beta^3}{\gamma} \frac{\sin^2\theta_{\mathrm{e}}}{(1-\beta\cos\theta_{\mathrm{e}})^4} \left\{ \left[1 + \frac{1}{2}\gamma(\gamma-1)(\gamma-2)(1-\beta\cos\theta_{\mathrm{e}})\right] \right. \\ &\left. + \left[P_3\,\cos(2\phi_{\mathrm{e}}) + P_1\,\sin(2\phi_{\mathrm{e}})\right] \left[1 - \frac{1}{2}\gamma(\gamma-1)(1-\beta\cos\theta_{\mathrm{e}})\right] \right\}. \end{aligned} \tag{2.173}$$

The Sauter DCS for unpolarized photons, Eq. (2.4), is obtained by averaging this expression over polarization directions, *i.e.*, over the azimuthal angle ϕ_{e}.

The initial direction of the photoelectron is determined by the polar and azimuthal angles, θ_e and ϕ_e, respectively. Because of the isotropy of the medium, the PDF of the polar angle is the same as for unpolarised photons, and random values of $\cos\theta_e$ can be generated using the sampling algorithm described in Section 2.1.1.1. For a given polar angle, the unnormalised PDF of the azimuthal angle is

$$p(\phi_e) = A + B\,[P_3\,\cos(2\phi_e) + P_1\,\sin(2\phi_e)] \tag{2.174}$$

with

$$A = 1 + \frac{1}{2}\gamma(\gamma-1)(\gamma-2)(1-\beta\cos\theta_e) \tag{2.175a}$$

and

$$B = 1 - \frac{1}{2}\gamma(\gamma-1)(1-\beta\cos\theta_e). \tag{2.175b}$$

This function has its extrema at angles $\phi_{e,max}$ such that

$$\tan(2\phi_{e,max}) = P_1/P_3 \tag{2.176}$$

and its maximum value is

$$\begin{aligned} p_{max} \;=\; \max\Big\{ & A + B\,[P_3\,\cos(2\phi_{e,max}) + P_1\,\sin(2\phi_{e,max})]\,, \\ & A - B\,[P_3\,\cos(2\phi_{e,max}) + P_1\,\sin(2\phi_{e,max})] \Big\}\,. \end{aligned} \tag{2.177}$$

Random values of ϕ_e can be generated easily by using the rejection method (see Section 1.2.5) with the uniform distribution $\pi(\phi_e) = U_{0,2\pi}(\phi_e)$. The sampling efficiency depends on the degree of polarisation; it equals unity for unpolarised photons ($P = 0$) and is slightly larger than 0.5 in the least favourable case of low-energy photons in pure polarization states ($P = 1$).

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.

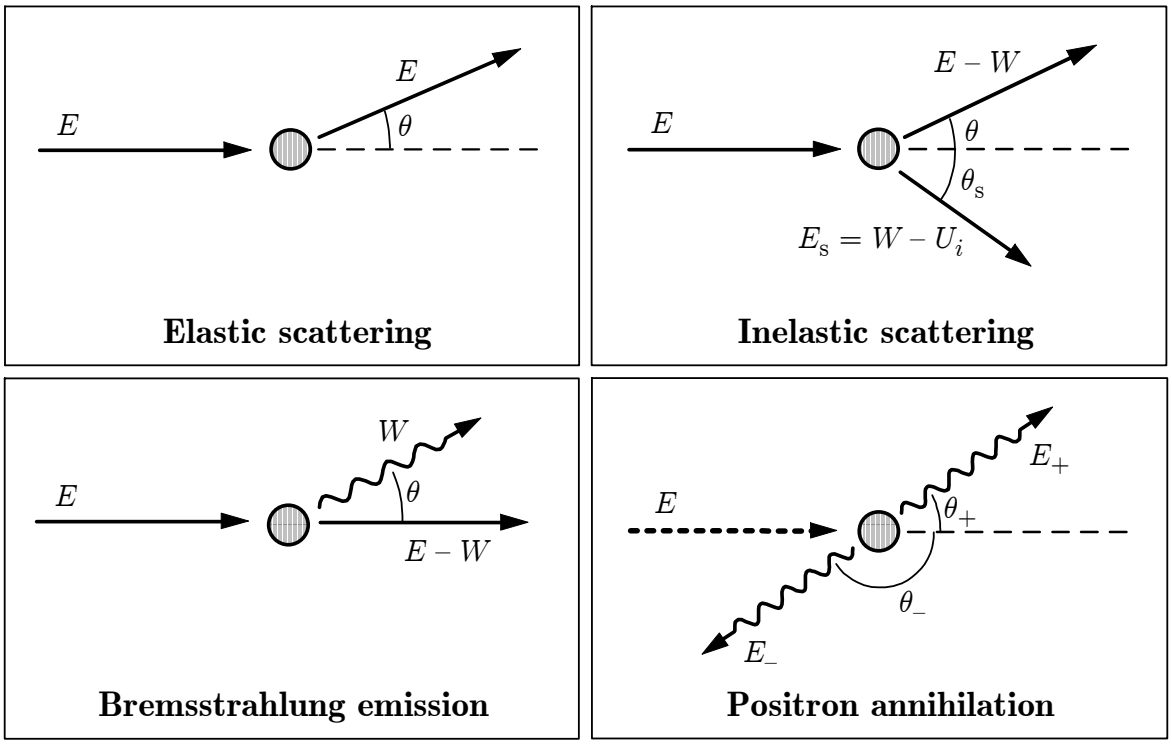

Figure 3.1: Basic interactions of electrons and positrons with matter.

The possible interactions of electrons and positrons with the medium are elastic

scattering, inelastic collisions and bremsstrahlung emission (see Fig. 3.1); positrons can also undergo annihilation, either in flight or at rest. 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 both 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.*, 1993a). 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, 1988a,b,c), ITS3 (Halbleib *et al.*, 1992), EGS4 (Nelson *et al.*, 1985), GEANT3 (Brun *et al.*, 1987), 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.*, 2006). 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 in Appendix A). Because of the large mass of the target ($\sim 3600 Z m_e$), 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 $\rho_{\rm e}(\mathbf{r})$ can be calculated by using available Hartree–Fock codes (*e.g.*, the one of Desclaux, 1975, 1977). 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 $\rho_{\rm e}(r)$. 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)

$$\rho_{\rm n}(r) = \frac{\rho_0}{\exp\left[(r - R_{\rm n})(4\ln 3/t)\right] + 1}, \tag{3.1}$$

where $R_{\rm n}$ is the mean radius (or half-density radius) and t is the surface (or "skin") thickness (defined as the distance over which $\rho_{\rm n}$ drops from ~ 0.9 to ~ 0.1 of its central value). These two parameters are given by

$$R_{\rm n} = 1.07 \times 10^{-15} A_{\rm w}^{1/3}\ {\rm m}, \qquad t = 2.4 \times 10^{-15}\ {\rm m}, \tag{3.2}$$

where $A_{\rm w}$ is the atomic weight of the element. The constant ρ_0, which equals twice the proton density at $r = R_{\rm n}$, is determined by normalisation,

$$Z = 4\pi \int_0^\infty \rho_{\rm n}(r)\, r^2\, {\rm d}r. \tag{3.3}$$

The electrostatic potential of the target atom is

$$\begin{aligned}\varphi(r) \;=\; & e\,4\pi \left[\frac{1}{r}\int_0^r \rho_{\rm n}(r')r'^2\,{\rm d}r' + \int_r^\infty \rho_{\rm n}(r')r'\,{\rm d}r'\right] \\ & - e\,4\pi \left[\frac{1}{r}\int_0^r \rho_{\rm e}(r')r'^2\,{\rm d}r' + \int_r^\infty \rho_{\rm e}(r')r'\,{\rm d}r'\right].\end{aligned} \tag{3.4}$$

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

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

where z_0 is the charge of the projectile in units of e (-1 for electrons, $+1$ for positrons). The term $V_{\rm ex}(r)$, which applies only for electrons, represents a local approximation to the exchange interaction between the projectile and the atomic electrons (Salvat, 1998, 2003). We shall limit our considerations to the case of spin-unpolarised projectiles, *i.e.*, their spin is randomly oriented. The effect of elastic interactions can then 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 $\mathrm{d}\Omega$ about the direction (θ, ϕ) is given by (Walker, 1971; Salvat *et al.*, 2005)

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

where

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

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

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

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)]

$$\mu = (1 - \cos\theta)/2 \tag{3.9}$$

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

$$\frac{\mathrm{d}\sigma_{\mathrm{el}}}{\mathrm{d}\mu} = 4\pi \frac{\mathrm{d}\sigma_{\mathrm{el}}}{\mathrm{d}\Omega}. \tag{3.10}$$

The total elastic cross section is given by

$$\sigma_{\mathrm{el}} = \int \frac{\mathrm{d}\sigma_{\mathrm{el}}}{\mathrm{d}\Omega}\,\mathrm{d}\Omega = \int \frac{\mathrm{d}\sigma_{\mathrm{el}}}{\mathrm{d}\mu}\,\mathrm{d}\mu. \tag{3.11}$$

We can also write

$$\frac{\mathrm{d}\sigma_{\mathrm{el}}}{\mathrm{d}\mu} = \sigma_{\mathrm{el}}\, p_{\mathrm{el}}(\mu), \tag{3.12}$$

where $p_{\mathrm{el}}(\mu)$ 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

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

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

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

$$\sigma_{\mathrm{el},\ell} \equiv \int \left[1 - P_\ell(\cos\theta)\right] \frac{\mathrm{d}\sigma_{\mathrm{el}}}{\mathrm{d}\Omega}\, \mathrm{d}\Omega, \tag{3.14}$$

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

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

The ℓ-th transport mean free path is defined by

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

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

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

and

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

where the notation $\langle\cdots\rangle$ indicates the average value in a single collision. The quantities $\lambda_{\mathrm{el},1}$ and $\lambda_{\mathrm{el},2}$, 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,

$$\lambda_{\mathrm{el},1}^{-1} = \mathcal{N}\sigma_{\mathrm{el},1} = \frac{2}{\lambda_{\mathrm{el}}} \langle\mu\rangle, \tag{3.19}$$

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 $2\lambda_{\mathrm{el},1}^{-1}$ is sometimes called the "scattering power"[1].

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

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, 1977). 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 high energies, ELSEPA computes cross sections for elastic scattering 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) (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 above[2]. 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, $d\sigma_{el}/d\Omega$, total cross sections, σ_{el}, and first and second transport cross sections, $\sigma_{el,1}$ and $\sigma_{el,2}$, 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.p08` and `peldxZZZ.p08`, 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 $\mu \equiv (1 - \cos\theta)/2$. The ELSEPA database is distributed with the ICRU Report 77 (2007), which includes a graphical interface that displays plots of the DCS and the first transport cross section.

Figure 3.2 displays DCSs from the ELSEPA database for elastic scattering of electrons and positrons of various energies by atoms of beryllium, aluminum, rubidium and gold (similar plots for other elements are given in the ICRU Report 77 and by Salvat and Fernández-Varea, 2009). 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, 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 (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, $\sigma_{el,2} \simeq 3\sigma_{el,1}$ ($\lambda_{el,2} \simeq \lambda_{el,1}/3$). The total cross section, $\propto 1/(\rho\lambda_{el})$, 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 $v_{n.r.} = (2E/m_e)^{1/2}$ increases without limit with E and the calculated

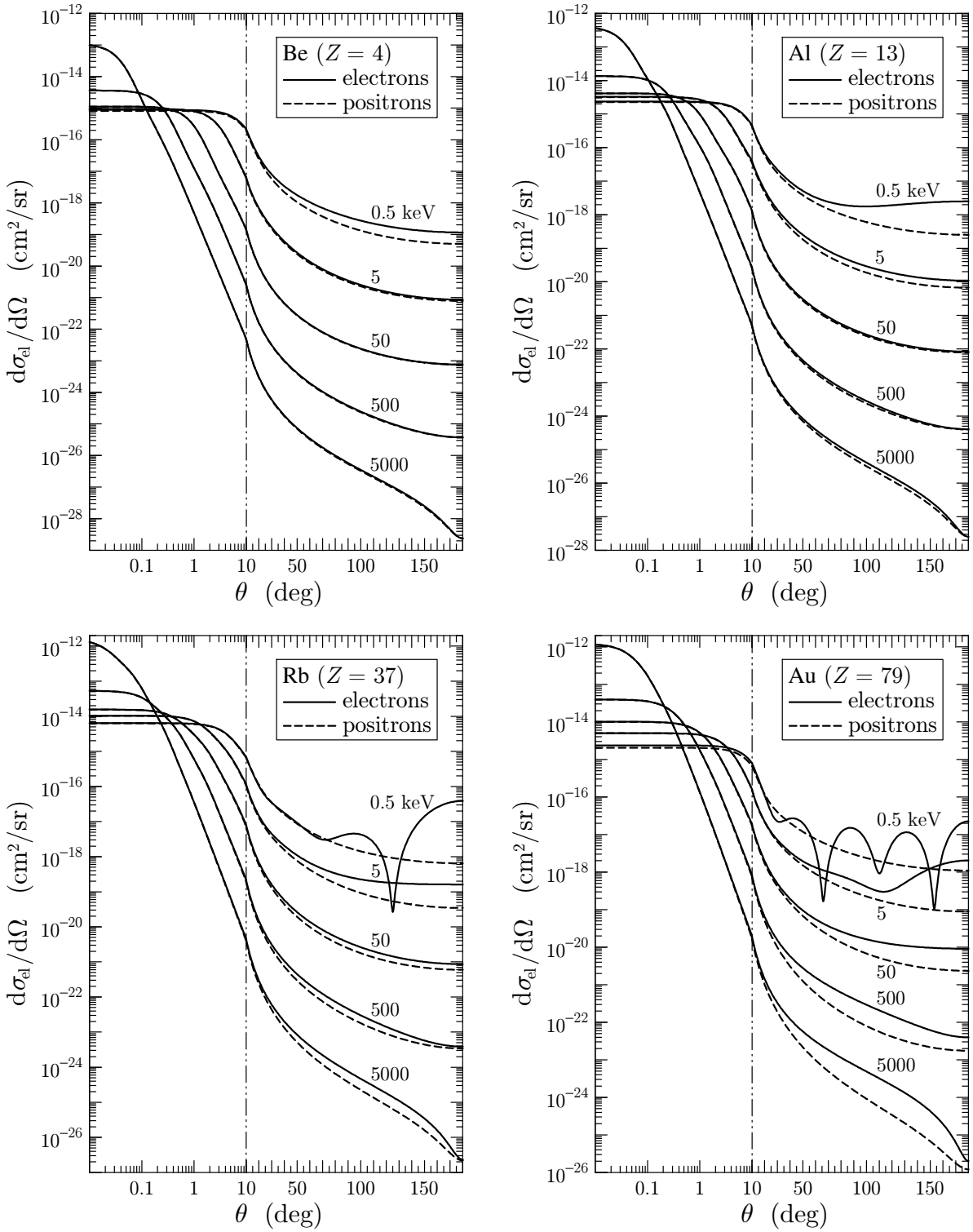

Figure 3.2: DCS for elastic scattering of electrons and positrons by beryllium, aluminum, rubidium and gold atoms as a function of the polar deflection angle θ. Notice the change from logarithmic to linear scale at $\theta = 10$ deg.

non-relativistic total cross section tends to zero at high energies.

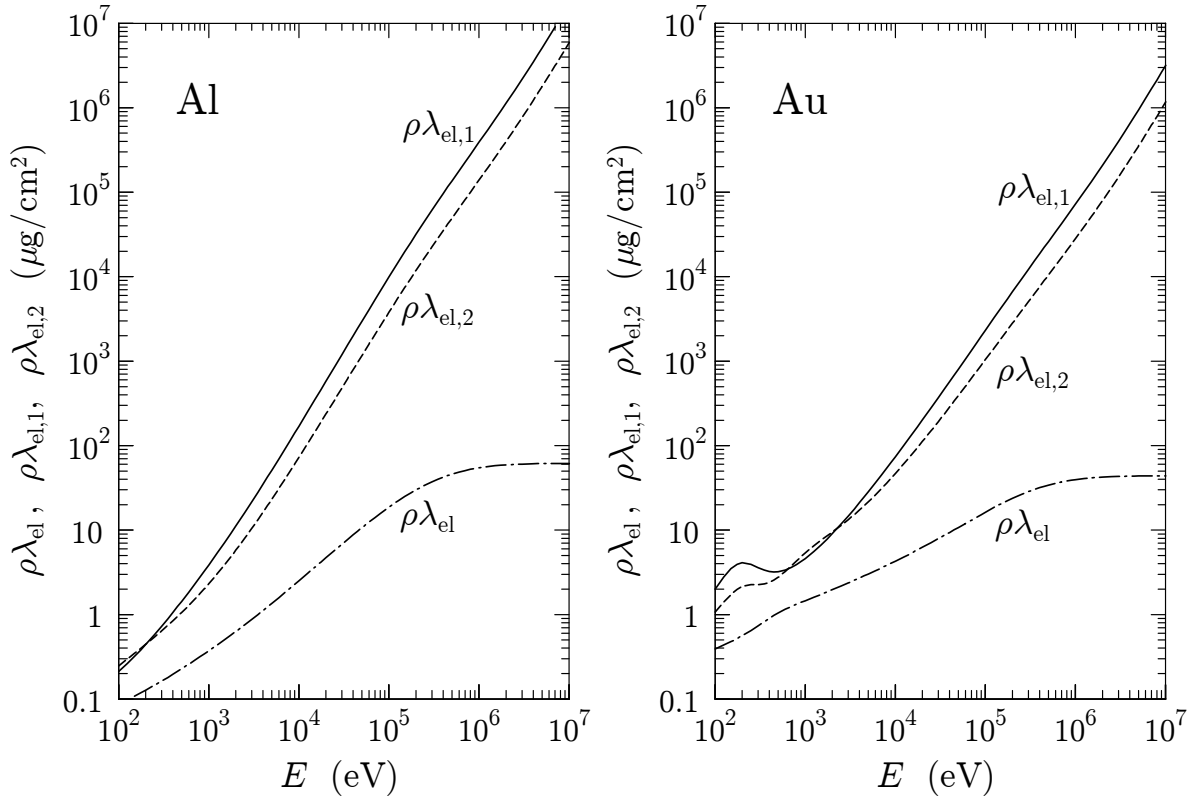

Figure 3.3: Elastic mean free path, $\lambda_{\rm el}$, and first and second transport mean free paths, $\lambda_{\rm el,1}$ and $\lambda_{\rm el,2}$, for electrons scattered in aluminium and gold as functions of the kinetic energy of the projectile. (Adapted from Fernández-Varea *et al.*, 1993a).

In PENELOPE, the DCSs for compounds (and mixtures) are calculated from atomic total and transport cross sections by means of the additivity approximation (incoherent 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 (see Salvat *et al.*, 2005)

$$\frac{{\rm d}\sigma_{\rm el}}{{\rm d}\Omega} = \sum_{i,j} \frac{\sin(q a_{ij}/\hbar)}{q a_{ij}/\hbar} \left[f_i(\theta) f_j^*(\theta) + g_i(\theta) g_j^*(\theta) \right] , \tag{3.20}$$

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 50 eV, 250 eV and 1.25 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.

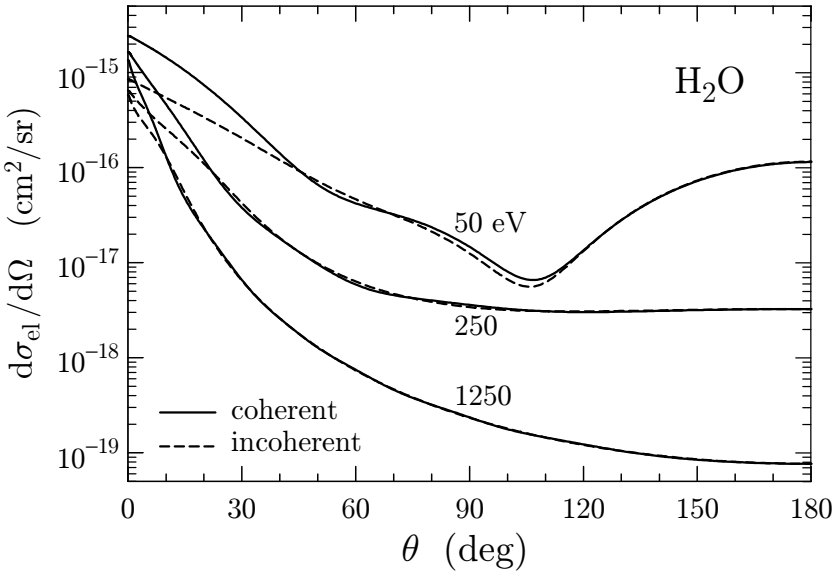

Figure 3.4: DCSs for elastic scattering of electrons by water molecules, calculated as the coherent sum of scattered waves, Eq. (3.20), and from the additivity approximation (incoherent sum).

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; $i = 1, \ldots, 200$), 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,

$$p_{\rm el}(E_i;\mu) = \frac{1}{\sigma_{\rm el}}\,\frac{{\rm d}\sigma_{\rm el}}{{\rm d}\mu} = \frac{4\pi}{\sigma_{\rm el}}\,\frac{{\rm d}\sigma_{\rm el}}{{\rm d}\Omega}, \tag{3.21}$$

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 $p_{\rm int}(E,\mu)$ 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},

$$p_{\rm el,int}(E;\mu) = \pi_i\, p_{\rm el}(E_i;\mu) + \pi_{i+1}\, p_{\rm el}(E_{i+1};\mu) \tag{3.22}$$

with interpolation weights

$$\pi_i = \frac{\ln E_{i+1} - \ln E}{\ln E_{i+1} - \ln E_i} \qquad \text{and} \qquad \pi_{i+1} = \frac{\ln E - \ln E_i}{\ln E_{i+1} - \ln E_i}, \tag{3.23}$$

respectively. Note that these weights are positive and add to unity. Therefore the random sampling of μ from $p_{\rm int}(E;\mu)$ 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 $p_{\rm el}(E_k;\mu)$ 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 only the set of parameters and flags used by the RITA sampling routine for each E_i to be stored in memory.

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

$$\int_0^{\mu} p_{\rm el}(E;\mu')\,{\rm d}\mu' = \xi, \tag{3.24}$$

where the PDF $p_{\rm el}(E;\mu)$ 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 $\lambda_{\rm el}^{(\rm h)}$ between hard elastic events as a function of energy (see Section 4.3). The relation

$$\frac{1}{\lambda_{\rm el}^{(\rm h)}} = \frac{1}{\lambda_{\rm el}} \int_{\mu_c}^{1} p_{\rm el}(E;\mu)\,{\rm d}\mu, \tag{3.25}$$

then determines the cutoff deflection $\mu_c(E)$. We note that

$$\int_0^{\mu_c} p_{\rm el}(E;\mu)\,{\rm d}\mu = 1 - \frac{\lambda_{\rm el}}{\lambda_{\rm el}^{(\rm h)}} \equiv \xi_c. \tag{3.26}$$

Therefore, to generate random values of μ, distributed according to the PDF $p_{\rm el}(E;\mu)$ 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)

$$\int_0^{\mu} p_{\rm el}(E;\mu)\,{\rm d}\mu = \xi_c + \xi(1-\xi_c), \tag{3.27}$$

yields values of μ distributed according to $p_{\rm el}(E;\mu)$ 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 $\sigma_{\rm el}(E)$ and the first and second transport cross sections $\sigma_{\rm el,1}$ and $\sigma_{\rm el,2}$. 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.

The MW DCS is expressed as

$$\frac{\mathrm{d}\sigma_{\mathrm{el}}^{(\mathrm{MW})}}{\mathrm{d}\mu} = \sigma_{\mathrm{el}}\, p_{\mathrm{MW}}(\mu), \tag{3.28}$$

where the single-scattering distribution $p_{\mathrm{MW}}(\mu)$ 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 $p_{\mathrm{MW}}(\mu)$ are equal to those of the actual (partial-wave) DCS:

$$\langle\mu\rangle_{\mathrm{MW}} \equiv \int_0^1 \mu p_{\mathrm{MW}}(\mu)\,\mathrm{d}\mu = \langle\mu\rangle = \frac{1}{2}\frac{\sigma_{\mathrm{el},1}}{\sigma_{\mathrm{el}}} \tag{3.29}$$

and

$$\langle\mu^2\rangle_{\mathrm{MW}} \equiv \int_0^1 \mu^2 p_{\mathrm{MW}}(\mu)\,\mathrm{d}\mu = \langle\mu^2\rangle = \frac{1}{2}\frac{\sigma_{\mathrm{el},1}}{\sigma_{\mathrm{el}}} - \frac{1}{6}\frac{\sigma_{\mathrm{el},2}}{\sigma_{\mathrm{el}}}. \tag{3.30}$$

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 $p_{\mathrm{MW}}(\mu)$.

To set the distribution $p_{\mathrm{MW}}(\mu)$, we start from the Wentzel (1927) angular distribution,

$$p_{\mathrm{W},A_0}(\mu) \equiv \frac{A_0(1+A_0)}{(\mu+A_0)^2}, \qquad (A_0 > 0) \tag{3.31}$$

which describes scattering by an exponentially-screened Coulomb field within the Born approximation (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.*, 1993a). The first moments of the Wentzel distribution are

$$\langle\mu\rangle_{\mathrm{W},A_0} = \int_0^1 \mu\,\frac{A_0(1+A_0)}{(\mu+A_0)^2}\,\mathrm{d}\mu = A_0\left[(1+A_0)\ln\left(\frac{1+A_0}{A_0}\right) - 1\right] \tag{3.32}$$

and

$$\langle\mu^2\rangle_{\mathrm{W},A_0} = \int_0^1 \mu^2\,\frac{A_0(1+A_0)}{(\mu+A_0)^2}\,\mathrm{d}\mu = A_0\left[1 - 2\langle\mu\rangle_{\mathrm{W},A_0}\right]. \tag{3.33}$$

Let us define the value of the screening constant A_0 so that $\langle\mu\rangle_{\mathrm{W},A_0} = \langle\mu\rangle$. 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 $\langle\mu^2\rangle_{\mathrm{W},A_0} \neq \langle\mu^2\rangle$. At low energies, the Wentzel distribution that gives the correct average deflection is too "narrow" [$\langle\mu^2\rangle_{\mathrm{W},A_0} < \langle\mu^2\rangle$ 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 $\langle\mu^2\rangle_{\mathrm{W},A_0} > \langle\mu^2\rangle$ (the Wentzel distribution is too wide), we take $p_{\mathrm{MW}}(\mu)$ as a statistical admixture of the Wentzel distribution and a delta distribution (a zero-width, fixed-scattering-angle process)

$$p_{\mathrm{MW-I}}(\mu) = (1-B)\, p_{\mathrm{W},A}(\mu) + B\, \delta(\mu - \langle\mu\rangle) \tag{3.34}$$

with

$$A = A_0 \qquad \text{and} \qquad B = \frac{\langle\mu^2\rangle_{\mathrm{W},A} - \langle\mu^2\rangle}{\langle\mu^2\rangle_{\mathrm{W},A} - \langle\mu\rangle^2}. \tag{3.35}$$

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 is not visible in the multiple-scattering angular distributions.

• Case II. If $\langle\mu^2\rangle_{\mathrm{W},A_0} < \langle\mu^2\rangle$ (the Wentzel distribution is too narrow), we express $p_{\mathrm{MW}}(\mu)$ 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),

$$p_{\mathrm{MW-II}}(\mu) = (1-B)\, p_{\mathrm{W},A}(\mu) + B\; 8\,(\mu - 1/2)\,\Theta\,(\mu - 1/2)\,. \tag{3.36}$$

The parameters A and B are obtained from the conditions (3.29) and (3.30), which give

$$\begin{aligned} (1-B)\,\langle\mu\rangle_{\mathrm{W},A} + B\,\frac{5}{6} &= \langle\mu\rangle \\ (1-B)\,\langle\mu^2\rangle_{\mathrm{W},A} + B\,\frac{17}{24} &= \langle\mu^2\rangle. \end{aligned} \tag{3.37}$$

From the first of these equations,

$$B = \frac{\langle\mu\rangle - \langle\mu\rangle_{\mathrm{W},A}}{(5/6) - \langle\mu\rangle_{\mathrm{W},A}}. \tag{3.38}$$

Inserting this value in the second of Eqs. (3.37), we obtain

$$\left(\frac{17}{24} - \langle\mu^2\rangle\right)\langle\mu\rangle_{\mathrm{W},A} - \left(\frac{5}{6} - \langle\mu\rangle\right)\langle\mu^2\rangle_{\mathrm{W},A} = \frac{17}{24}\langle\mu\rangle - \frac{5}{6}\langle\mu^2\rangle. \tag{3.39}$$

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 $p_{\rm MW}(\mu)$ 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 $\sigma_{\rm el}$, $\langle\mu\rangle$ and $\langle\mu^2\rangle$.

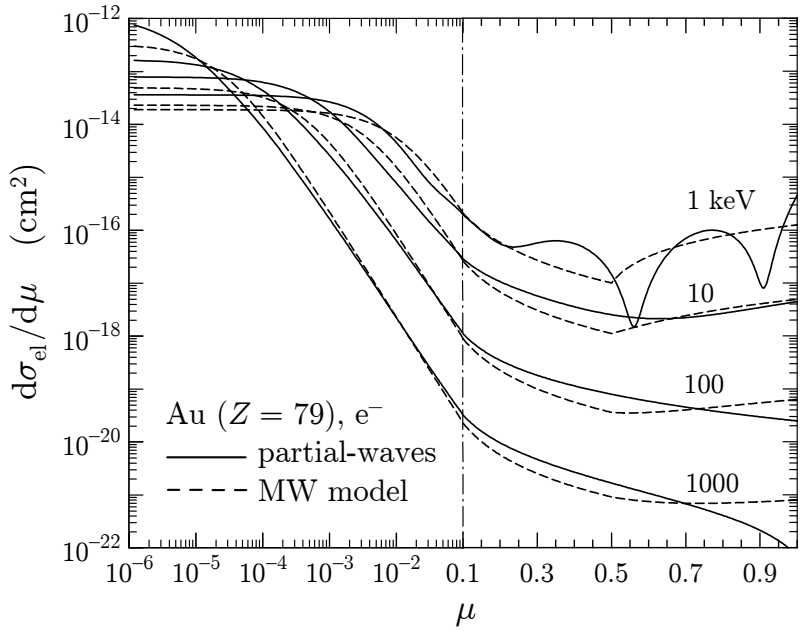

Figure 3.5: Partial-wave and MW model DCSs for elastic scattering of electrons by gold atoms. Notice the change from logarithmic to linear scale at $\mu = 0.1$.

As indicated above, the parameters of the MW model are determined by the characteristic functions $\sigma_{\rm el}(E)$, $\sigma_{\rm el,1}(E)$ and $\sigma_{\rm el,2}(E)$. PENELOPE reads these functions from the MW database files [99 files named `pdeelZZ.p08`, 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 $\langle\mu\rangle_{{\rm W},A}$, 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, the strength of the delta contribution increases rapidly with energy, indicating that the partial-wave DCS is much narrower than the Wentzel distribution.

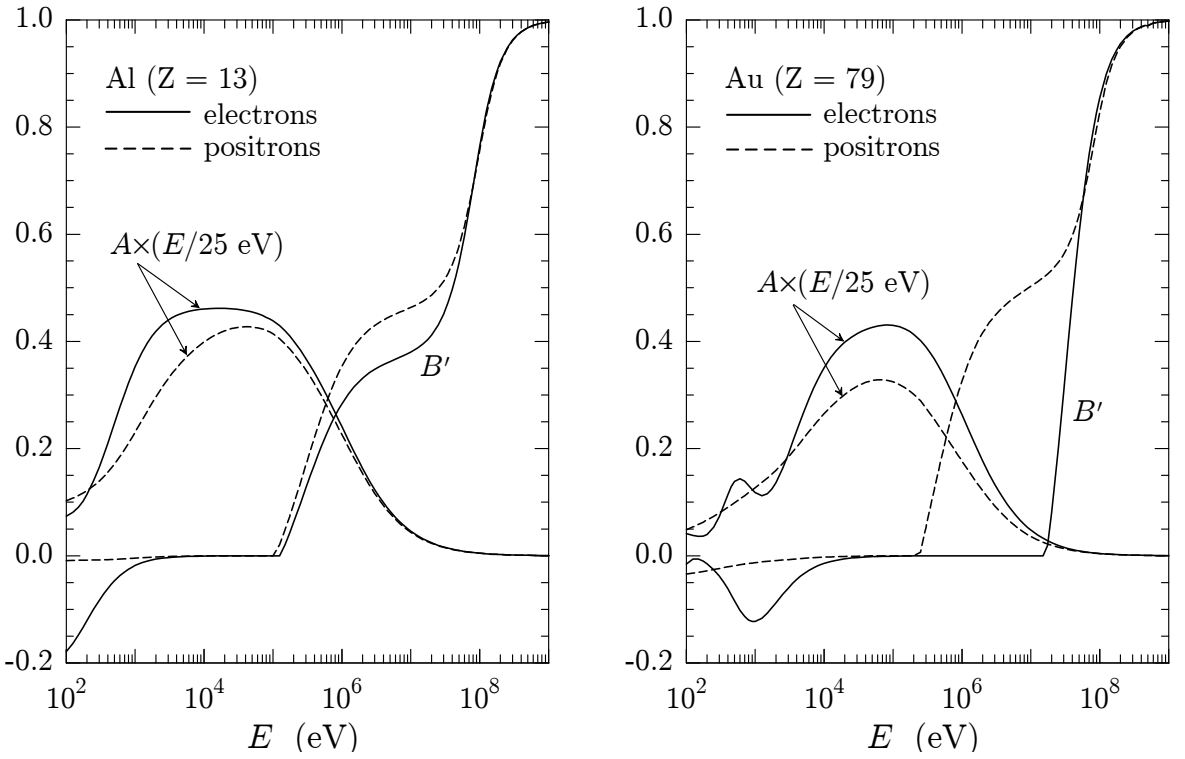

Figure 3.6: Parameters of the MW model for scattering of electrons and positrons by aluminium and gold atoms.

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 $p_{\mathrm{MW-I}}(\mu)$ is

$$\mathcal{P}_{\mathrm{MW-I}}(\mu) \equiv \int_0^{\mu} p_{\mathrm{MW-I}}(\mu')\,\mathrm{d}\mu' = \begin{cases} (1-B)\dfrac{(1+A)\mu}{A+\mu} & \text{if } 0 \le \mu < \langle\mu\rangle, \\ B + (1-B)\dfrac{(1+A)\mu}{A+\mu} & \text{if } \langle\mu\rangle \le \mu \le 1. \end{cases} \tag{3.40}$$

Because of the analytical simplicity of this function, 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

$$\mu = \mathcal{P}^{-1}_{\mathrm{MW-I}}(\xi), \tag{3.41}$$

where $\mathcal{P}^{-1}_{\mathrm{MW-I}}(\xi)$ is the inverse of the cumulative distribution function, which is given by

$$\mathcal{P}^{-1}_{\mathrm{MW-I}}(\xi) = \begin{cases} \dfrac{\xi A}{(1-B)(1+A)-\xi} & \text{if } \ 0 \le \xi < \xi_0, \\ \langle\mu\rangle & \text{if } \ \xi_0 \le \xi < \xi_0 + B, \\ \dfrac{(\xi - B)A}{(1-B)(1+A)-(\xi-B)} & \text{if } \ \xi_0 + B \le \xi \le 1, \end{cases} \tag{3.42}$$

with

$$\xi_0 = (1-B)\frac{(1+A)\langle\mu\rangle}{A+\langle\mu\rangle}. \tag{3.43}$$

To sample μ in the restricted interval (μ_c,1), we can still use the inverse-transform method, Eq. (3.41), but with the random number ξ sampled uniformly in the interval (ξ_c,1) with

$$\xi_\mathrm{c} = \mathcal{P}_{\mathrm{MW-I}}(\mu_\mathrm{c}). \tag{3.44}$$

• Case II. The cumulative distribution function is

$$\begin{aligned} \mathcal{P}_{\mathrm{MW-II}}(\mu) \ &\equiv \ \int_0^\mu p_{\mathrm{MW-II}}(\mu')\,\mathrm{d}\mu' \\ &= \begin{cases} (1-B)\dfrac{(1+A)\mu}{A+\mu} & \text{if } 0 \le \mu < \frac{1}{2}, \\ (1-B)\dfrac{(1+A)\mu}{A+\mu} + B\,4\left[\mu^2 - \mu + \dfrac{1}{4}\right] & \text{if } \frac{1}{2} \le \mu \le 1. \end{cases} \end{aligned} \tag{3.45}$$

In principle, to sample μ in (0,1), we can adopt the inverse-transform method. The sampling equation

$$\xi = \mathcal{P}_{\mathrm{MW-II}}(\mu) \tag{3.46}$$

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 (μ_c,1) we use the composition method, which is easier than solving Eq. (3.46). Notice that 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. Extension of the theory to inelastic collisions in condensed materials has been discussed by Fano (1963). Formal aspects of quantum theory for condensed matter are quite complicated. Fortunately, the results are essentially equivalent to those from classical dielectric theory (Lindhard, 1954; Fernández-Varea *et al.*, 2005).

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

$$Q(Q + 2\mathrm{m_e}c^2) = (cq)^2. \tag{3.47}$$

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)

$$\frac{\mathrm{d}^2\sigma_{\mathrm{in}}}{\mathrm{d}W\,\mathrm{d}Q} = \frac{2\pi z_0^2 e^4}{\mathrm{m_e}v^2}\left(\frac{2\mathrm{m_e}c^2}{WQ(Q+2\mathrm{m_e}c^2)} + \frac{\beta^2\sin^2\theta_{\mathrm{r}} W 2\mathrm{m_e}c^2}{[Q(Q+2\mathrm{m_e}c^2)-W^2]^2}\right)\frac{\mathrm{d}f(Q,W)}{\mathrm{d}W}, \tag{3.48}$$

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),

$$\cos^2\theta_{\mathrm{r}} = \frac{W^2/\beta^2}{Q(Q+2\mathrm{m_e}c^2)}\left(1 + \frac{Q(Q+2\mathrm{m_e}c^2)-W^2}{2W(E+\mathrm{m_e}c^2)}\right)^2. \tag{3.49}$$

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 $\mathrm{d}f(Q,W)/\mathrm{d}W$ 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, (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 (Manson, 1972; Bote and Salvat, 2008). However, 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 Salvat and Fernández-Varea, 1992).

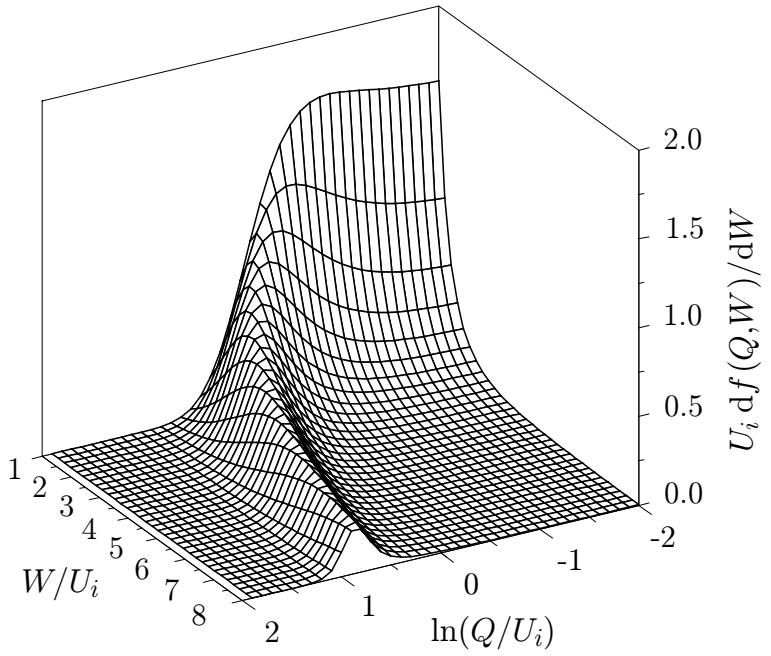

Figure 3.7: The GOS for ionisation of the hydrogen atom ($Z = 1$) in the ground state. All energies are in units of the ionisation energy $U_i = 13.6$ eV. The GOS for ionisation of (non-relativistic) hydrogenic ions is independent of Z if energies are expressed in units of the ionisation energy. (From Fernández-Varea *et al.*, 2005).

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 polarises 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.*, results from the former agree with those from quantum calculations for a low-density medium) if one assumes the identity

$$\frac{\mathrm{d}f(Q,W)}{\mathrm{d}W} \equiv W\frac{Q+\mathrm{m_e}c^2}{\mathrm{m_e}c^2}\frac{2Z}{\pi\Omega_\mathrm{p}^2}\,\mathrm{Im}\left(\frac{-1}{\epsilon(Q,W)}\right), \tag{3.50}$$

where Ω_p is the plasma energy of a free-electron gas with the electron density of the medium, given by

$$\Omega_\mathrm{p}^2 = 4\pi\mathcal{N}Z\hbar^2e^2/\mathrm{m_e}. \tag{3.51}$$

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)],

$$\frac{\mathrm{d}^2\sigma_\mathrm{in}}{\mathrm{d}W\,\mathrm{d}Q} = \frac{2\pi z_0^2e^4}{\mathrm{m_e}v^2}\frac{\mathrm{d}f(Q,W)}{\mathrm{d}W}\left(\frac{2\mathrm{m_e}c^2}{WQ(Q+2\mathrm{m_e}c^2)} + \left\{\frac{\beta^2\sin^2\theta_\mathrm{r}W2\mathrm{m_e}c^2}{[Q(Q+2\mathrm{m_e}c^2)-W^2]^2} - \mathcal{D}(Q,W)\right\}\right), \tag{3.52}$$

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 $Q \to 0$, the GOS reduces to the optical oscillator strength (OOS),

$$\frac{\mathrm{d}f(W)}{\mathrm{d}W} \equiv \frac{\mathrm{d}f(Q=0,W)}{\mathrm{d}W}, \tag{3.53}$$

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 Fernández-Varea *et al.*, 1993b, and references therein). The GOS satisfies the Bethe sum rule (Inokuti, 1971)

$$\int_0^\infty \frac{\mathrm{d}f(Q,W)}{\mathrm{d}W}\,\mathrm{d}W = Z \qquad \text{for any } Q. \tag{3.54}$$

This sum rule, which is a result from non-relativistic theory (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)

$$Z\ln I = \int_0^\infty \ln W\,\frac{\mathrm{d}f(W)}{\mathrm{d}W}\,\mathrm{d}W, \tag{3.55}$$

plays a central role in the Bethe stopping power formula [Eq. (3.120)]. This quantity has been determined empirically for a large number of materials (see Berger and Seltzer, 1982; ICRU Report 37, 1984) 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

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 (Salvat and Fernández-Varea, 1992). 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 on 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)

$$F_k(Q,W) = \delta(W-W_k)\Theta(Q_k-Q) + \delta(W-Q)\Theta(Q-Q_k), \tag{3.56}$$

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 ($W = Q$). Note that close interactions are allowed for energy transfers W larger than Q_k; for bound shells, we set $Q_k = U_k$ (Fig. 3.8a). It is worth mentioning that the definition (3.56) of the δ-oscillator differs from the one used in previous versions of PENELOPE, which allowed only excitations with $W > W_k$ [usually, W_k is larger than U_k, see Eq. (3.63) below]. Notice that the oscillator GOS satisfies the sum rule

$$\int_0^\infty F_k(Q,W)\,\mathrm{d}W = 1 \qquad \text{for any } Q \tag{3.57}$$

and, consequently, a δ-oscillator corresponds to one electron in the target. The Liljequist GOS model for the whole atom is given by

$$\frac{\mathrm{d}f(Q,W)}{\mathrm{d}W} = \sum_k f_k \left[\delta(W-W_k)\Theta(Q_k-Q) + \delta(W-Q)\Theta(Q-Q_k)\right]. \tag{3.58}$$

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

$$\frac{\mathrm{d}f(W)}{\mathrm{d}W} = \sum_k f_k\,\delta(W - W_k), \tag{3.59}$$

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; ICRU Report 37, 1984), the oscillator strengths must satisfy the Bethe sum rule (3.54),

$$\sum\nolimits_k f_k = Z, \tag{3.60}$$

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,

$$\sum\nolimits_k f_k \ln W_k = Z \ln I. \tag{3.61}$$

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.

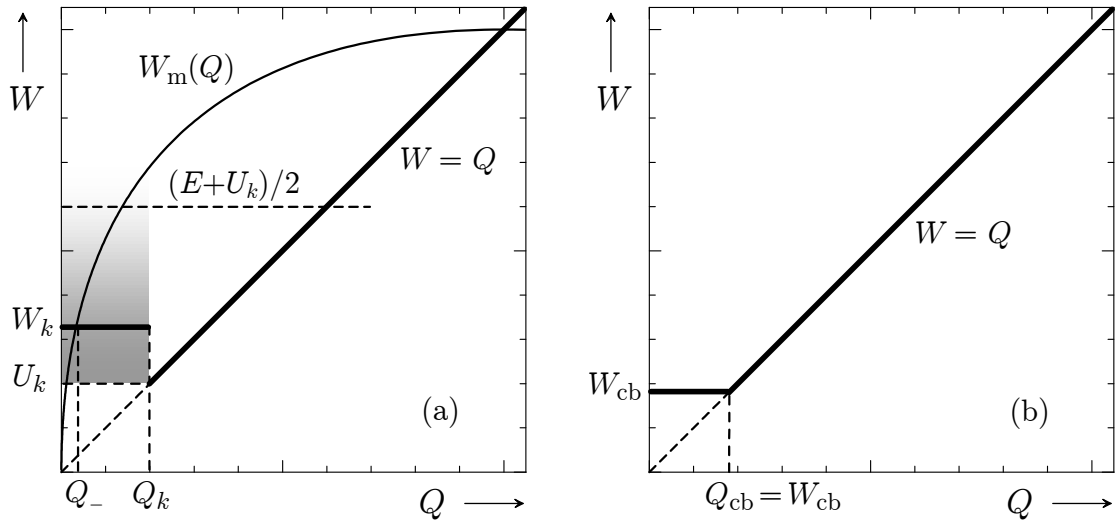

Figure 3.8: (a) Oscillator model for the GOS of an inner shell with $U_k = 2$ keV. The continuous curve represents the maximum allowed energy loss as a function of the recoil energy, $W_\mathrm{m}(Q)$, for electrons/positrons with $E = 10$ keV. For distant interactions the possible recoil energies lie in the interval from Q_- to Q_k. Recoil energies larger than Q_k correspond to close interactions. The largest allowed energy loss W_max is $(E + U_k)/2$ for electrons and E for positrons (see text). (b) Oscillator-GOS model for excitations of the conduction band of conductors.

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, $U_{\rm cb} = 0$. Excitations of the conduction band will be described by a single oscillator, with oscillator strength $f_{\rm cb}$, resonance energy $W_{\rm cb}$ and cutoff recoil energy $Q_{\rm cb} = W_{\rm cb}$ (Fig. 3.8b). The parameters $f_{\rm cb}$ and $W_{\rm cb}$ 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 $f_{\rm cb}$ (as the number of electrons with ionisation energies less than, say, 15 eV) and set the resonance energy $W_{\rm cb}$ equal to the plasmon energy of a free-electron gas with the same density as that of conduction electrons,

$$W_{\rm cb} = \sqrt{4\pi\mathcal{N} f_{\rm cb}\hbar^2 e^2/{\rm m_e}} = \sqrt{\frac{f_{\rm cb}}{Z}}\,\Omega_{\rm p}. \tag{3.62}$$

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 Kittel, 1976). A similar approach, with $f_{\rm cb}$ set equal to the lowest chemical valence of an element, was adopted by Sternheimer *et al.* (1982, 1983, 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

$$W_k = \sqrt{(aU_k)^2 + \frac{2}{3}\,\frac{f_k}{Z}\Omega_{\rm p}^2}, \tag{3.63}$$

where U_k is the ionisation energy and $\Omega_{\rm p}$ is the plasma energy corresponding to the total electron density in the material, Eq. (3.51). The term $2f_k\Omega_{\rm p}^2/3Z$ 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

$$Z\ln I = f_{\rm cb}\ln W_{\rm cb} + \sum_k f_k \ln\sqrt{(aU_k)^2 + \frac{2}{3}\,\frac{f_k}{Z}\,\Omega_{\rm p}^2}. \tag{3.64}$$

For a one-shell system, such as the hydrogen atom, relations (3.60) and (3.61) imply that the resonance energy W_k 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_k and the number of electrons Z_k 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 shell ionisation energies given by Carlson (1975) are used.

The adopted I-values are those proposed by Berger and Seltzer (1982), which were also used to generate the ICRU Report 37 (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 neglected. 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 $m \gg \mathrm{m_e}$. 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 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 global transport properties. In what follows, K, L and M shells with ionisation energies U_k larger than the absorption energies of electrons or photons, `EABS(1)` or `EABS(2)` (see Section 7.1.2), will be referred to as "inner" shells. Electron shells other than K, L and M shells, or with $U_k < \min($`EABS(1)`, `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 $W < W_{\mathrm{cc}}$) 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 $\mathrm{X_xY_y}$, 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 $Z_{\mathrm{M}} = \mathrm{x}Z_{\mathrm{X}} + \mathrm{y}Z_{\mathrm{Y}}$, 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

$$Z_{\mathrm{M}} \ln I = \mathrm{x}Z_{\mathrm{X}} \ln I_{\mathrm{X}} + \mathrm{y}Z_{\mathrm{Y}} \ln I_{\mathrm{Y}}, \tag{3.65}$$

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,

$$\frac{\mathrm{d}^2\sigma_{\mathrm{in}}}{\mathrm{d}W\,\mathrm{d}Q} = \frac{\mathrm{d}^2\sigma_{\mathrm{dis,l}}}{\mathrm{d}W\,\mathrm{d}Q} + \frac{\mathrm{d}^2\sigma_{\mathrm{dis,t}}}{\mathrm{d}W\,\mathrm{d}Q} + \frac{\mathrm{d}^2\sigma_{\mathrm{clo}}}{\mathrm{d}W\,\mathrm{d}Q}. \tag{3.66}$$

The DCS for distant longitudinal interactions is given by the first term in Eq. (3.52),

$$\frac{\mathrm{d}^2\sigma_{\mathrm{dis,l}}}{\mathrm{d}W\,\mathrm{d}Q} = \frac{2\pi e^4}{\mathrm{m_e}v^2}\sum_k f_k \frac{1}{W}\,\frac{2\mathrm{m_e}c^2}{Q(Q+2\mathrm{m_e}c^2)}\,\delta(W-W_k)\,\Theta(Q_k-Q). \tag{3.67}$$

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,

$$\begin{aligned}\frac{\mathrm{d}^2\sigma_{\mathrm{dis,t}}}{\mathrm{d}W\,\mathrm{d}Q} \;=\; & \frac{2\pi e^4}{\mathrm{m_e}v^2}\sum_k f_k \frac{1}{W}\left\{\ln\left(\frac{1}{1-\beta^2}\right) - \beta^2 - \delta_{\mathrm{F}}\right\} \\ & \times \delta(W-W_k)\,\Theta(Q_k-Q)\,\delta(Q-Q_-),\end{aligned} \tag{3.68}$$

where Q_- is the minimum recoil energy[3] 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)

$$\delta_{\mathrm{F}} \equiv \frac{1}{Z}\int_0^\infty \frac{\mathrm{d}f(Q=0,W)}{\mathrm{d}W}\ln\left(1+\frac{L^2}{W^2}\right)\mathrm{d}W - \frac{L^2}{\Omega_{\mathrm{p}}^2}\left(1-\beta^2\right), \tag{3.69}$$

[3]The recoil energy Q_- corresponds to $\theta = 0$, *i.e.*, we consider that the projectile is not deflected by distant transverse interactions.

where L is a real-valued function of β^2 defined as the positive root of the following equation (Inokuti and Smith, 1982):

$$\mathcal{F}(L) \equiv \frac{1}{Z}\Omega_{\rm p}^2 \int_0^\infty \frac{1}{W^2+L^2}\frac{{\rm d}f(Q=0,W)}{{\rm d}W}\,{\rm d}W = 1-\beta^2. \tag{3.70}$$

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 $\delta_{\rm F}=0$. Therefore, the function $L(\beta^2)$ starts 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

$$\mathcal{F}(L) = \frac{1}{Z}\Omega_{\rm p}^2 \sum_k \frac{f_k}{W_k^2+L^2} \tag{3.71}$$

and

$$\delta_{\rm F} \equiv \frac{1}{Z}\sum_k f_k \ln\left(1+\frac{L^2}{W_k^2}\right) - \frac{L^2}{\Omega_{\rm p}^2}\left(1-\beta^2\right). \tag{3.72}$$

In the high-energy limit ($\beta \to 1$), the L value resulting from Eq. (3.70) is large ($L \gg W_k$) and can be approximated as $L^2 = \Omega_{\rm p}^2/(1-\beta^2)$. Then, using the Bethe sum rule ($\sum f_k = Z$) and the relation (3.61), we obtain

$$\delta_{\rm F} \simeq \ln\left(\frac{\Omega_{\rm p}^2}{(1-\beta^2)I^2}\right) - 1, \qquad \text{when } \beta \to 1. \tag{3.73}$$

The DCS for close collisions is given by

$$\frac{{\rm d}^2\sigma_{\rm clo}}{{\rm d}W\,{\rm d}Q} = \frac{2\pi e^4}{{\rm m_e}v^2}\sum_k f_k \frac{1}{W}\left(\frac{2{\rm m_e}c^2}{W(W+2{\rm m_e}c^2)} + \frac{\beta^2\sin^2\theta_{\rm clo}}{2{\rm m_e}c^2}\right)\delta(W-Q)\,\Theta(Q-Q_k),$$

where $\theta_{\rm clo}$ is the recoil angle, defined by Eq. (3.49) with $Q=W$,

$$\cos^2\theta_{\rm clo} = \frac{W}{E}\,\frac{E+2{\rm m_e}c^2}{W+2{\rm m_e}c^2}. \tag{3.74}$$

It is worth mentioning that the formula (3.48) was derived by Fano (1963) under the assumption that the projectile is a charged particle with a mass much larger than that of the electron. When this assumption is not applied (see Bote and Salvat, 2008), the DCS for close collisions takes the following form

$$\frac{{\rm d}^2\sigma_{\rm clo}}{{\rm d}W\,{\rm d}Q} = \frac{2\pi e^4}{{\rm m_e}v^2}\sum_k f_k \frac{1}{W^2}\left(1-\frac{(2E-W+4{\rm m_e}c^2)W}{2(E+{\rm m_e}c^2)^2}\right)\delta(W-Q)\,\Theta(Q-Q_k). \tag{3.75}$$

3.2.2.1 Distant interactions with inner-shell electrons

The formulas (3.67) and (3.68) provide quite a realistic description of distant interactions with electrons in the conduction band of certain solids, such as aluminium and silicon, which behave as a nearly-free-electron gas with a characteristic plasmon-excitation line. In general, for real media, distant excitations of weakly-bound electrons have a continuous energy spectrum that can be described approximately by using several δ-oscillators of the "conduction-band" type (right diagram in Fig. 3.8) with oscillator strengths f_k and resonance energies W_k selected so as to mimic the continuous OOS of the medium (Fernández-Varea *et al.*, 1993b). If the spacing between resonance energies is small enough, the energy-loss distributions obtained from the simulation have realistic appearances.

In the case of inner-shells, however, the energy loss distribution associated with single distant interactions with a given atomic electron shell is approximated as a single resonance (a δ-distribution), while the actual distribution is continuous for energy losses above the ionisation threshold. As a consequence, energy loss spectra simulated from the DCSs (3.67) and (3.68) show unphysical narrow peaks at energy losses that are multiples of the resonance energies. To get rid of this kind of artefact, we will spread the resonance line by sampling the energy loss in distant interactions with inner-shell electrons from a continuous distribution $p_{\rm dis}(W)$ that has the correct average value, $\langle W\rangle = W_k$. In principle, this distribution can be obtained by considering a realistic GOS for inner-shell excitations. Since such an approach is too elaborate for our purposes, we will adopt a simple distribution that differs from zero within a finite energy-loss interval extending from the ionisation threshold, U_k, up to a maximum value $W_{\rm dis}$. The shaded area (for $Q < Q_k$) in Fig. 3.8a represents the distribution $p_{\rm dis}(W)$, which replaces the delta function $\delta(W - W_k)$.

To facilitate random sampling, we use a "triangle" distribution,

$$p_{\rm dis}(W) = \begin{cases} \dfrac{2}{(W_{\rm dis}-U_k)^2}\,(W_{\rm dis}-W) & \text{if } U_k \le W < W_{\rm dis}, \\ 0 & \text{otherwise.} \end{cases} \tag{3.76}$$

This distribution is convenient because it is determined by only the maximum energy loss $W_{\rm dis}$ and, moreover, it allows all calculations to be performed analytically. Qualitative considerations based on the dipole approximation indicate that the maximum energy loss $W_{\rm dis}$ is nearly independent of the kinetic energy of the projectile when the latter is much higher than U_k. The requirement

$$\langle W\rangle = \int_0^\infty W p_{\rm dis}(W)\,{\rm d}W = W_k$$

implies that

$$W_{\rm dis} = 3W_k - 2U_k. \tag{3.77}$$

Evidently, when E approaches the ionisation threshold, $W_{\rm dis}$ should decrease, because energy losses larger than E are forbidden. For the sake of consistency, when E decreases

below the value (3.77) we will re-define the resonance energy W_k to ensure that the DCSs vary smoothly with E. We set

$$W'_k = \begin{cases} W_k & \text{if } E > 3W_k - 2U_k, \\ (E + 2U_k)/3 & \text{otherwise,} \end{cases} \tag{3.78}$$

and, consistently, we replace the resonance energy by this modified value in all formulas pertaining to the distant excitations of inner shells. Thus, the maximum allowed energy loss in these excitations, Eq. (3.77), is given by

$$W_{\rm dis} = 3W'_k - 2U_k, \tag{3.79}$$

which never exceeds E. Additionally, to get total cross sections with the approximate threshold behaviour predicted by the (plane-wave) Born approximation (see Section 3.2.6), we also replace the cutoff recoil energy Q_k, which for inner shells is equal to U_k, by the modified value

$$Q'_k = \begin{cases} U_k & \text{if } E > 3W_k - 2U_k, \\ U_k \dfrac{E}{3W_k - 2U_k} & \text{otherwise.} \end{cases} \tag{3.80}$$

This replacement lowers an excessively high peak of the total cross section at energies E slightly above U_k, which is caused by the downward shift of the resonance energy W'_k (see Fig. 3.9 below).

The DCSs for distant longitudinal and transverse excitations of an inner shell are obtained from expressions (3.67) and (3.68) by replacing the factor $\delta(W - W_k)$ by the distribution $p_{\rm dis}(W)$. That is,

$$\frac{{\rm d}^2\sigma_{\rm dis,l}}{{\rm d}W\,{\rm d}Q} = \frac{2\pi e^4}{{\rm m_e}v^2} \sum_k f_k \frac{1}{W_k}\, p_{\rm dis}(W) \frac{2{\rm m_e}c^2}{Q(Q + 2{\rm m_e}c^2)}\, \Theta(Q'_k - Q) \tag{3.81}$$

and

$$\begin{aligned} \frac{{\rm d}^2\sigma_{\rm dis,t}}{{\rm d}W\,{\rm d}Q} \;=\; & \frac{2\pi e^4}{{\rm m_e}v^2} \sum_k f_k \frac{1}{W_k}\, p_{\rm dis}(W) \left\{ \ln\left(\frac{1}{1-\beta^2}\right) - \beta^2 - \delta_{\rm F} \right\} \\ & \times \Theta(Q'_k - Q)\, \delta(Q - Q_-). \end{aligned} \tag{3.82}$$

At this point, it is convenient to mention that the δ-oscillator model provides a consistent description of angular deflections in inelastic collisions (Negreanu *et al.*, 2005), and therefore, it is convenient to sample the scattering angle θ according to this model. Consequently, in the above equations the minimum recoil energy Q_-, which defines the angular distribution [see Eqs. (3.126) and (3.129) below], will be set equal to the value corresponding to the modified resonance energy W'_k, *i.e.*,

$$Q_- = \sqrt{\left[\sqrt{E(E + 2{\rm m_e}c^2)} - \sqrt{(E - W'_k)(E - W'_k + 2{\rm m_e}c^2)}\right]^2 + {\rm m_e^2}c^4} - {\rm m_e}c^2\,. \tag{3.83}$$

This *ad hoc* modification has the further advantage of simplifying the calculation of integrals of the restricted angular DCSs (see Section 4.3.2).

3.2.2.2 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,

$$\frac{\mathrm{d}^2\sigma_{\mathrm{M}}}{\mathrm{d}W\,\mathrm{d}Q} = \frac{2\pi e^4}{\mathrm{m_e}v^2}\frac{1}{W^2}\left[1+\left(\frac{W}{E-W}\right)^2-\frac{W}{E-W}\right.$$
$$\left.+a\left(\frac{W}{E-W}+\frac{W^2}{E^2}\right)\right]\delta(W-Q), \tag{3.84}$$

where

$$a=\left(\frac{E}{E+\mathrm{m_e}c^2}\right)^2=\left(\frac{\gamma-1}{\gamma}\right)^2. \tag{3.85}$$

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

$$\frac{\mathrm{d}^2\sigma^{(-)}_{\mathrm{clo}}}{\mathrm{d}W\,\mathrm{d}Q}=\frac{2\pi e^4}{\mathrm{m_e}v^2}\sum_k f_k\frac{1}{W^2}F^{(-)}(E,W)\delta(W-Q)\,\Theta(Q-Q'_k), \tag{3.86}$$

with

$$F^{(-)}(E,W)\equiv 1+\left(\frac{W}{E'-W}\right)^2-\frac{W}{E'-W}+a\left(\frac{W}{E'-W}+\frac{W^2}{E'^2}\right). \tag{3.87}$$

For collisions with free electrons at rest, we should take $E'=E$ [cf. Eq. (3.84)]. However, in close collisions with bound electrons, the wave function of the projectile is distorted by the electrostatic field of the target atom, which produces an increase in the effective kinetic energy of the projectile. To account partially for this "Coulomb effect", the factor $F^{(-)}(E,W)$ of each term on the right-hand side of Eq. (3.86) is calculated with the energy $E'=E+U_k$. The replacement of E by the effective kinetic energy E' yields cross sections in better agreement with results from experimental measurements and from more elaborate calculations (see Fig. 3.9 below). On the other hand, in the final state we have two indistinguishable free electrons, and it is natural to consider the fastest one as being the "primary". Accordingly, the maximum allowed energy transfer in collisions (close and distant) with electrons in the k-th shell is set equal to

$$W_{\max}=E'/2=(E+U_k)/2. \tag{3.88}$$

3.2.2.3 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 ($> 2\mathrm{m_e}c^2$) is available (either from real or virtual photons). A positron does not interact with matter as a typical (stable) positively charged particle, since the competing process of annihilation followed by re-creation can cause the same transitions as "direct" scattering (see 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,

$$\frac{\mathrm{d}^2\sigma_\mathrm{B}}{\mathrm{d}W\mathrm{d}Q} = \frac{2\pi e^4}{\mathrm{m_e}v^2}\frac{1}{W^2}\left[1 - b_1\frac{W}{E} + b_2\left(\frac{W}{E}\right)^2 - b_3\left(\frac{W}{E}\right)^3 + b_4\left(\frac{W}{E}\right)^4\right]\delta(W-Q), \tag{3.89}$$

where

$$\begin{aligned} b_1 &= \left(\frac{\gamma-1}{\gamma}\right)^2\frac{2(\gamma+1)^2-1}{\gamma^2-1}, \qquad b_2 = \left(\frac{\gamma-1}{\gamma}\right)^2\frac{3(\gamma+1)^2+1}{(\gamma+1)^2}, \\ b_3 &= \left(\frac{\gamma-1}{\gamma}\right)^2\frac{2\gamma(\gamma-1)}{(\gamma+1)^2}, \qquad b_4 = \left(\frac{\gamma-1}{\gamma}\right)^2\frac{(\gamma-1)^2}{(\gamma+1)^2}. \end{aligned} \tag{3.90}$$

To approximately account 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,

$$F^{(+)}(E,W) = 1 - b_1\frac{W}{E} + b_2\left(\frac{W}{E}\right)^2 - b_3\left(\frac{W}{E}\right)^3 + b_4\left(\frac{W}{E}\right)^4. \tag{3.91}$$

That is,

$$\frac{\mathrm{d}^2\sigma^{(+)}_\mathrm{clo}}{\mathrm{d}W\,\mathrm{d}Q} = \frac{2\pi e^4}{\mathrm{m_e}v^2}\sum_k f_k\frac{1}{W^2}F^{(+)}(E,W)\delta(W-Q)\,\Theta(Q-Q'_k). \tag{3.92}$$

Notice that the maximum energy loss in collisions of positrons with energy E is $W_\mathrm{max} = E$.

3.2.3 Integrated cross sections

The energy-loss DCS is defined as

$$\frac{\mathrm{d}\sigma_\mathrm{in}}{\mathrm{d}W} \equiv \int_{Q_-}^{Q_+}\frac{\mathrm{d}^2\sigma_\mathrm{in}}{\mathrm{d}W\,\mathrm{d}Q}\,\mathrm{d}Q = \frac{\mathrm{d}\sigma_\mathrm{dis,l}}{\mathrm{d}W} + \frac{\mathrm{d}\sigma_\mathrm{dis,t}}{\mathrm{d}W} + \frac{\mathrm{d}\sigma_\mathrm{clo}}{\mathrm{d}W}, \tag{3.93}$$

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

$$\frac{\mathrm{d}\sigma_\mathrm{dis,l}}{\mathrm{d}W} = \frac{2\pi e^4}{\mathrm{m_e}v^2}\sum_k f_k\frac{1}{W}\,p_\mathrm{dis}(W)\ln\left(\frac{Q'_k}{Q_-}\,\frac{Q_- + 2\mathrm{m_e}c^2}{Q'_k + 2\mathrm{m_e}c^2}\right)\Theta(Q'_k - Q_-) \tag{3.94}$$

and

$$\frac{\mathrm{d}\sigma_{\mathrm{dis,t}}}{\mathrm{d}W} = \frac{2\pi e^4}{\mathrm{m_e}v^2}\sum_k f_k \frac{1}{W}\, p_{\mathrm{dis}}(W)\left\{\ln\left(\frac{1}{1-\beta^2}\right) - \beta^2 - \delta_{\mathrm{F}}\right\}\Theta(Q'_k - Q_-), \tag{3.95}$$

respectively. Note that hereafter we express these DCSs by assuming that all terms correspond to inner shells; in the case of the conduction band the distribution $p_{\mathrm{dis}}(W)$ and the cutoff recoil energy Q'_k should be replaced by $\delta(W - W_k)$ and Q_k, respectively. The energy-loss DCS for close collisions is

$$\frac{\mathrm{d}\sigma^{(\pm)}_{\mathrm{clo}}}{\mathrm{d}W} = \frac{2\pi e^4}{\mathrm{m_e}v^2}\sum_k f_k \frac{1}{W^2} F^{(\pm)}(E, W)\,\Theta(W - Q_k). \tag{3.96}$$

The PDF of the energy loss in a single inelastic collision is given by

$$p_{\mathrm{in}}(W) = \frac{1}{\sigma_{\mathrm{in}}}\frac{\mathrm{d}\sigma_{\mathrm{in}}}{\mathrm{d}W}, \tag{3.97}$$

where

$$\sigma_{\mathrm{in}} = \int_0^{W_{\mathrm{max}}} \frac{\mathrm{d}\sigma_{\mathrm{in}}}{\mathrm{d}W}\,\mathrm{d}W \tag{3.98}$$

is the total cross section for inelastic interactions. It is convenient to introduce the quantities

$$\sigma^{(n)}_{\mathrm{in}} \equiv \int_0^{W_{\mathrm{max}}} W^n \frac{\mathrm{d}\sigma_{\mathrm{in}}}{\mathrm{d}W}\,\mathrm{d}W = \sigma_{\mathrm{in}}\int_0^{W_{\mathrm{max}}} W^n p_{\mathrm{in}}(W)\,\mathrm{d}W = \sigma_{\mathrm{in}}\,\langle W^n\rangle, \tag{3.99}$$

where $\langle W^n\rangle$ denotes the n-th moment of the energy loss in a single collision (notice that $\sigma^{(0)}_{\mathrm{in}} = \sigma_{\mathrm{in}}$). $\sigma^{(1)}_{\mathrm{in}}$ and $\sigma^{(2)}_{\mathrm{in}}$ 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

$$\lambda^{-1}_{\mathrm{in}} = \mathcal{N}\sigma_{\mathrm{in}}, \tag{3.100}$$

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^2_{in} are defined by

$$S_{\mathrm{in}} = \mathcal{N}\sigma^{(1)}_{\mathrm{in}} = \frac{\langle W\rangle}{\lambda_{\mathrm{in}}} \tag{3.101}$$

and

$$\Omega^2_{\mathrm{in}} = \mathcal{N}\sigma^{(2)}_{\mathrm{in}} = \frac{\langle W^2\rangle}{\lambda_{\mathrm{in}}}. \tag{3.102}$$

Notice that the stopping power gives the average energy loss per unit path length[4]. The physical meaning of the straggling parameter is less direct. Consider a monoenergetic

[4]The term "stopping power" is somewhat misleading; in fact, S_{in} has the dimensions of force.

electron (or positron) beam of energy E that impinges normally on a foil of material of (small) thickness $\mathrm{d}s$, and assume that the electrons do not scatter (*i.e.*, they are not deflected) in the foil. The product $\Omega_{\mathrm{in}}^2\,\mathrm{d}s$ then gives the variance of the energy distribution of the beam after traversing the foil (see also Section 4.2).

The integrated cross sections $\sigma_{\mathrm{in}}^{(n)}$ can be calculated as

$$\sigma_{\mathrm{in}}^{(n)} = \sigma_{\mathrm{dis,l}}^{(n)} + \sigma_{\mathrm{dis,t}}^{(n)} + \sigma_{\mathrm{clo}}^{(n)}. \tag{3.103}$$

The contributions from distant longitudinal and transverse interactions are

$$\sigma_{\mathrm{dis,l}}^{(n)} = \frac{2\pi e^4}{\mathrm{m_e}v^2}\sum_k f_k \ln\left(\frac{Q'_k}{Q_-}\,\frac{Q_- + 2\mathrm{m_e}c^2}{Q'_k + 2\mathrm{m_e}c^2}\right)\int_0^{W_{\max}} W^{n-1}p_{\mathrm{dis}}(W)\,\mathrm{d}W \tag{3.104}$$

and

$$\sigma_{\mathrm{dis,t}}^{(n)} = \frac{2\pi e^4}{\mathrm{m_e}v^2}\sum_k f_k \left\{\ln\left(\frac{1}{1-\beta^2}\right) - \beta^2 - \delta_{\mathrm{F}}\right\}\int_0^{W_{\max}} W^{n-1}p_{\mathrm{dis}}(W)\,\mathrm{d}W\,, \tag{3.105}$$

respectively.

The integrated cross sections for close collisions are

$$\sigma_{\mathrm{clo}}^{(n)} = \frac{2\pi e^4}{\mathrm{m_e}v^2}\sum_k f_k \int_{Q'_k}^{W_{\max}} W^{n-2}F^{(\pm)}(E,W)\,\mathrm{d}W. \tag{3.106}$$

In the case of electrons, the integrals in this formula are of the form

$$J_n^{(-)} = \int W^{n-2}\left[1 + \left(\frac{W}{E-W}\right)^2 - \frac{(1-a)W}{E-W} + \frac{aW^2}{E^2}\right]\mathrm{d}W \tag{3.107}$$

and can be calculated analytically. For the orders 0, 1 and 2 we have

$$J_0^{(-)} = -\frac{1}{W} + \frac{1}{E-W} + \frac{1-a}{E}\ln\left(\frac{E-W}{W}\right) + \frac{aW}{E^2}, \tag{3.108}$$

$$J_1^{(-)} = \ln W + \frac{E}{E-W} + (2-a)\ln(E-W) + \frac{aW^2}{2E^2} \tag{3.109}$$

and

$$J_2^{(-)} = (2-a)W + \frac{2E^2 - W^2}{E-W} + (3-a)E\ln(E-W) + \frac{aW^3}{3E^2}. \tag{3.110}$$

For positrons, the integrals in (3.106),

$$J_n^{(+)} \equiv \int W^{n-2}\left[1 - b_1\frac{W}{E} + b_2\left(\frac{W}{E}\right)^2 - b_3\left(\frac{W}{E}\right)^3 + b_4\left(\frac{W}{E}\right)^4\right]\mathrm{d}W, \tag{3.111}$$

can also be evaluated analytically as

$$J_0^{(+)} = -\frac{1}{W} - b_1\frac{\ln W}{E} + b_2\frac{W}{E^2} - b_3\frac{W^2}{2E^3} + b_4\frac{W^3}{3E^4}, \tag{3.112}$$

$$J_1^{(+)} = \ln W - b_1\frac{W}{E} + b_2\frac{W^2}{2E^2} - b_3\frac{W^3}{3E^3} + b_4\frac{W^4}{4E^4} \tag{3.113}$$

and

$$J_2^{(+)} = W - b_1\frac{W^2}{2E} + b_2\frac{W^3}{3E^2} - b_3\frac{W^4}{4E^3} + b_4\frac{W^5}{5E^4}. \tag{3.114}$$

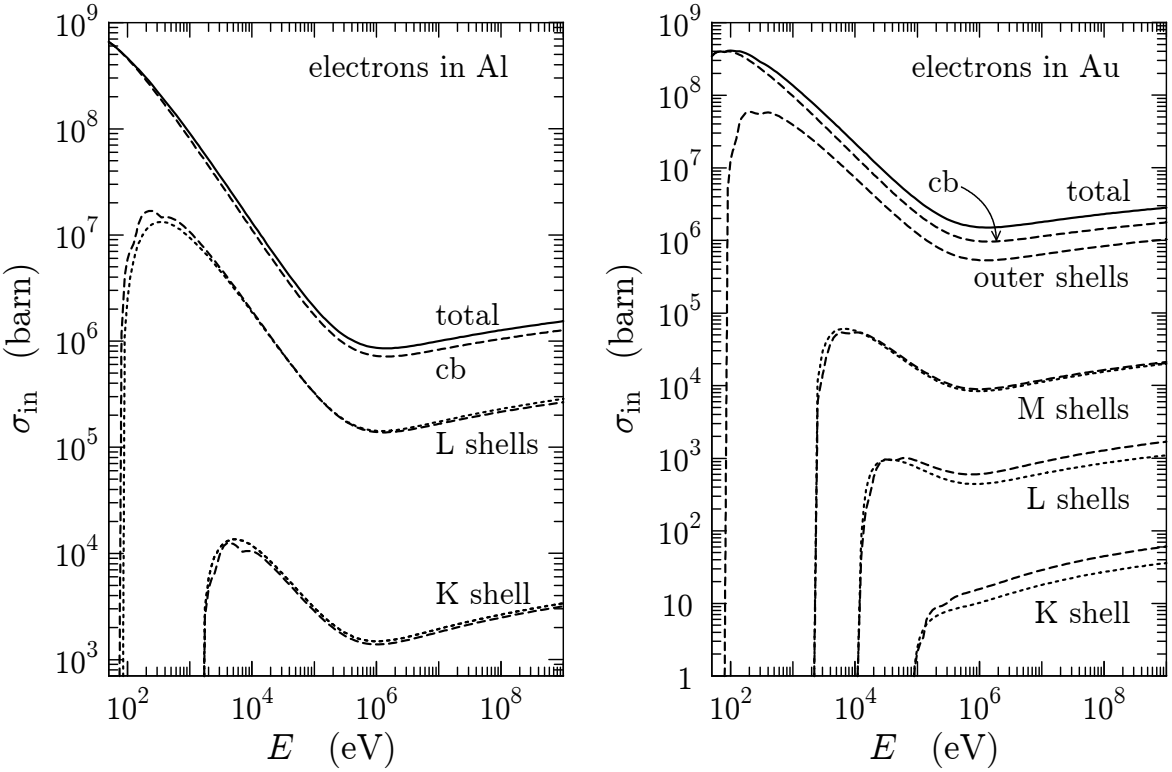

Figure 3.9: Total inelastic cross sections for electrons in aluminium and gold and contributions from the K shell, L and M shells, conduction band (cb) and outer shells, calculated from our model GOS ignoring density effect corrections (*i.e.*, with $\delta_{\rm F} = 0$). In the calculations we have used the following values of the oscillator strength and the resonance energy for conduction-band excitations: $f_{\rm cb} = 3$ and $W_{\rm cb} = 15.8$ eV for aluminium, and $f_{\rm cb} = 11$ and $W_{\rm cb} = 40$ eV for Au. The dotted curves represent ionisation cross sections for the K shell and the L and M shells taken from the numerical DWBA database described in Section 3.2.6. Note: 1 barn=10^{-24} cm^2.

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 shell", "L shells" and "M shells" represent

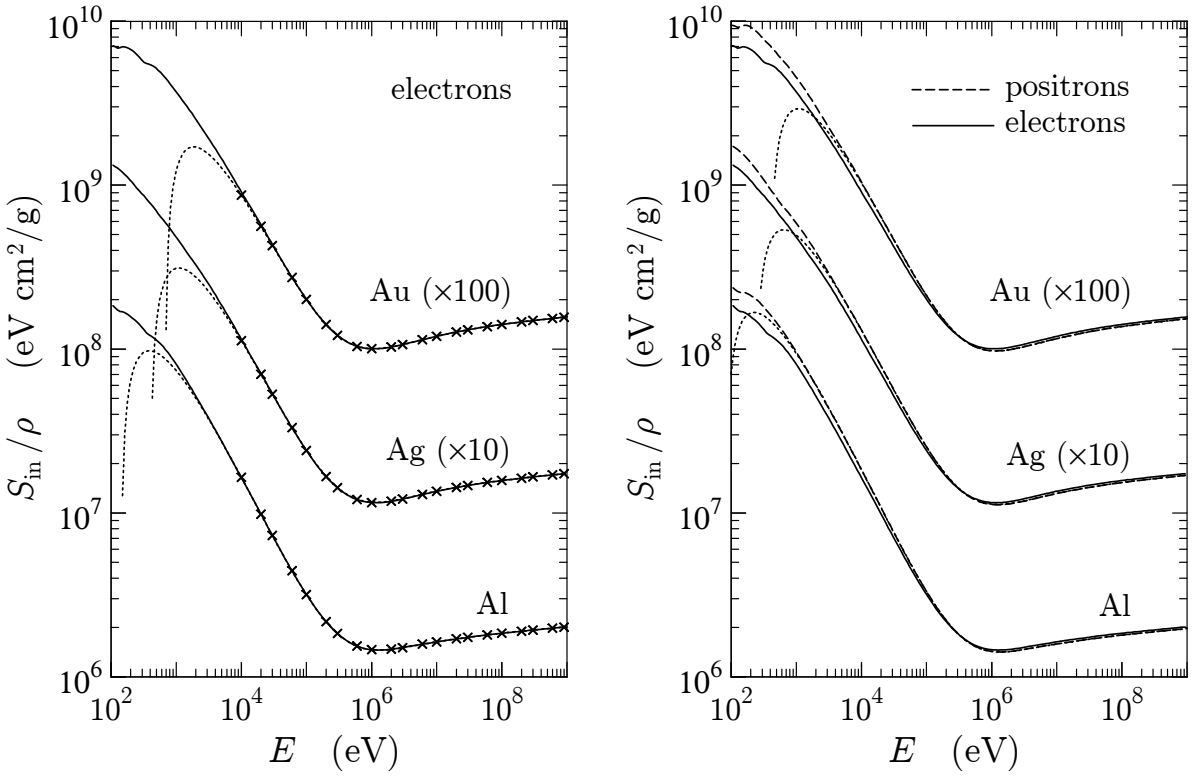

Figure 3.10: Collision stopping power $S_{\rm in}/\rho$ for electrons and positrons in aluminium, silver (×10) and gold (×100) as a function of the kinetic energy. Continuous and dashed curves are results from the present model. Crosses are data from the tables in the ICRU Report 37 (1984) and in Berger and Seltzer (1982). The dotted curves are predictions from the Bethe formula (3.120), for electrons and positrons. (Adapted from Salvat and Fernández-Varea, 1992).

cross sections for ionisation in these shells (with the added contributions from L and M subshells). The cross section for ionisation of a bound shell decreases rapidly with the shell ionisation energy U_k (because energy transfers less than U_k, 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. 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-, L and M-shell ionisation cross sections obtained from the numerical DWBA database described in Section 3.2.6, which are known to agree reasonably well with experimental data (Bote and Salvat, 2008; Llovet *et al.*, 2014a). We see that there are significant differences between the cross sections from the database 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. Still, the qualitative good agreement between our cross sections and those from the numerical database is noteworthy.

Collision stopping powers for electrons in aluminium, silver and gold obtained from the present analytical model are compared with sample values from the ICRU Report 37 (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 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 $W_{\rm cb}$, 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, $W_{\rm cb}$ 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 $W_{\rm cb}$. To generate data for aluminium, Fig. 3.11, we have set $W_{\rm cb} = 15$ eV, which is the measured energy of volume plasmons in the metal [Eq. (3.62) with $f_{\rm cb} = 3$ conduction electrons per atom gives $W_{\rm cb} = 15.8$ 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 $f_{\rm cb} = 11$ conduction electrons per atom gives $W_{\rm cb} = 30$ eV. Figure 3.11 shows stopping powers and mean free paths for electrons in gold obtained with $W_{\rm cb} = 30$ 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 $U_k \ll 2{\rm m_e}c^2$ (for the most unfavourable case of the K shell of heavy elements, U_k is of the order of $2{\rm m_e}c^2/10$). Under these circumstances, $Q_- \ll 2{\rm m_e}c^2$ and we can use the approximation [see Eq. (A.35)]

$$Q_- \simeq W_k^2/(2{\rm m_e}c^2\beta^2). \tag{3.115}$$

The contribution from distant (longitudinal and transverse) interactions to the stopping cross section is then [see Eqs. (3.94) and (3.95)]

$$\sigma_{\rm dis}^{(1)} \simeq \frac{2\pi e^4}{{\rm m_e}v^2}\sum_k f_k \left\{\ln\left(\frac{Q_k\, 2{\rm m_e}c^2\,\beta^2}{W_k^2}\right) + \ln\left(\frac{1}{1-\beta^2}\right) - \beta^2 - \delta_{\rm F}\right\}. \tag{3.116}$$

The contribution of close interactions is given by

$$\sigma_{\rm clo}^{(1)} = \frac{2\pi e^4}{{\rm m_e}v^2}\sum_k f_k \int_{Q_k}^{W_{\rm max}} W^{-1}F^{(\pm)}(E,W)\,{\rm d}W. \tag{3.117}$$

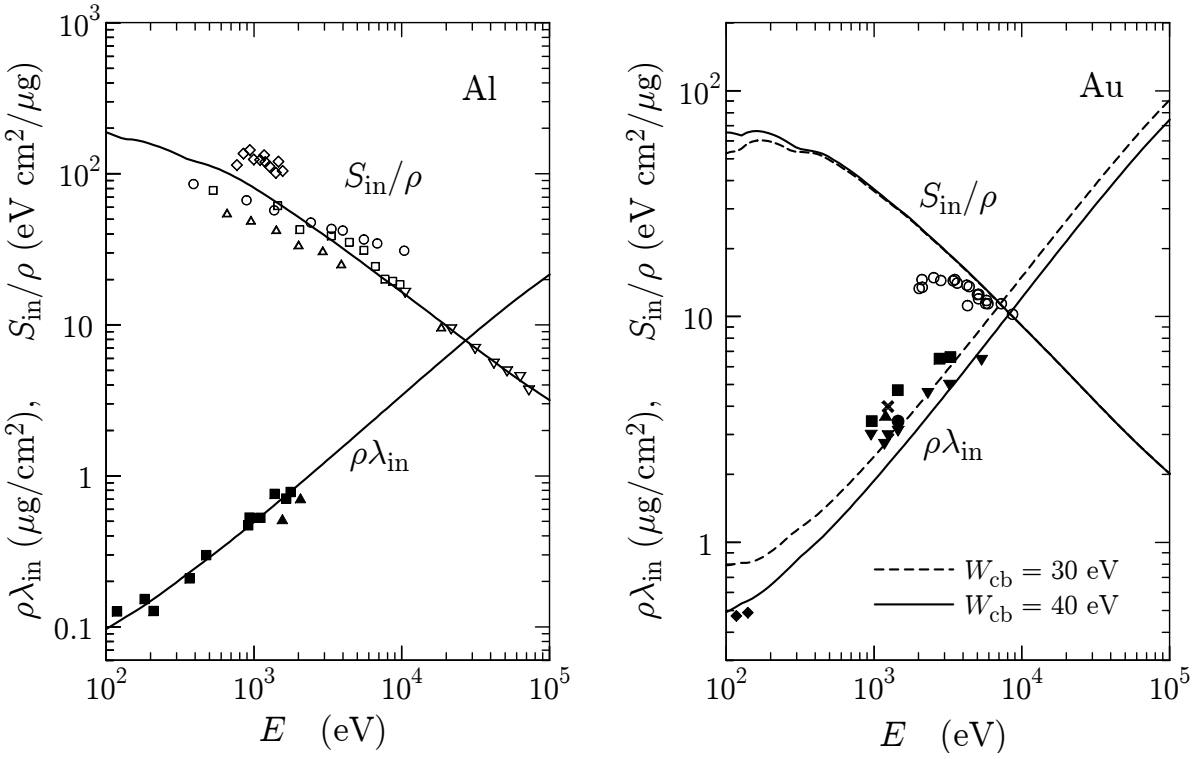

Figure 3.11: Collision mean free path and stopping power for low-energy electrons in aluminium and gold. The plotted quantities are $\rho\lambda_{\rm in}$ and $S_{\rm in}/\rho$. Special symbols are experimental data from different sources (see Fernández-Varea *et al.*, 1993b); closed symbols for mean free paths and open symbols for stopping powers. (Adapted from Salvat and Fernández-Varea, 1992).

Recalling that $E \gg Q_k \simeq U_k$, we have

$$\sigma_{\rm clo}^{(1)} \simeq \frac{2\pi e^4}{{\rm m_e}v^2}\sum_k f_k \left\{ \ln\left(\frac{E}{Q_k}\right) + 1 - \left[1+\beta^2+2\sqrt{1-\beta^2}\right]\ln 2 + \frac{1}{8}\left(1-\sqrt{1-\beta^2}\right)^2 \right\} \tag{3.118}$$

for electrons and

$$\sigma_{\rm clo}^{(1)} \simeq \frac{2\pi e^4}{{\rm m_e}v^2}\sum_k f_k \left\{ \ln\left(\frac{E}{Q_k}\right) - b_1 + \frac{b_2}{2} - \frac{b_3}{3} + \frac{b_4}{4} \right\} \tag{3.119}$$

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,

$$S_{\rm in} \equiv \mathcal{N}\left(\sigma_{\rm dis}^{(1)} + \sigma_{\rm clo}^{(1)}\right) = \mathcal{N}\frac{2\pi e^4}{{\rm m_e}v^2} Z \left\{ \ln\left(\frac{E^2}{I^2}\,\frac{\gamma+1}{2}\right) + f^{(\pm)}(\gamma) - \delta_{\rm F} \right\}, \tag{3.120}$$

where

$$f^{(-)}(\gamma) = 1 - \beta^2 - \frac{2\gamma - 1}{\gamma^2} \ln 2 + \frac{1}{8} \left(\frac{\gamma - 1}{\gamma} \right)^2 \tag{3.121}$$

and

$$f^{(+)}(\gamma) = 2 \ln 2 - \frac{\beta^2}{12} \left[23 + \frac{14}{\gamma + 1} + \frac{10}{(\gamma + 1)^2} + \frac{4}{(\gamma + 1)^3} \right] \tag{3.122}$$

for electrons and positrons, respectively. This formula can be derived from very general arguments that do not require knowledge of 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 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

$$S_{\text{in}} \simeq \mathcal{N} \frac{2\pi e^4}{\mathrm{m_e} v^2} Z \left\{ \ln \left(\frac{E^2}{\Omega_{\text{p}}^2} \frac{\gamma + 1}{2\gamma^2} \right) + f^{(\pm)}(\gamma) + 1 \right\}. \tag{3.123}$$

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 (3.93)–(3.96) permit the random sampling of the energy loss W and the angular deflection θ by using purely analytical methods (Salvat and Fernández-Varea, 1992). 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 ($W_{\text{cc}} = 0$).

The first stage of the simulation is the selection of the active oscillator, for which we need to know the restricted total cross section,

$$\begin{aligned} \sigma(W_{\text{cc}}) &\equiv \int_{W_{\text{cc}}}^{W_{\max}} \frac{\mathrm{d}\sigma_{\text{in}}}{\mathrm{d}W} \, \mathrm{d}W = \sigma_{\text{dis,l}}(W_{\text{cc}}) + \sigma_{\text{dis,t}}(W_{\text{cc}}) + \sigma_{\text{clo}}(W_{\text{cc}}) \\ &= \sum_k \sigma_k(W_{\text{cc}}), \end{aligned} \tag{3.124}$$

as well as the contribution of each oscillator, $\sigma_k(W_{\rm cc})$. The active oscillator is sampled from the point probabilities $p_k = \sigma_k(W_{\rm cc})/\sigma(W_{\rm cc})$. 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 $\sigma_k(W_{\rm cc})$ of all the oscillators were calculated for each interaction].

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

The energy loss in distant excitations of the k-th oscillator is sampled from the corresponding energy-loss DCS. In the case of the conduction band, $W = W_k$. For inner shells, the energy loss follows the distribution $p_{\rm dist}(W)$, Eq. (3.76), restricted to the interval $(W_{\rm cc}, W_{\rm dis})$. Using the inverse-transform method (Section 1.2.2) we find that random values of the energy loss restricted to an arbitrary subinterval (W_1, W_2) $[U_k \leq W_1 < W_2 \leq W_{\rm dis}]$ are provided by the following sampling formula,

$$W = W_{\rm dis} - \sqrt{(W_{\rm dis} - W_1)^2 - \xi(W_2 - W_1)(2W_{\rm dis} - W_1 - W_2)}, \tag{3.125}$$

where ξ is a random number.

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 PDF of the polar scattering angle θ is that of the δ-oscillator [see the comments after Eq. (3.82)]. The (unnormalised) PDF of Q is defined by Eq. (3.67),

$$P_{{\rm d}k}(Q) = \begin{cases} \dfrac{1}{Q\,[1 + Q/(2{\rm m_e}c^2)]} & \text{if } Q_- < Q < Q'_k, \\ 0 & \text{otherwise,} \end{cases} \tag{3.126}$$

where Q_- is the minimum recoil energy, Eq. (A.31), corresponding to either the resonance energy W_k (for the conduction band) or the modified resonance energy W'_k, Eq. (3.78) (for an inner shell). The quantity Q'_k is the modified cutoff recoil energy, Eq. (3.80) (modified only for inner-shell excitations). Random sampling from the PDF (3.126) can be performed by the inverse-transform method, which gives the sampling formula

$$Q = Q_{\rm S} \left\{ \left[\frac{Q_{\rm S}}{Q'_k} \left(1 + \frac{Q'_k}{2{\rm m_e}c^2} \right) \right]^{\xi} - \frac{Q_{\rm S}}{2{\rm m_e}c^2} \right\}^{-1}, \tag{3.127}$$

where

$$Q_{\rm S} \equiv \frac{Q_-}{1+Q_-/\left(2{\rm m_e}c^2\right)}. \tag{3.128}$$

Once the energy loss and the recoil energy have been sampled, the polar scattering angle θ is determined from Eq. (A.40),

$$\cos\theta = \frac{E(E+2{\rm m_e}c^2)+(E-W'_k)(E-W'_k+2{\rm m_e}c^2)-Q(Q+2{\rm m_e}c^2)}{2\sqrt{E(E+2{\rm m_e}c^2)\,(E-W'_k)(E-W'_k+2{\rm m_e}c^2)}}\,, \tag{3.129}$$

where we have considered that $W = W'_k$. 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 $\kappa \equiv W/(E+U_k)$. The PDF of κ in close collisions of electrons with the k-th oscillator is given by [see Eqs. (3.86) and (3.87)]

$$P_k^{(-)}(\kappa) \equiv \kappa^{-2}F^{(-)}(E,W)\,\Theta(\kappa-\kappa_{\rm c})\,\Theta\left(\frac{1}{2}-\kappa\right) = \left[\frac{1}{\kappa^2}+\frac{1}{(1-\kappa)^2}\right.$$
$$\left.-\frac{1}{\kappa(1-\kappa)}+a\left(1+\frac{1}{\kappa(1-\kappa)}\right)\right]\,\Theta(\kappa-\kappa_{\rm c})\,\Theta\left(\frac{1}{2}-\kappa\right), \tag{3.130}$$

with $\kappa_{\rm c} \equiv \max(Q_k, W_{\rm cc})/E$. Notice that the maximum allowed value of κ is 1/2. Here, normalisation is irrelevant.

We introduce the distribution

$$\Phi^{(-)}(\kappa) \equiv (\kappa^{-2}+5a)\,\Theta(\kappa-\kappa_{\rm c})\,\Theta\left(\frac{1}{2}-\kappa\right), \qquad a \equiv \left(\frac{\gamma-1}{\gamma}\right)^2. \tag{3.131}$$

It may be shown that $\Phi^{(-)} > P_k^{(-)}$ in the interval $(\kappa_{\rm c}, \frac{1}{2})$. Therefore, we can sample the reduced energy loss κ from the PDF (3.130) by using the rejection method (see Section 1.2.5) with trial values sampled from the distribution (3.131) and acceptance probability $P_k^{(-)}/\Phi^{(-)}$.

Random sampling from the PDF (3.131), can be performed by using the composition method (Section 1.2.6). We consider the following decomposition of the (normalised) PDF given by Eq. (3.131):

$$\Phi^{(-)}_{\rm norm}(\kappa) = \frac{1}{1+5a\kappa_{\rm c}/2}\left[p_1(\kappa)+(5a\kappa_{\rm c}/2)p_2(\kappa)\right], \tag{3.132}$$

where

$$p_1(\kappa) = \frac{\kappa_{\rm c}}{1-2\kappa_{\rm c}}\kappa^{-2}, \qquad p_2(\kappa) = \frac{2}{1-2\kappa_{\rm c}} \tag{3.133}$$

are normalised PDFs in the interval $(\kappa_{\rm c}, \frac{1}{2})$. Random values of κ from the PDF (3.131) 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 (3.130) proceeds as follows:

(i) Sample κ from the distribution given by Eq. (3.131).

(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 (Salvat and Fernández-Varea, 1992).

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 $W = \kappa(E + U_k)$, the polar scattering angle θ is obtained from Eq. (A.40) with $Q = W$. This yields

$$\cos^2\theta = \frac{E - W}{E}\frac{E + 2m_e c^2}{E - W + 2m_e c^2}, \tag{3.134}$$

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. (3.91) and (3.92)]

$$\begin{aligned} P_k^{(+)}(\kappa) &= \kappa^{-2} F_k^{(+)}(E,W)\,\Theta(\kappa-\kappa_{\rm c})\,\Theta(1-\kappa) \\ &= \left[\frac{1}{\kappa^2} - \frac{b_1}{\kappa} + b_2 - b_3\kappa + b_4\kappa^2\right]\Theta(\kappa-\kappa_{\rm c})\,\Theta(1-\kappa) \end{aligned} \tag{3.135}$$

with $\kappa_{\rm c} \equiv \max(Q_k, W_{\rm cc})/E$. The maximum allowed reduced energy loss is 1. Again, normalisation is not important.

Consider the distribution

$$\Phi^{(+)}(\kappa) \equiv \kappa^{-2}\Theta(\kappa-\kappa_{\rm c})\,\Theta(1-\kappa). \tag{3.136}$$

It is easy to see that $\Phi^{(+)} > P_k^{(+)}$ in the interval $(\kappa_{\rm c}, 1)$. Therefore, we can generate κ from the PDF, Eq. (3.135), by using the rejection method with trial values sampled from the distribution of Eq. (3.136) 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 (3.135) is:

(i) Sample κ from the PDF (3.136), as $\kappa = \kappa_{\rm c}/[1-\xi(1-\kappa_{\rm c})]$.

(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 $\kappa_{\rm c}$, 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 $E_{\rm s} = W - U_i$, 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 is 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)],

$$\cos^2\theta_\mathrm{s} = \frac{W^2/\beta^2}{Q(Q+2\mathrm{m_e}c^2)}\left(1+\frac{Q(Q+2\mathrm{m_e}c^2)-W^2}{2W(E+\mathrm{m_e}c^2)}\right)^2 . \tag{3.137}$$

For distant interactions, the value of W should be set equal to the modified resonance energy W_k' (W_k for the conduction band), because the recoil energy Q was sampled under the assumption that $W = W_k'$ [see Eq. (3.126)]. In the case of close collisions ($Q = W$), expression (3.137) simplifies to

$$\cos\theta_\mathrm{s}\,(Q=W) = \left(\frac{W}{E}\,\frac{E+2\mathrm{m_e}c^2}{W+2\mathrm{m_e}c^2}\right)^{1/2} , \tag{3.138}$$

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 $\phi_\mathrm{s} = \pi + \phi$.

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

Reliable values of the ionisation cross section for projectile electrons with kinetic energy E greater than about 30 times the ionisation energy U_i of the active electron shell can be calculated using the plane-wave (first) Born approximation (PWBA). This approximation worsens when the energy of the projectile decreases, mainly because it neglects the distortion of the projectile wave functions caused by the electrostatic field of the

target atom and, in the case of electron collisions, it does not account for exchange effects (which arise from the indistinguishability of the projectile and target electrons). As a consequence, the PWBA yields the same cross sections for electrons and positrons.

A more elaborate theoretical description of total ionisation cross sections is obtained from the relativistic distorted-wave Born approximation (DWBA), which consistently accounts for the effects of both distortion and exchange (see Bote and Salvat, 2008, and references therein). DWBA calculations yield total cross sections in fairly good agreement with experimental data for ionisation of K, L and M shells (see Llovet *et al.*, 2014a). These calculations involve the expansion of free-state wave functions as partial wave series and the subsequent evaluation of multiple radial integrals. Since the convergence of partial-wave series worsens with increasing kinetic energies, DWBA calculations are feasible only for projectiles with relatively small energies, up to about $20U_i$.

We have calculated an extensive numerical database of ionisation cross sections for K, L, M and N shells of all the elements from hydrogen ($Z = 1$) to einsteinium ($Z = 99$), for projectiles with kinetic energies from threshold up to 1 GeV. The theoretical model adopted in the calculations combines the DWBA and the PWBA, as described by Bote and Salvat (2008). The DWBA is used to calculate the ionisation cross section for projectiles with energies from $\sim U_i$ up to $16U_i$. For higher energies, the cross section is obtained by multiplying the PWBA cross section by an empirical energy-dependent scaling factor, which tends to unity at high energies where the PWBA is expected to be reliable. This calculation scheme accounts for differences between the cross sections for ionisation by electrons and positrons. Bote *et al.* (2009, 2011) published analytical formulas for the easy calculation of cross sections for ionisation of K, L and M shells of neutral atoms by impact of electrons and positrons. These formulas are coded in a simple Fortran program that yields cross sections that agree with those in the numerical database to within about 1%. The present database also includes ionisation cross sections for N shells of heavy elements, which were calculated more recently using the same methodology as for the K, L and M shells. In the following, the cross section for ionisation of the i-th shell of a given element by electron ($-$) and positron ($+$) impact will be denoted by $\sigma_{\mathrm{si},i}^{(\pm)}(E)$, where E is the kinetic energy of the particle.

Figure 3.12 displays the cross sections of the numerical database for the ionisation of K shells and L and M subshells of argon, silver, gold, and uranium atoms by impact of electrons and positrons, together with results from the PWBA for K and L shells (Scofield, 1978). The differences between the cross sections of the database and the PWBA at relatively low energies are due to exchange and Coulomb corrections, which are not considered in Scofield's PWBA calculations. The dashed curves in Fig. 3.12 represent cross sections for ionisation by positron impact. Differences between cross sections for electrons and positrons arise from the distortion of the projectile wave functions by the electrostatic field of the target atom, which is attractive for electrons and repulsive for positrons, and from electron exchange effects. It is worth mentioning that the theoretical model used to generate the database disregards the influence of the polarisability of the medium (density effect) on inner-shell ionisation. This effect causes a reduction of the ionisation cross section for projectiles with very high energies, which

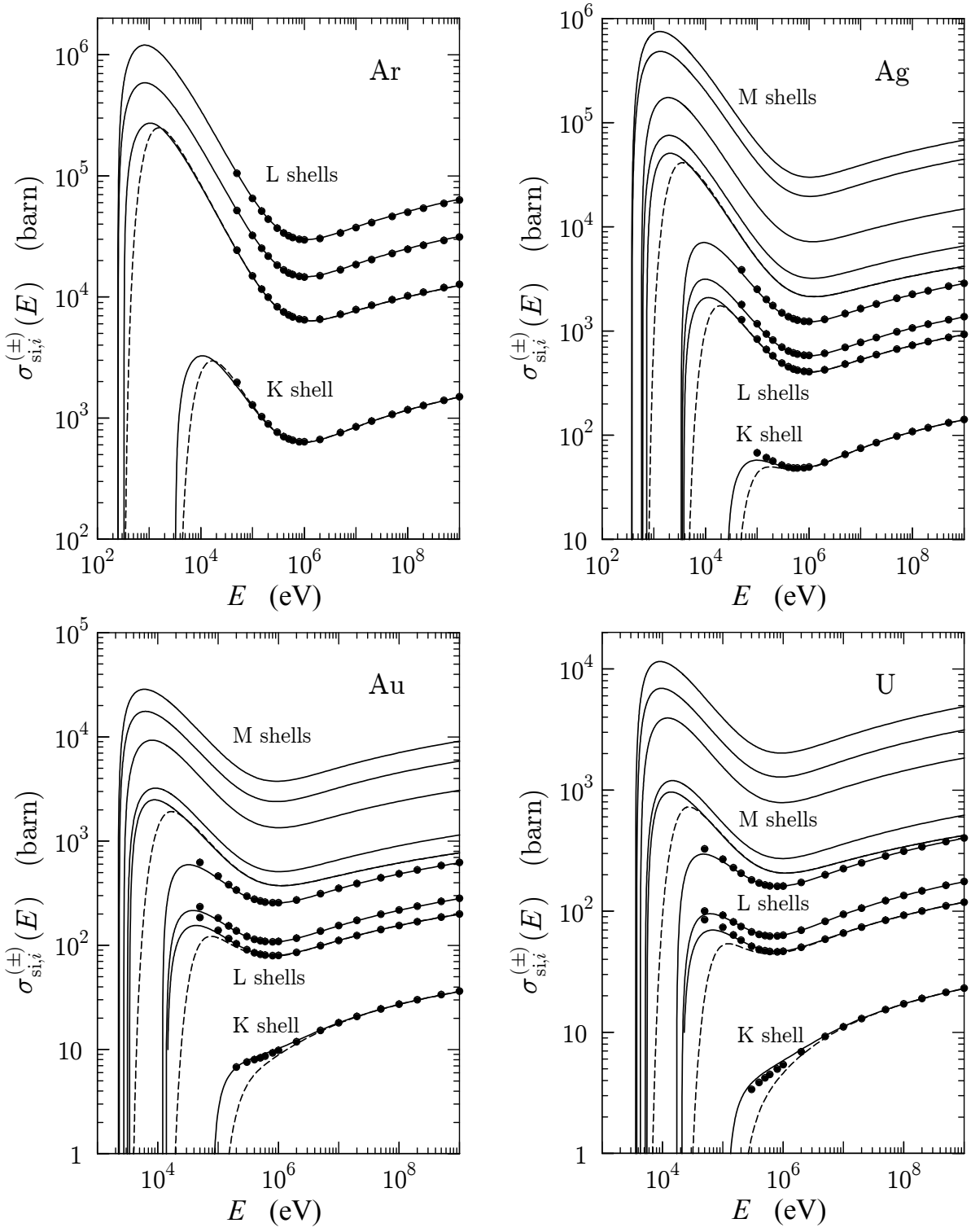

Figure 3.12: Cross sections for ionisation of the K shell and the L and M subshells of argon, silver, gold and uranium atoms by electron and positron impact as functions of the kinetic energy E of the projectiles. Solid and dashed curves represent the cross sections from the numerical database for electrons and positrons, respectively. Circles are values calculated by Scofield (1978) using the PWBA.

has been considered, *e.g.*, by Scofield (1978).

In the present version of PENELOPE, ionisation cross sections are obtained from the numerical database, which consists of 198 files with tables of the cross sections for ionisation of K, L, M and N shells of the elements ($Z = 1-99$) by electron and positron impact. The grid of energies where $\sigma_{\mathrm{si},i}^{(\pm)}(E)$ is tabulated varies with the element, because it has a higher density of points above each ionisation threshold to allow accurate linear log-log interpolation. The atomic cross section for inner shell ionisation is obtained as

$$\sigma_{\mathrm{si,atom}}^{(\pm)}(E) = \sum_i \sigma_{\mathrm{si},i}^{(\pm)}(E), \tag{3.139}$$

where the summation extends over the inner shells of the atom. In the simulation, we consider only K, L, M and N shells with ionisation energies U_i larger than the absorption energies of electrons or photons, `EABS(1)` or `EABS(2)` (see Section 7.1.2). There is no need to simulate ionisations of shells that are unable to originate x rays and Auger electrons with energies higher than the corresponding cutoffs. In the case of compounds, the molecular cross section is evaluated as (additivity approximation)

$$\sigma_{\mathrm{si,mol}}^{(\pm)}(E) = \sum_i \sigma_{\mathrm{si},i}^{(\pm)}(E), \tag{3.140}$$

where the summation now extends over all inner shells of the atoms in the molecule.

3.2.6.1 Simulation of ionizing collisions

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 (see Fig. 3.9). 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.

In the first versions of PENELOPE the impact ionisation of inner shells was considered as a separate process, which did not alter the energy and the direction of the projectile particle. With the adopted DWBA cross sections, this approach had the advantage of yielding a nearly correct number of ionisations per unit path length, without altering the modelling of inelastic collisions, but at the expense of occasionally producing negative doses (x rays left sites where no energy was deposited). In the 2014 version of the code we changed strategy and considered ionisations of inner shells as real inelastic collisions. This was accomplished by setting the total cross section for collisions with electrons in each inner shell equal to the DWBA cross section, and by renormalising the total cross sections of outer shells so as to keep the value of the collision stopping power unaltered. The energy loss and the recoil energy in ionizing events are sampled using the algorithm described in Section 3.2.5. With the present scheme, x rays and Auger electrons are emitted only when the projectile delivers sufficient energy, thus preventing the occurrence of negative energy deposits.

Our renormalisation procedure is based on the observation that the total cross section is dominated by contributions from the lower oscillators (*i.e.*, those with smaller resonance energies). Let us consider the total cross section, $\sigma_{\rm in}$, decomposed into contributions from the various oscillators,

$$\sigma_{\rm in}(E) = \sum_i \sigma_{{\rm in},i}(E) + \sum_j \sigma_{{\rm in},j}(E), \tag{3.141}$$

where the first summation is over inner shells (K to N7 shells with binding energies less than the cut-off energy $E_{\rm c}$ defined by Eq. (2.112)), which we have separated from the contributions from outer shells. In order to get the correct average number of ionisations, we replace the total cross section of each inner shell, $\sigma_{{\rm in},i}(E)$, with the DWBA cross section $\sigma^{(\pm)}_{{\rm si},i}(E)$, without altering details of the PDFs of the energy loss and scattering angle defined by the GOS model. Furthermore, to approximately account for the density effect, we reduce the DWBA cross sections by a factor equal to the ratio of the cross sections obtained from the GOS model with and without the density effect correction, $\delta_{\rm F}$. Finally, to keep the stopping power unaltered, the total cross section of outer shells, $\sigma_{{\rm in},j}(E)$, is multiplied by an energy-dependent scaling factor, $N(E)$, the same for all outer shells, such that the adopted stopping power is reproduced exactly. That is, we set

$$\sigma_{\rm in}(E) = \sum_i \sigma_{{\rm si},i}(E) + N(E) \sum_j \sigma_{{\rm in},j}(E). \tag{3.142}$$

Formally, this modification is equivalent to replacing the oscillator strengths f_j of outer shells with $f'_j = f_j N(E)$.

Hard inelastic collisions with inner shells are assumed to ionize the target atom and the relaxation of the resulting vacancies is simulated by the `RELAX` subroutine, using the transition probabilities given in the Evaluated Atomic Data Library of Perkins *et al.* (1991). When the age of particles is considered, the finite lifetime of atomic states is accounted for by using decay times sampled from the PDF (2.116) (see Section 2.6.1).

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, 1977). 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 $\mathrm{m_e}c^2$, 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 (2.78), 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)

$$\frac{\mathrm{d}\sigma_{\mathrm{br}}^{(\mathrm{BH})}}{\mathrm{d}W} = r_{\mathrm{e}}^2 \alpha Z(Z+\eta)\frac{1}{W}\left[\epsilon^2\,\varphi_1(b) + \frac{4}{3}(1-\epsilon)\,\varphi_2(b)\right], \tag{3.143}$$

where α is the fine-structure constant, r_{e} is the classical electron radius,

$$\epsilon = \frac{W}{E+\mathrm{m_e}c^2} = \frac{W}{\gamma\,\mathrm{m_e}c^2}, \qquad b = \frac{R\mathrm{m_e}c}{\hbar}\,\frac{1}{2\gamma}\,\frac{\epsilon}{1-\epsilon}, \tag{3.144}$$

and

$$\begin{aligned}
\varphi_1(b) &= 4\ln(R\mathrm{m_e}c/\hbar) + 2 - 2\ln(1+b^2) - 4b\arctan(b^{-1}),\\
\varphi_2(b) &= 4\ln(R\mathrm{m_e}c/\hbar) + \frac{7}{3} - 2\ln(1+b^2) - 6b\arctan(b^{-1})\\
&\quad - b^2\left[4 - 4b\arctan(b^{-1}) - 3\ln(1+b^{-2})\right].
\end{aligned} \tag{3.145}$$

The quantity η in Eq. (3.143) accounts for production of bremsstrahlung in the field of the atomic electrons (see 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 $W\mathrm{d}\sigma_{\mathrm{br}}/\mathrm{d}W$ varies smoothly with E and W. It is therefore customary to express the DCS for bremsstrahlung emission by electrons in the form

$$\frac{\mathrm{d}\sigma_{\mathrm{br}}}{\mathrm{d}W} = \frac{Z^2}{\beta^2}\,\frac{1}{W}\,\chi(Z,E,\kappa), \tag{3.146}$$

where W is the energy of the emitted photon, κ is the reduced photon energy, defined as

$$\kappa \equiv W/E, \tag{3.147}$$

which takes values between 0 and 1. The quantity

$$\chi(Z,E,\kappa) = (\beta^2/Z^2)W\frac{\mathrm{d}\sigma_{\mathrm{br}}}{\mathrm{d}W} \tag{3.148}$$

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, 1981) 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 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 $\mathrm{X_xY_y}$, whose molecules consist of x atoms of the element X and y atoms of the element Y. The molecular DCS is

$$\frac{\mathrm{d}\sigma_{\mathrm{br,mol}}}{\mathrm{d}W} = \mathrm{x}\frac{Z_{\mathrm{X}}^2}{\beta^2}\,\frac{1}{W}\,\chi(Z_{\mathrm{X}},E,\kappa) + \mathrm{y}\frac{Z_{\mathrm{Y}}^2}{\beta^2}\,\frac{1}{W}\,\chi(Z_{\mathrm{Y}},E,\kappa). \tag{3.149}$$

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. (3.146),

$$\frac{\mathrm{d}\sigma_{\mathrm{br,mol}}}{\mathrm{d}W} = \frac{Z_{\mathrm{eq}}^2}{\beta^2}\,\frac{1}{W}\,\chi_{\mathrm{mol}}(Z_{\mathrm{eq}},E,\kappa), \tag{3.150}$$

where

$$Z_{\mathrm{eq}}^2 \equiv \frac{1}{\mathrm{x}+\mathrm{y}}\left(\mathrm{x}Z_{\mathrm{X}}^2 + \mathrm{y}Z_{\mathrm{Y}}^2\right) \tag{3.151}$$

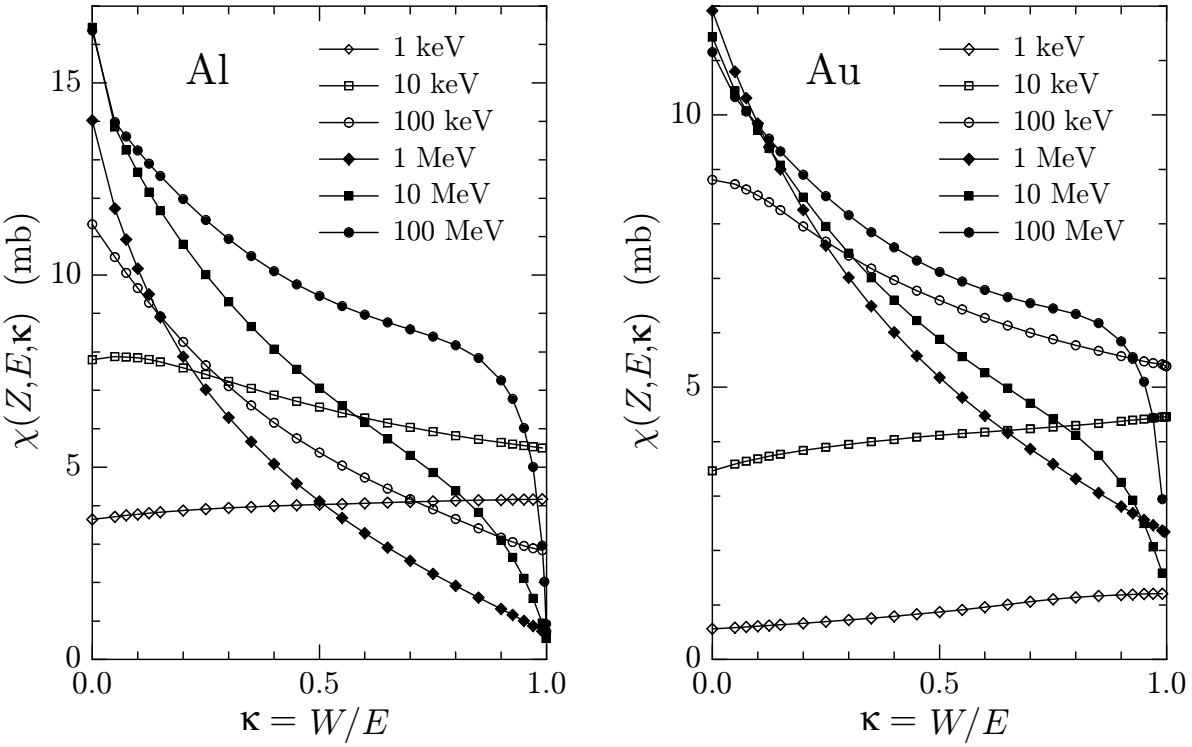

Figure 3.13: Numerical scaled bremsstrahlung energy-loss DCSs of aluminium and gold for electrons with the indicated energies (Seltzer and Berger, 1986) [1 mb = 10^{-27} cm^2]. (Adapted from Salvat *et al.*, 2006).

is the "equivalent" atomic number $Z_{\rm eq}$ and

$$\chi_{\rm mol}(Z_{\rm eq},E,\kappa) = \frac{{\rm x}Z_{\rm X}^2}{Z_{\rm eq}^2}\chi(Z_{\rm X},E,\kappa) + \frac{{\rm y}Z_{\rm Y}^2}{Z_{\rm eq}^2}\chi(Z_{\rm Y},E,\kappa) \tag{3.152}$$

is the molecular scaled DCS. Radiative events will be sampled directly from the molecular DCS (3.150). 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.*,

$$\frac{{\rm d}\sigma_{\rm br}^{(+)}}{{\rm d}W} = F_{\rm p}(Z,E)\,\frac{{\rm d}\sigma_{\rm br}^{(-)}}{{\rm d}W}. \tag{3.153}$$

The factor $F_{\rm p}(Z,E)$ 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

$$\begin{aligned} F_{\rm p}(Z,E) \;=\; & 1-\exp(-1.2359\times10^{-1}\,t+6.1274\times10^{-2}\,t^2-3.1516\times10^{-2}\,t^3 \\ & +7.7446\times10^{-3}\,t^4-1.0595\times10^{-3}\,t^5+7.0568\times10^{-5}\,t^6 \\ & -1.8080\times10^{-6}\,t^7), \end{aligned} \tag{3.154}$$

where

$$t=\ln\left(1+\frac{10^6}{Z^2}\frac{E}{{\rm m_e}c^2}\right). \tag{3.155}$$

Expression (3.154) reproduces the values of $F_{\rm p}(Z,E)$ 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 (3.146) at $W=0$ (the so-called infrared divergence), which is associated with the null mass of the photon. Nevertheless, the cross section for emission of photons with reduced energy larger than a given cutoff value $W_{\rm cr}$ is finite. The corresponding mean free path is

$$\lambda_{\rm br}^{-1}(E;W_{\rm cr})\equiv\mathcal{N}\int_{W_{\rm cr}}^{E}\frac{{\rm d}\sigma_{\rm br}}{{\rm d}W}\,{\rm d}W=\mathcal{N}\frac{Z^2}{\beta^2}\int_{\kappa_{\rm cr}}^{1}\frac{1}{\kappa}\,\chi(Z,E,\kappa)\,{\rm d}\kappa, \tag{3.156}$$

where $\kappa_{\rm cr}=W_{\rm cr}/E$. Note that $\lambda_{\rm br}^{-1}(E;W_{\rm cr})$ gives the average number of photons emitted per unit path length with energy greater than $W_{\rm cr}$.

The radiative stopping power and the radiative energy straggling parameter, are defined by

$$S_{\rm br}(E)\equiv\mathcal{N}\int_0^E W\frac{{\rm d}\sigma_{\rm br}}{{\rm d}W}\,{\rm d}W=\mathcal{N}\frac{Z^2}{\beta^2}E\int_0^1\chi(Z,E,\kappa)\,{\rm d}\kappa \tag{3.157}$$

and

$$\Omega_{\rm br}^2(E)\equiv\mathcal{N}\int_0^E W^2\frac{{\rm d}\sigma_{\rm br}}{{\rm d}W}\,{\rm d}W=\mathcal{N}\frac{Z^2}{\beta^2}E^2\int_0^1\kappa\,\chi(Z,E,\kappa)\,{\rm d}\kappa, \tag{3.158}$$

respectively. Notice that these quantities are finite. For the kinetic energies E_i of the grid, they are easily calculated from the tabulated scaled DCS by using linear interpolation in κ. For positrons, the definitions (3.156)–(3.158) must be multiplied by the factor $F_{\rm p}(Z,E)$ [Eq. (3.154)]. The quantities $S_{\rm br}(E)$ and $\Omega_{\rm br}^2(E)$ give, respectively, the average energy loss per unit path length and the increase of variance of the energy-loss distribution of particles with kinetic energy E (see Section 4.2).

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 defined by Eq. (3.146) practically coincide with those given by Berger and Seltzer (1982) and in the ICRU Report 37 (1984). To leave room for future

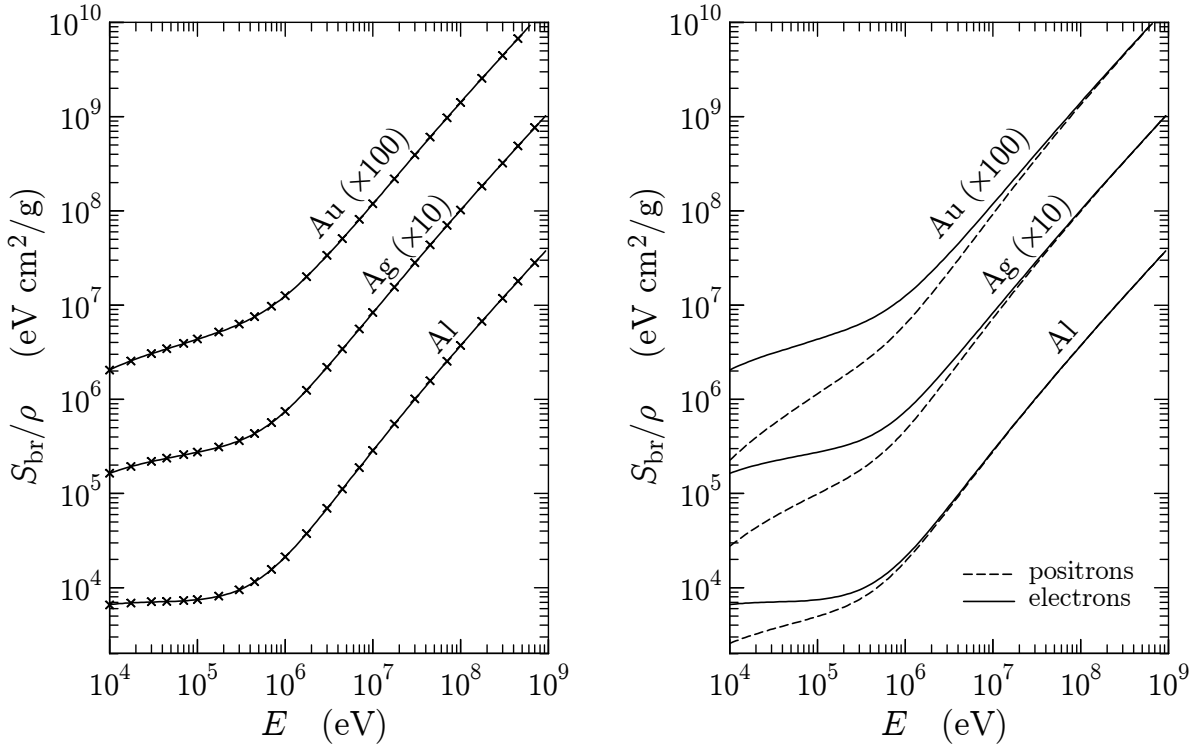

Figure 3.14: Radiative stopping power $S_{\rm br}/\rho$ for electrons and positrons in aluminium, silver (×10) and gold (×100) as a function of the kinetic energy. Solid and dashed curves are results from the present model. Crosses are data from the tables in the ICRU Report 37 (1984) and in Berger and Seltzer (1982). (Adapted from Salvat and Fernández-Varea, 1992).

improvements, PENELOPE reads the radiative stopping power for electrons from the input material data file, and renormalises the DCS, Eq. (3.146), (*i.e.*, multiplies it by a κ-independent factor) so as to exactly reproduce the input radiative stopping power.

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 $\mathrm{d}s$ within a medium, it loses an (average) energy $\mathrm{d}E = -S(E)\,\mathrm{d}s$, where

$$S(E) = S_{\rm in}(E) + S_{\rm br}(E) = -\frac{\mathrm{d}E}{\mathrm{d}s} \tag{3.159}$$

is the total (collisional+radiative) stopping power. Figure 3.15 compares the radiative and collision stopping powers of aluminium, silver and gold for electrons. It is seen that radiative losses dominate for projectiles with kinetic energies above a critical value, $E_{\rm crit}$ (indicated by diamonds in Fig. 3.15), which decreases with the atomic number. Electrons and positrons with energies well above $E_{\rm crit}$ tend to radiate high-energy photons, with a corresponding sudden reduction of the kinetic energy in each event. For energies higher than $E_{\rm crit}$ the radiative stopping power is approximately proportional to the energy,

$$S_{\rm br}(E) \simeq E/X_0 \tag{3.160}$$

where the distance X_0 is known as the *radiation length*. The values of this parameter, determined graphically from the $S_{\rm br}(E)$ curves in Fig. 3.15, are $\rho X_0 = 25$ g/cm^2 for aluminium, $\rho X_0 = 9.2$ g/cm^2 for silver, and $\rho X_0 = 6.6$ g/cm^2 for gold. Since the radiative stopping powers for electrons and positrons with very high energies are equal (see Fig. 3.14), these particles have the same radiation lengths. The energy straggling parameter of high-energy electrons and positrons is also dominated by radiative events.

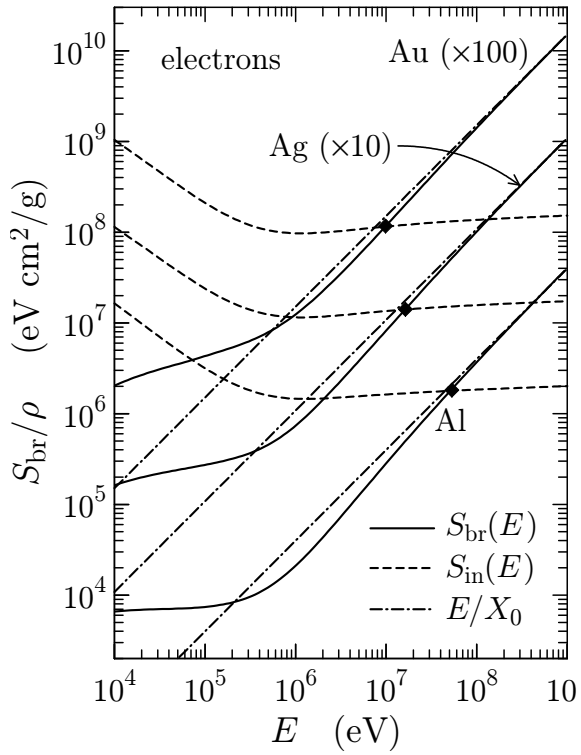

Figure 3.15: Radiative and collision stopping powers for electrons in aluminium, silver (×10) and gold (×100) as functions of the kinetic energy (solid and dashed curves, respectively). Dot-dashed lines represent the high-energy approximation given by Eq. (3.160). Diamonds indicate the critical energy $E_{\rm crit}$ at which the radiative stopping power starts dominating for each material.

3.3.2.1 CSDA range and radiative yield

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 provides only the average value of the energy lost by a particle along a given path length.

A quantity of much practical importance is the so-called *CSDA range* (or *Bethe range*), $R(E)$, which is defined as the average path length travelled by a particle of

kinetic energy E (in an infinite medium) in the course of its slowing down, *i.e.*, before being absorbed. It is given by

$$R(E) = \int \frac{\mathrm{d}s}{\mathrm{d}E'}\,\mathrm{d}E' = \int_{E_{\mathrm{abs}}}^{E} \frac{\mathrm{d}E'}{S(E')}, \tag{3.161}$$

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 3.16 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 method). Compare Fig. 3.16 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 in the CSDA ranges of these particles.

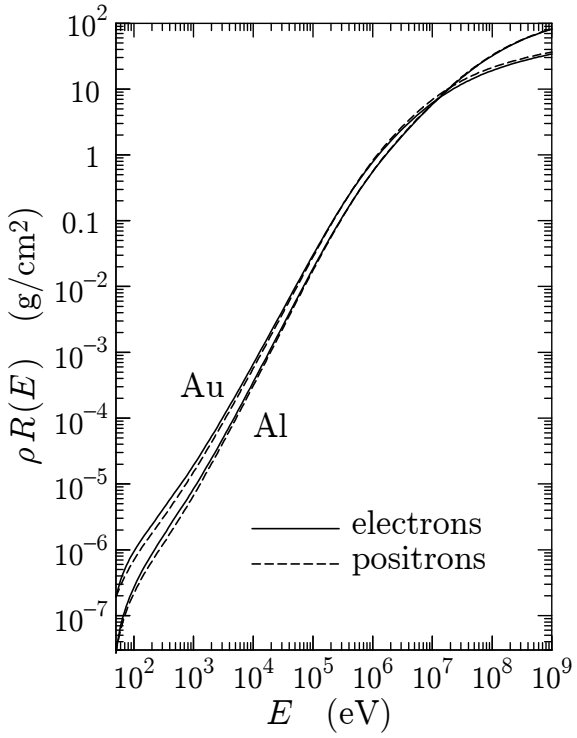

Figure 3.16: CSDA ranges for electrons and positrons in aluminium and gold as functions of the kinetic energy of the particle.

The CSDA can also be used to estimate the stopping time of electrons and positrons. Assuming that particles are effectively absorbed when their kinetic energy reaches the value $E_{\mathrm{abs}} = 50$ eV, the lower limit of the energy range covered by the PENELOPE database, the average time of flight of an electron or positron with initial kinetic energy

E is

$$t_{\text{stop}} = \int \frac{\mathrm{d}s}{v} = \int_{E_{\text{abs}}}^{E} \frac{1}{v(E)} \frac{\mathrm{d}E}{S(E)} \,, \tag{3.162}$$

where $v(E) = \beta c$ is the (energy-dependent) speed of a particle with kinetic energy E [see Eq. (A.7)].

The CSDA sets a correspondence between the travelled path length s and the average energy E, which takes a simple form when the energy of the projectile is well above the critical energy E_{crit}. Under such conditions, radiative energy losses dominate and the approximation (3.160) applies. That is,

$$S(E) \simeq S_{\text{br}}(E) \simeq E/X_0 \,. \tag{3.163}$$

Electrons with initial energy $E(0)$ after travelling a path length s acquire an average energy $E(s)$. The relation

$$s = \int_{E(s)}^{E(0)} \frac{\mathrm{d}E'}{S(E')} \simeq X_0 \ln\left(\frac{E(0)}{E(s)}\right) \tag{3.164}$$

implies that

$$E(s) = E(0) \exp(-s/X_0) \,. \tag{3.165}$$

Hence, when the electron travels a distance equal to the radiation length X_0 the energy drops by a factor e ~ 2.71. The importance of the radiation length is due to this straightforward physical interpretation.

The *radiative yield* or *bremsstrahlung yield*, $Y_{\text{br}}(E)$, is the average fraction of the initial kinetic energy E of an electron that is emitted as bremsstrahlung photons when the electron comes to rest, assuming that the medium where it moves is unbounded. The average energy emitted as radiation is

$$E_{\text{br}} = \int_0^{R(E)} \left(\frac{\mathrm{d}E'}{\mathrm{d}s}\right)_{\text{br}} \mathrm{d}s = \int_0^{E} \left(\frac{\mathrm{d}E'}{\mathrm{d}s}\right)_{\text{br}} \frac{\mathrm{d}s}{\mathrm{d}E'} \,\mathrm{d}E' = \int_0^{E} \frac{S_{\text{br}}(E')}{S(E')} \,\mathrm{d}E'. \tag{3.166}$$

Hence,

$$Y_{\text{br}}(E) = \frac{E_{\text{br}}}{E} = \frac{1}{E} \int_0^{E} \frac{S_{\text{br}}(E')}{S(E')} \,\mathrm{d}E' \,. \tag{3.167}$$

Knowledge of the radiative yield is useful, *e.g.*, to calculate the energy taken by bremsstrahlung photons away from the active volume of an electron detector. Figure 3.17 displays radiative yields of electrons and positrons in carbon, aluminium, silver and gold, as functions of the kinetic energy E. It shows that $Y_{\text{br}}(E)$ increases with the atomic number of the medium, and is roughly proportional to Z. The radiative yields of positrons are smaller than those of electrons, because of their smaller stopping power (see Fig. 3.14). The relative differences increase with atomic number, and decrease when the kinetic energy of the particle increases.

A related quantity is the *photon number yield*, $Y_{\text{ph}}(E; W_{\text{cr}})$, which is defined as the average number of photons having energy W larger than the given cutoff energy, W_{cr},

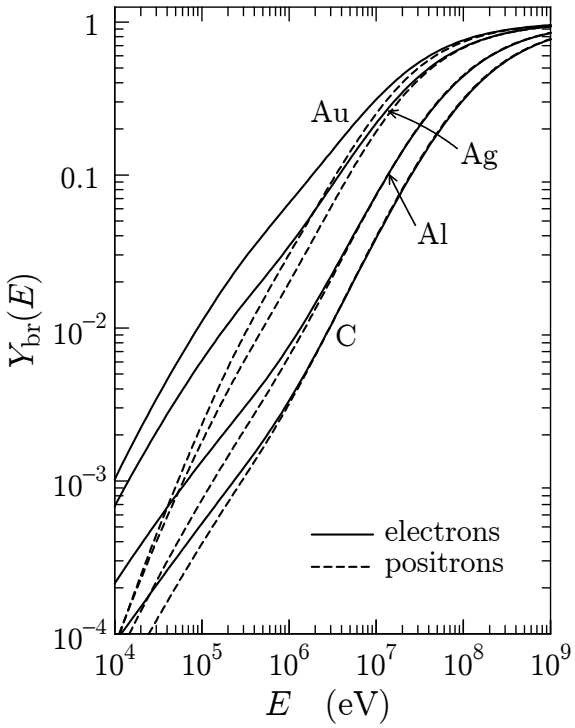

Figure 3.17: Radiative yields of electrons and positrons in carbon, aluminium, silver and gold.

that are emitted during the slowing down of a particle with initial kinetic energy E. It can be calculated as follows,

$$Y_{\rm ph}(E;W_{\rm cr}) = \int_0^{R(E)} \lambda_{\rm br}^{-1}(E';W_{\rm cr})\,{\rm d}s = \int_{E_{\rm abs}}^{E} \frac{\lambda_{\rm br}^{-1}(E';W_{\rm cr})}{S(E')}\,{\rm d}E' \,. \tag{3.168}$$

Note that the number of photons emitted with energies W in an interval (W_1, W_2) can be obtained as the difference between the corresponding photon number yields, *i.e.*, $Y_{\rm ph}(E;W_1) - Y_{\rm ph}(E;W_2)$. That is, from $Y_{\rm ph}(E;W_{\rm cr})$ we can build the energy spectrum of *all* photons emitted by a particle with initial kinetic energy E when it comes to rest. From the photon number yield we can also obtain the average number $N_{\rm ph}$ of photons emitted with energy $W > W_{\rm cr}$ when a particle slows from an initial energy E_0 down to an arbitrary final energy E, lower than E_0,

$$N_{\rm ph} = Y_{\rm ph}(E_0;W_{\rm cr}) - Y_{\rm ph}(E;W_{\rm cr})\,. \tag{3.169}$$

Photon number yields of electrons and positrons in aluminium and gold, for various cutoff energies $W_{\rm cr}$, are displayed in Fig. 3.18. Again, the yields are smaller for positrons than for electrons, and the differences are larger for heavier elements.

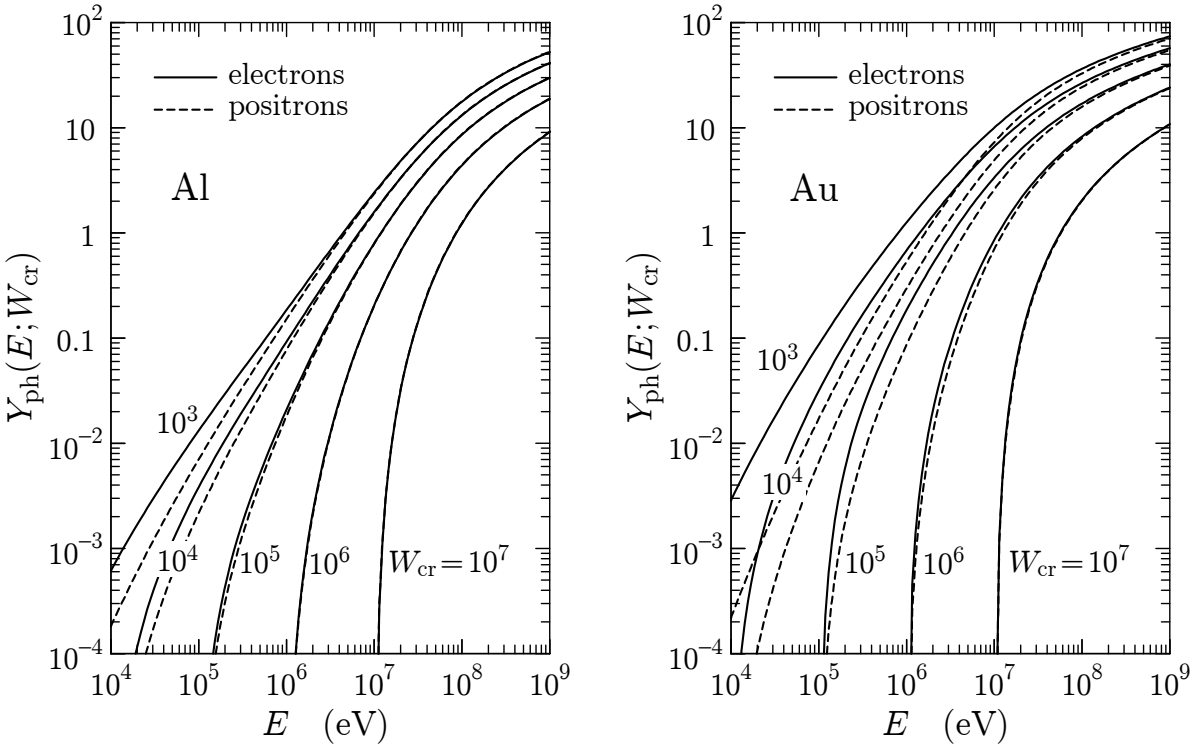

Figure 3.18: Photon number yields of electrons and positrons in aluminium and gold, for the indicated values of the cutoff energy $W_{\rm cr}$, as functions of the kinetic energy E.

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

$$\frac{{\rm d}^2\sigma_{\rm br}}{{\rm d}W\,{\rm d}(\cos\theta)} = \frac{{\rm d}\sigma_{\rm br}}{{\rm d}W}\,p_{\rm br}(Z,E,\kappa;\cos\theta) = \frac{Z^2}{\beta^2}\,\frac{1}{W}\,\chi(Z,E,\kappa)\,p_{\rm br}(Z,E,\kappa;\cos\theta), \quad (3.170)$$

where $p_{\rm br}(Z,E,\kappa;\cos\theta)$ is the PDF of $\cos\theta$.

Numerical values of the "shape function" $p_{\rm br}(Z,E,\kappa;\cos\theta)$, calculated by partial-wave methods, were 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 κ = 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 well suited for random sampling of the photon direction. In PENELOPE we use a different parameterisation of the shape function [Eq. (3.175) below], defined by two parameters A and B, that allows the random sampling of $\cos\theta$ in a simple way. In previous versions of the code, the parameters A and B were determined, by least squares fitting, for the 144 benchmark shape functions given by Kissel *et al.* (1983). These parameters were found to vary smoothly with Z,

β and κ and they were calculated by cubic spline interpolations of their values for the benchmark cases.

Interpolation with such a sparse grid may introduce relatively large uncertainties, which were difficult to estimate from only the information in the tables of Kissel *et al.* Recently, Poškus (2018) has developed a robust computer code that calculates bremsstrahlung DCSs, using the same theoretical approach as Kissel *et al.* (1983), for electrons with kinetic energies up to 3 MeV. Results from this code practically coincide with the benchmark data of Kissel *et al.* To reduce the interpolation uncertainties, we have run Poškus' program and generated shape functions for the following expanded grid of elements and energies: Z = 1, 2, 5, 8, 11, 13, 26, 37, 47, 64, 79, 86,92; E = 1, 5, 10, 50, 100, 500, 1,000 keV; and κ = 0, 0.1, 0.2, 0.3, 0.4, 0.5 0.6, 0.7, 0.8, 0.95. The calculations were performed for a screened atomic potential, with the screening function approximated by a sum of three exponential terms with parameters determined by fitting the self-consistent Dirac–Hartree–Fock–Slater potential of neutral atoms (Salvat *et al.*, 1987).

3.3.3.1 Analytical shape functions

Since the use of numerical shape functions would increase the size of the database and complicate the sampling algorithms, we sought an analytical form that were flexible enough to approximate the numerical shape functions and, at the same time, well suited for random sampling of $\cos\theta$.

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 formulas originally derived by Kirkpatrick and Wiedmann (1945) [and subsequently modified by Statham (1976]. These formulas were obtained by fitting the bremsstrahlung DCS derived from Sommerfeld's theory. The shape function obtained from the Kirkpatrick–Wiedmann–Statham fit reads

$$p_{\mathrm{br}}^{(\mathrm{KWS})}(Z,E,\kappa;\cos\theta)=\frac{\sigma_x(1-\cos^2\theta)+\sigma_y(1+\cos^2\theta)}{(1-\beta\cos\theta)^2}, \tag{3.171}$$

where the quantities σ_x and σ_y are independent of θ. Although this simple formula predicts the global trends of the numerical shape functions obtained from partial-wave calculations 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 (3.171) is flexible enough to approximate the "true" (partial-wave) shape.

The analytical form (3.171) 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 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′,

$$p_{\mathrm{br,d}}(\cos\theta') = A\frac{3}{8}\left(1+\cos^2\theta'\right) + (1-A)\frac{3}{4}\left(1-\cos^2\theta'\right), \qquad (0 \le A \le 1), \tag{3.172}$$

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

$$\cos\theta = \frac{\cos\theta' + \beta}{1+\beta\cos\theta'}, \qquad \phi = \phi'. \tag{3.173}$$

Thus, the angular distribution in K reads

$$\begin{aligned} p_{\mathrm{br}}(\cos\theta) &= p_{\mathrm{br,d}}(\cos\theta')\,\frac{\mathrm{d}(\cos\theta')}{\mathrm{d}(\cos\theta)} \\ &= A\frac{3}{8}\left[1+\left(\frac{\cos\theta-\beta}{1-\beta\cos\theta}\right)^2\right]\frac{1-\beta^2}{(1-\beta\cos\theta)^2} \\ &+ (1-A)\frac{3}{4}\left[1-\left(\frac{\cos\theta-\beta}{1-\beta\cos\theta}\right)^2\right]\frac{1-\beta^2}{(1-\beta\cos\theta)^2}. \end{aligned} \tag{3.174}$$

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 (3.174) if one considers A and β as adjustable parameters (Acosta *et al.*, 2002). Explicitly, we write

$$\begin{aligned} p_{\mathrm{br,fit}}(\cos\theta) &= A\,\frac{3}{8}\left[1+\left(\frac{\cos\theta-B}{1-B\,\cos\theta}\right)^2\right]\frac{1-B}{(1-B\,\cos\theta)^2} \\ &+ (1-A)\,\frac{3}{4}\left[1-\left(\frac{\cos\theta-B}{1-B\,\cos\theta}\right)^2\right]\frac{1-B^2}{(1-B\,\cos\theta)^2}, \end{aligned} \tag{3.175}$$

where B is expected to be close to β. The parameters A and B have been determined, by least squares fitting, for the 910 combinations of atomic number, electron energy and reduced photon energy of the benchmark shape functions calculated by Poškus code. Results of this fit are compared with the original partial-wave shape functions in Fig. 3.19. The largest differences between the fits and the data were found for the

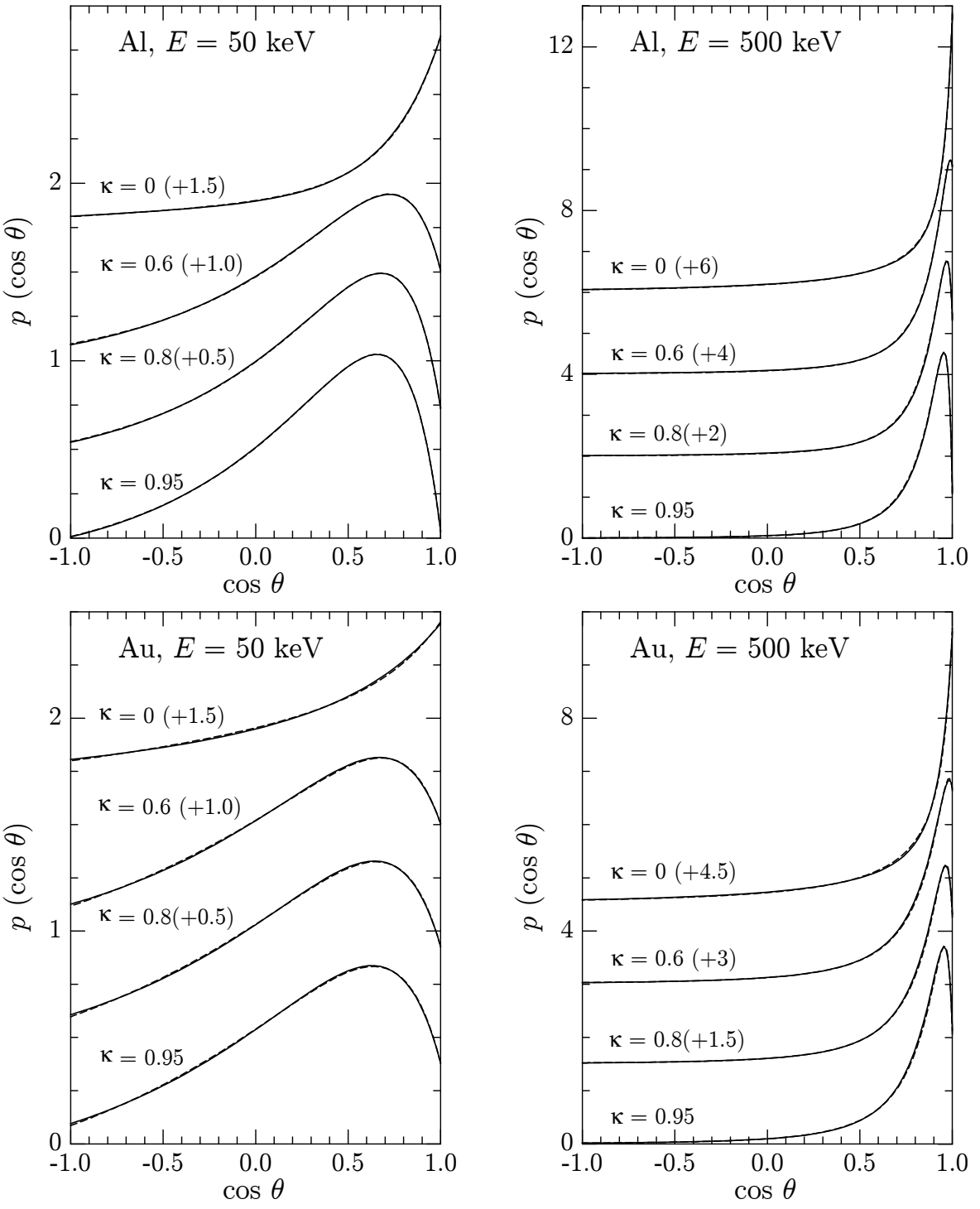

Figure 3.19: Shape functions (angular distributions) for bremsstrahlung emission by electrons of the indicated energies in the fields of aluminium and gold atoms. Dashed curves are partial-wave shape functions of Kissel *et al.* (1983). Continuous curves are the present analytical fits, Eq. (3.175). For visual aid, some curves have been shifted upwards in the amounts indicated in parentheses. (Adapted from Salvat *et al.*, 2006).

higher atomic numbers, but even then the fits are very accurate, as shown in Fig. 3.19. The quantities $\ln(AZ)$ and B vary smoothly with Z, β and κ, and will be obtained by interpolation of their values for the benchmark cases. This enables rapid evaluation of the shape function for any combination of Z, β and κ. Furthermore, 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 1 MeV, the shape function is approximated by the classical dipole distribution, *i.e.*, by the analytical form (3.175) with $A = 1$ and $B = \beta$.

As indicated above, in the case of compounds, the adopted angular distributions are those of the "equivalent" atomic number $Z_{\rm eq}$ given by Eq. (3.151). For the electron energies and the reduced photon energies of the benchmark cases, the quantity $\ln(AZ)$ varies smoothly with Z so that A can be calculated by cubic spline interpolation of $\ln(AZ)$ in Z. The parameter B is generally close to β, but for a given E and κ near 0 or 1, the difference $B - \beta$ may vary rapidly with Z. To avoid uncontrolled wiggling of cubic splines, B is evaluated by linear interpolation in Z.

The simulation code starts from a table of 70 values of $\ln(A)$ and B for the equivalent atomic number $Z_{\rm eq}$ of the material, at the seven electron energies (corresponding to $\beta = 0.0625, 0.139, 0.195, 0.413, 0.548, 0.863, 0.941$) and the ten reduced photon energies of the benchmark cases. In preparation of the sampling algorithm, expanded tables of these quantities are generated for a denser grid of 41 values of κ [$\kappa_i = (i-1)0.025$] by natural cubic spline interpolation in κ. During simulation, for given values of the projectile energy E and the reduced photon energy κ, the parameters A and B are determined by interpolation in this table. The interpolation scheme is as follows: first the values of $\ln A$ and B and κ corresponding to the energy E and to the reduced photon energies κ_i and κ_{i+1} with $i = 40\,\kappa + 1$ are calculated by cubic spline interpolation in β, and then $\ln A$ and B at E,κ are evaluated by linear interpolation in κ. Owing to the lack of numerical data for positrons, it is assumed that the shape function for positrons is the same as for electrons with equal energy.

3.3.4 Simulation of hard radiative events

Let us now consider simulation of hard radiative events ($W > W_{\rm cr}$) from the DCS defined by Eqs. (3.170) and (3.175). 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.

3.3.4.1 Sampling of the photon energy

The PDF for the reduced photon energy, $\kappa = W/E$, is given by [see Eq. (3.146)]

$$p_{\rm br}(E,\kappa) = \frac{1}{\kappa}\,\chi(Z,E,\kappa)\,\Theta(\kappa-\kappa_{\rm cr})\,\Theta(1-\kappa), \tag{3.176}$$

where $\kappa_{\rm cr} = W_{\rm cr}/E$ 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 (3.176) for an energy E_i in the grid, we express the interpolated scaled DCS as

$$\chi(Z,E_i,\kappa) = a_j + b_j\kappa \qquad \text{if } \kappa_j \le \kappa \le \kappa_{j+1}, \tag{3.177}$$

and introduce the cumulative distribution function,

$$\mathcal{P}_j = \int_{\kappa_{\rm cr}}^{\kappa_j} p(E_i,\kappa)\,{\rm d}\kappa, \tag{3.178}$$

which, for a piecewise linear χ, can be computed exactly. We also define

$$\chi_{{\rm max},j} = \max\Big\{\chi(Z,E,\kappa), \kappa\in(\kappa_j,\kappa_{j+1})\Big\} \qquad j=1,\ldots,32. \tag{3.179}$$

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 $\mathcal{P}_j \le \xi\mathcal{P}_{32} \le \mathcal{P}_{j+1}$ using the binary-search method.

(ii) Sample κ from the distribution κ^{-1} in the interval (κ_j,κ_{j+1}), *i.e.*,

$$\kappa = \kappa_j\,(\kappa_{j+1}/\kappa_j)^{\xi}\,. \tag{3.180}$$

(iii) Generate a new random number ξ. If $\xi\chi_{{\rm max},j} < a_j + b_j\kappa$, 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 $p_{\rm br}(E,\kappa)$ by interpolation along the energy axis and then perform 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)]

$$p_{\mathrm{br,int}}(E,\kappa) = \pi_i\, p_{\mathrm{br}}(E_i,\kappa) + \pi_{i+1}\, p_{\mathrm{br}}(E_{i+1},\kappa) \tag{3.181}$$

with the interpolation weights

$$\pi_i = \frac{\ln E_{i+1} - \ln E}{\ln E_{i+1} - \ln E_i}, \qquad \pi_{i+1} = \frac{\ln E - \ln E_i}{\ln E_{i+1} - \ln E_i}. \tag{3.182}$$

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 $p_{\mathrm{br,int}}(E,\kappa)$ 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 $p_{\mathrm{br}}(E_k,\kappa)$.

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. (3.175) results from a Lorentz transformation, with speed B, of the PDF (3.172). This means that we can sample the photon direction $\cos\theta'$ in the reference frame K′ from the PDF (3.172) and then apply the transformation (3.173) (with B instead of β) to get the direction $\cos\theta$ in the laboratory frame.

To generate random values of $\cos\theta$ from (3.175) we use the following algorithm, which combines 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 $\cos\theta = \dfrac{\cos\theta' + B}{1 + B\,\cos\theta'}$.

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 $E+2\mathrm{m_e}c^2$. 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

$$\zeta \equiv \frac{E_-}{E+2\mathrm{m_e}c^2}. \tag{3.183}$$

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]

$$\cos\theta_- = (\gamma^2-1)^{-1/2}(\gamma+1-1/\zeta) \tag{3.184}$$

and

$$\cos\theta_+ = (\gamma^2-1)^{-1/2}[\gamma+1-1/(1-\zeta)], \tag{3.185}$$

and azimuthal angles ϕ_- and $\phi_+ = \phi_- + \pi$. The quantity $\gamma = 1 + E/(\mathrm{m_e}c^2)$ 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

$$\zeta_{\min} = \frac{1}{\gamma+1+(\gamma^2-1)^{1/2}}. \tag{3.186}$$

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

$$\frac{\mathrm{d}\sigma_{\mathrm{an}}}{\mathrm{d}\zeta} = \frac{\pi r_{\mathrm{e}}^2}{(\gamma+1)(\gamma^2-1)}\left[S(\zeta)+S(1-\zeta)\right], \tag{3.187}$$

where

$$S(\zeta) = -(\gamma+1)^2 + (\gamma^2+4\gamma+1)\frac{1}{\zeta} - \frac{1}{\zeta^2}. \tag{3.188}$$

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 isotropic (see Fig. 3.20).

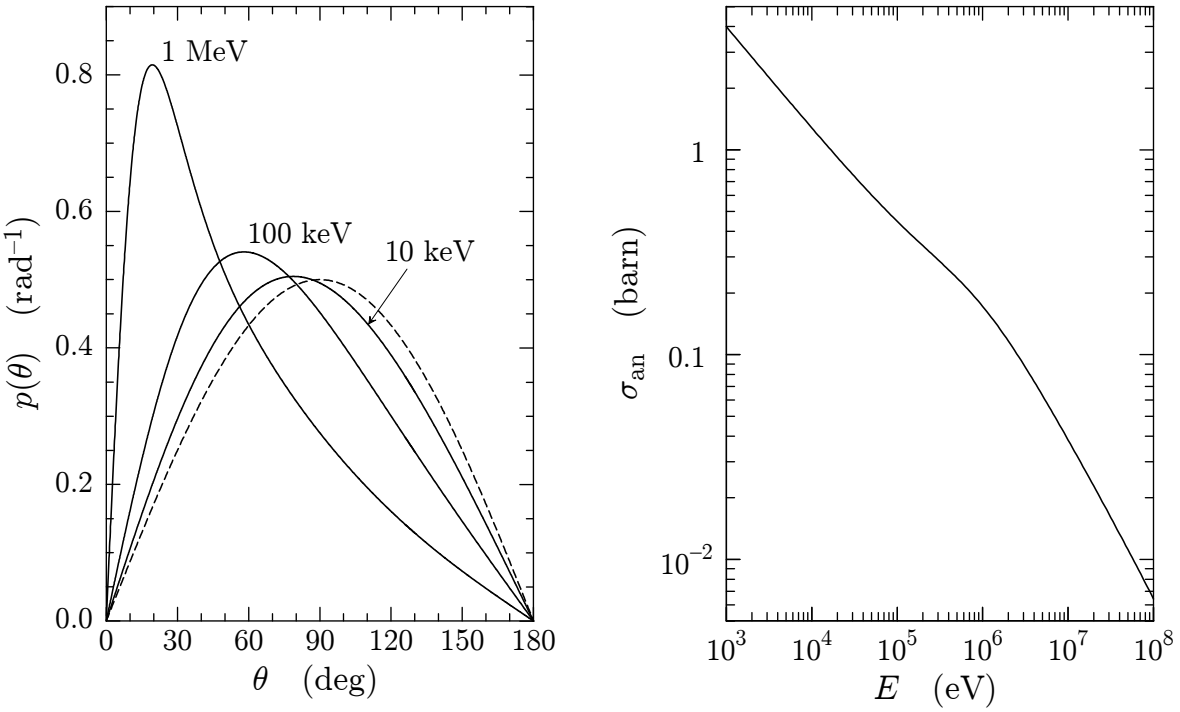

Figure 3.20: Left: angular distributions of photons produced by annihilation in flight of positrons with the indicated kinetic energies. The dashed line represents the isotropic distribution. Right: Annihilation cross section per target electron as a function of the kinetic energy of the positron.

The cross section (per target electron) for two-photon annihilation is

$$\sigma_{\rm an} = \int_{\zeta_{\rm min}}^{1/2} \frac{{\rm d}\sigma_{\rm an}}{{\rm d}\zeta}\,{\rm d}\zeta = \frac{\pi r_{\rm e}^2}{(\gamma+1)(\gamma^2-1)} \times \left\{(\gamma^2+4\gamma+1)\ln\left[\gamma+\left(\gamma^2-1\right)^{1/2}\right] - (3+\gamma)\left(\gamma^2-1\right)^{1/2}\right\}. \quad (3.189)$$

The annihilation mean free path is given by

$$\lambda_{\rm an}^{-1} = \mathcal{N} Z \sigma_{\rm an}, \quad (3.190)$$

where $\mathcal{N}Z$ is the density of electrons in the medium. The annihilation cross section is displayed in Fig. 3.20. 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)

$$p_{\rm an}(\zeta) = S(\zeta) + S(1-\zeta), \qquad \zeta_{\rm min} \leq \zeta \leq 1/2. \tag{3.191}$$

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

$$P(\upsilon) \equiv S(\upsilon), \qquad \zeta_{\rm min} \leq \upsilon \leq 1-\zeta_{\rm min} \tag{3.192}$$

and write it in the form

$$P(\upsilon) = \pi(\upsilon) g(\upsilon) \tag{3.193}$$

with

$$\pi(\upsilon) = \left[\ln\left(\frac{1-\zeta_{\rm min}}{\zeta_{\rm min}}\right)\right]^{-1} \frac{1}{\upsilon} \tag{3.194}$$

and

$$g(\upsilon) = \left[-(\gamma+1)^2\upsilon + (\gamma^2+4\gamma+1) - \frac{1}{\upsilon}\right]. \tag{3.195}$$

$\pi(\upsilon)$ is a proper PDF (*i.e.*, it is definite positive and normalised to unity) and $g(\upsilon)$ is a monotonically decreasing function. Random values of υ from the distribution $P(\upsilon)$ can be generated by using the following algorithm (rejection method):

(i) Sample a value υ from the distribution $\pi(\upsilon)$. This is easily done with the inverse-transform method, which yields the following sampling equation

$$\upsilon = \zeta_{\rm min}\left(\frac{1-\zeta_{\rm min}}{\zeta_{\rm min}}\right)^{\xi}. \tag{3.196}$$

(ii) Generate a new random number ξ.

(iii) If $\xi g(\zeta_{\rm min}) > g(\upsilon)$, go to step (i).

(iv) Deliver υ.

It is clear that the random value

$$\zeta = \min(\upsilon, 1-\upsilon) \tag{3.197}$$

follows the distribution given by Eq. (3.191) when υ is sampled from the distribution $P(\upsilon)$. 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 $E_- = \zeta(E + 2{\rm m_e}c^2)$ and $E_+ = (1-\zeta)(E+2{\rm m_e}c^2)$ are emitted in the directions given by Eqs. (3.184) and (3.185).

Chapter 4

Electron/positron transport mechanics

In principle, the scattering model and sampling techniques described in Chapter 3 allow 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 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 θ_c and W_c 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)

$$\frac{\partial f}{\partial s}+\hat{\mathbf{d}}\cdot\nabla f=\mathcal{N}\int\left[f(s;\mathbf{r},\hat{\mathbf{d}}')-f(s;\mathbf{r},\hat{\mathbf{d}})\right]\frac{\mathrm{d}\sigma_{\mathrm{el}}(\theta)}{\mathrm{d}\Omega}\,\mathrm{d}\Omega, \tag{4.1}$$

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 $\mathrm{d}\Omega$ around a direction defined by the polar angle χ is given by

$$F(s;\chi)=\int f(s;\mathbf{r},\hat{\mathbf{d}})\,\mathrm{d}\mathbf{r}=\sum_{\ell=0}^{\infty}\frac{2\ell+1}{4\pi}\exp(-s/\lambda_{\mathrm{el},\ell})P_\ell(\cos\chi), \tag{4.2}$$

where $P_\ell(\cos\chi)$ are Legendre polynomials and $\lambda_{\mathrm{el},\ell}=1/(\mathcal{N}\sigma_{\mathrm{el},\ell})$ 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,b). 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

$$\langle P_\ell(\cos\chi)\rangle\equiv 2\pi\int_{-1}^{1}P_\ell(\cos\chi)F(s;\chi)\,\mathrm{d}(\cos\chi)=\exp(-s/\lambda_{\mathrm{el},\ell}). \tag{4.3}$$

In particular, we have

$$\langle\cos\chi\rangle=\exp(-s/\lambda_{\mathrm{el},1}) \tag{4.4}$$

and

$$\langle\cos^2\chi\rangle=\frac{1}{3}\left[1+2\exp(-s/\lambda_{\mathrm{el},2})\right]. \tag{4.5}$$

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

$$\langle z\rangle \equiv 2\pi\int z f(s;\mathbf{r},\hat{\mathbf{d}})\,\mathrm{d}(\cos\chi)\,\mathrm{d}\mathbf{r} = \lambda_{\mathrm{el},1}\left[1-\exp(-s/\lambda_{\mathrm{el},1})\right], \tag{4.6}$$

$$\begin{aligned}\langle x^2+y^2\rangle &\equiv 2\pi\int \left(x^2+y^2\right) f(s;\mathbf{r},\hat{\mathbf{d}})\,\mathrm{d}(\cos\chi)\,\mathrm{d}\mathbf{r}\\ &= \frac{4}{3}\int_0^s \mathrm{d}t\,\exp(-t/\lambda_{\mathrm{el},1})\int_0^t\left[1-\exp(-u/\lambda_{\mathrm{el},2})\right]\exp(u/\lambda_{\mathrm{el},1})\,\mathrm{d}u,\end{aligned} \tag{4.7}$$

$$\begin{aligned}\langle z\cos\chi\rangle &\equiv 2\pi\int z\cos\chi f(s;\mathbf{r},\hat{\mathbf{d}})\,\mathrm{d}(\cos\chi)\,\mathrm{d}\mathbf{r}\\ &= \exp(-s/\lambda_{\mathrm{el},1})\int_0^s\left[1+2\exp(-t/\lambda_{\mathrm{el},2})\right]\exp(t/\lambda_{\mathrm{el},1})\,\mathrm{d}t.\end{aligned} \tag{4.8}$$

It is worth observing that the quantities (4.4)–(4.8) are completely determined by the values of the transport mean free paths $\lambda_{\mathrm{el},1}$ and $\lambda_{\mathrm{el},2}$; 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, $\lambda_{\mathrm{el},1} \gg \lambda_{\mathrm{el}}$ (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.*, 1993a; 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 $\theta<\theta_{\mathrm{c}}$) are described by means of a multiple-scattering approach.

In practice, the mixed algorithm will be defined by specifying the mean free path $\lambda_{\mathrm{el}}^{(\mathrm{h})}$ between hard elastic events, defined by [see Eq. (3.25)]

$$\frac{1}{\lambda_{\mathrm{el}}^{(\mathrm{h})}} = \mathcal{N}2\pi\int_{\theta_{\mathrm{c}}}^{\pi}\frac{\mathrm{d}\sigma_{\mathrm{el}}(\theta)}{\mathrm{d}\Omega}\sin\theta\,\mathrm{d}\theta. \tag{4.9}$$

This equation determines the cutoff angle θ_{c} as a function of $\lambda_{\mathrm{el}}^{(\mathrm{h})}$. A convenient recipe to set the mean free path $\lambda_{\mathrm{el}}^{(\mathrm{h})}$ is

$$\lambda_{\mathrm{el}}^{(\mathrm{h})}(E) = \max\left\{\lambda_{\mathrm{el}}(E), C_1\lambda_{\mathrm{el},1}(E)\right\}, \tag{4.10}$$

where C_1 is a pre-selected small constant (say, less than ~ 0.1). For increasing energies, λ_{el} attains a constant value and $\lambda_{\mathrm{el},1}$ 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 $\lambda_{\rm el}^{\rm (h)}$ will reduce to the actual mean free path $\lambda_{\rm el}$ for low energies. In this case, soft collisions cease to occur ($\theta_{\rm c}=0$) and mixed simulation becomes purely detailed. It is worth noticing that, when mixed simulation is effective (*i.e.*, when $\lambda_{\rm el}^{\rm (h)} > \lambda_{\rm el}$), the mean angular deflection in a path length $\lambda_{\rm el}^{\rm (h)}$ is [see Eq. (4.4)]

$$1-\langle\cos\chi\rangle = 1-\exp(-\lambda_{\rm el}^{\rm (h)}/\lambda_{\rm el,1}) \simeq C_1. \tag{4.11}$$

Hence, when using the prescription (4.10), the average angular deflection due to all elastic collisions occurring along a path length $\lambda_{\rm el}^{\rm (h)}$ equals C_1.

The PDF of the step length s between two successive hard collisions is

$$p(s) = \frac{1}{\lambda_{\rm el}^{\rm (h)}}\exp(-s/\lambda_{\rm el}^{\rm (h)}), \tag{4.12}$$

and random values of s can be generated by means of the sampling formula, Eq. (1.36)

$$s = -\lambda_{\rm el}^{\rm (h)}\ln\xi. \tag{4.13}$$

The (unnormalised) PDF of the polar deflection θ in single hard collisions is

$$p^{\rm (h)}(\theta) = \frac{{\rm d}\sigma_{\rm el}(\theta)}{{\rm d}\Omega}\sin\theta\,\Theta(\theta-\theta_{\rm c}), \tag{4.14}$$

where $\Theta(x)$ stands for the step function.

The inverse transport mean free paths $\lambda_{{\rm el},\ell}^{-1}$, see Eqs. (3.14) and (3.16), for the actual scattering process can be split into contributions from soft and hard collisions, that is

$$\frac{1}{\lambda_{{\rm el},\ell}} = \frac{1}{\lambda_{{\rm el},\ell}^{\rm (s)}} + \frac{1}{\lambda_{{\rm el},\ell}^{\rm (h)}}, \tag{4.15}$$

where

$$\frac{1}{\lambda_{{\rm el},\ell}^{\rm (s)}} = \mathcal{N}2\pi\int_0^{\theta_{\rm c}}[1-P_\ell(\cos\theta)]\frac{{\rm d}\sigma_{\rm el}(\theta)}{{\rm d}\Omega}\sin\theta\,{\rm d}\theta \tag{4.16a}$$

and

$$\frac{1}{\lambda_{{\rm el},\ell}^{\rm (h)}} = \mathcal{N}2\pi\int_{\theta_{\rm c}}^{\pi}[1-P_\ell(\cos\theta)]\frac{{\rm d}\sigma_{\rm el}(\theta)}{{\rm d}\Omega}\sin\theta\,{\rm d}\theta. \tag{4.16b}$$

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

$$F^{\rm (s)}(s;\chi) = \sum_{\ell=0}^{\infty}\frac{2\ell+1}{4\pi}\exp(-s/\lambda_{{\rm el},\ell}^{\rm (s)})P_\ell(\cos\chi). \tag{4.17}$$

The exact average longitudinal and transverse displacements at the end of the step are given by [see Eqs. (4.6) and (4.7)]

$$\langle z\rangle^{(\mathrm{s})} = \lambda_{\mathrm{el},1}^{(\mathrm{s})}\left[1-\exp(-s/\lambda_{\mathrm{el},1}^{(\mathrm{s})})\right] = s\left[1-\frac{1}{2}\left(\frac{s}{\lambda_{\mathrm{el},1}^{(\mathrm{s})}}\right)+\frac{1}{6}\left(\frac{s}{\lambda_{\mathrm{el},1}^{(\mathrm{s})}}\right)^2-\ldots\right], \tag{4.18}$$

$$\langle x^2+y^2\rangle^{(\mathrm{s})} = \frac{2}{9}\frac{s^3}{\lambda_{\mathrm{el},2}^{(\mathrm{s})}}\left[1-\frac{1}{4}\left(1+\frac{\lambda_{\mathrm{el},1}^{(\mathrm{s})}}{\lambda_{\mathrm{el},2}^{(\mathrm{s})}}\right)\left(\frac{s}{\lambda_{\mathrm{el},1}^{(\mathrm{s})}}\right)+\ldots\right], \tag{4.19}$$

where $\lambda_{\mathrm{el},1}^{(\mathrm{s})}$, the first transport mean free path for soft collisions, is larger than $\lambda_{\mathrm{el},1}$. As the mean free path between hard collisions is normally much less than $\lambda_{\mathrm{el},1}^{(\mathrm{s})}$ (depending on the value of C_1), the value $s/\lambda_{\mathrm{el},1}^{(\mathrm{s})}$ is, on average, much less than unity (note that $\langle s\rangle = \lambda_{\mathrm{el}}^{(\mathrm{h})}$). Therefore, the global effect of the soft collisions in the step, *i.e.*, the change in direction of movement *and* lateral displacement, is relatively small (because part of the deflection is caused by the hard interaction at the end of the step, which is simulated *exactly*).

In PENELOPE, the angular deflection and lateral displacement due to the multiple soft collisions in a step of length s are simulated by means of the random-hinge method[1] (Fernández-Varea *et al.*, 1993a; Salvat, 2015). 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) A single artificial soft scattering event (a hinge) then takes place, in which the electron changes its direction of movement according to the multiple-scattering distribution $F^{(\mathrm{s})}(s;\chi)$.

(iii) Finally, the electron moves a distance $s-\tau$ in the new direction.

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

$$\langle z\rangle_{\mathrm{sim}}^{(\mathrm{s})} = \frac{s}{2}+\frac{s}{2}\langle\cos\chi\rangle^{(\mathrm{s})} = s\left[1-\frac{1}{2}\left(\frac{s}{\lambda_{\mathrm{el},1}^{(\mathrm{s})}}\right)+\frac{1}{4}\left(\frac{s}{\lambda_{\mathrm{el},1}^{(\mathrm{s})}}\right)^2-\ldots\right], \tag{4.20}$$

which agrees closely with the exact result given by Eq. (4.18). Furthermore, the average simulated transverse displacement is

$$\begin{aligned}\langle x^2+y^2\rangle_{\mathrm{sim}}^{(\mathrm{s})} &= \langle(s-\tau)^2\sin^2\chi\rangle_{\mathrm{sim}}^{(\mathrm{s})} = \frac{1}{3}s^2\left(1-\langle\cos^2\chi\rangle^{(\mathrm{s})}\right)\\ &= \frac{2}{9}\frac{s^3}{\lambda_{\mathrm{el},2}^{(\mathrm{s})}}\left[1-\frac{1}{2}\frac{\lambda_{\mathrm{el},1}^{(\mathrm{s})}}{\lambda_{\mathrm{el},2}^{(\mathrm{s})}}\left(\frac{s}{\lambda_{\mathrm{el},1}^{(\mathrm{s})}}\right)+\ldots\right],\end{aligned} \tag{4.21}$$

[1] The name was coined by Ron Kensek.

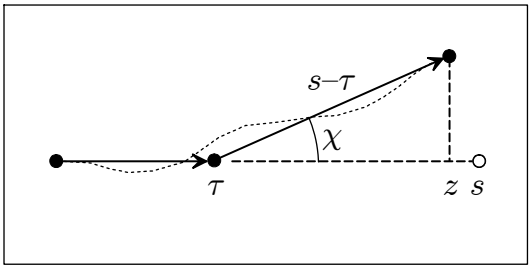

Figure 4.1: Simulation of the global effect of soft collisions between two consecutive hard collisions by the random-hinge method. (Adapted from Fernández-Varea *et al.*, 1993a).

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 transport when the step length s is much shorter than the first transport mean free path $\lambda_{\mathrm{el},1}$ so that the global angular deflection and lateral displacement are small. Surprisingly, the method 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.

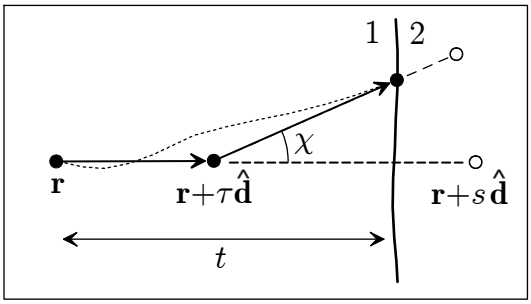

Figure 4.2: Simulation of a track near the crossing of an interface (Salvat, 2015).

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 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 hinge occurs in region "1", the angular deflection at the hinge is sampled from the distribution $F^{(\mathrm{s})}(s;\chi)$. Otherwise, the electron reaches the interface without changing its direction of movement. Assuming

$s \ll \lambda_{\mathrm{el},1}^{(\mathrm{s})}$, the mean angular deflection due to soft collisions is

$$1 - \langle \cos\chi \rangle^{(\mathrm{s})} = 1 - \exp(-s/\lambda_{\mathrm{el},1}^{(\mathrm{s})}) \simeq \frac{s}{\lambda_{\mathrm{el},1}^{(\mathrm{s})}}. \tag{4.22}$$

Additionally, 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

$$1 - \langle \cos\chi \rangle = \frac{t}{s}\left(1 - \langle \cos\chi \rangle^{(\mathrm{s})}\right) \simeq \frac{t}{\lambda_{\mathrm{el},1}^{(\mathrm{s})}}, \tag{4.23}$$

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 hinge 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 step 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 $F^{(\mathrm{s})}(s;\chi)$. 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 $F^{(\mathrm{s})}(s;\chi)$ may be calculated by using the small-angle approximation (see 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

$$P_\ell(\cos\theta) \simeq 1 - \frac{1}{4}\ell(\ell+1)\theta^2 \tag{4.24}$$

into Eq. (4.16a) we get

$$\frac{1}{\lambda_{\mathrm{el},\ell}^{(\mathrm{s})}} = \mathcal{N} 2\pi \frac{\ell(\ell+1)}{4} \int_0^{\theta_{\mathrm{c}}} \theta^2 \frac{\mathrm{d}\sigma_{\mathrm{el}}(\theta)}{\mathrm{d}\Omega} \sin\theta \,\mathrm{d}\theta = \frac{\ell(\ell+1)}{2} \frac{1}{\lambda_{\mathrm{el},1}^{(\mathrm{s})}}, \tag{4.25}$$

i.e., the transport mean free paths $\lambda_{\mathrm{el},\ell}^{(\mathrm{s})}$ are completely determined by the single value $\lambda_{\mathrm{el},1}^{(\mathrm{s})}$. The angular distribution $F^{(\mathrm{s})}$ then simplifies to

$$F^{(\mathrm{s})}(s;\chi) = \sum_{\ell=0}^{\infty} \frac{2\ell+1}{4\pi} \exp\left[-\frac{\ell(\ell+1)}{2}\frac{s}{\lambda_{\mathrm{el},1}^{(\mathrm{s})}}\right] P_\ell(\cos\chi). \tag{4.26}$$

This expression can be evaluated by using the Molière (1948) approximation for the Legendre polynomials and we obtain (see Fernández-Varea *et al.*, 1993a)

$$F^{(\mathrm{s})}(s;\chi) = \frac{1}{2\pi}\left(\frac{\chi}{\sin\chi}\right)^{1/2}\frac{\lambda^{(\mathrm{s})}_{\mathrm{el},1}}{s}\exp\left[\frac{s}{8\lambda^{(\mathrm{s})}_{\mathrm{el},1}} - \frac{\lambda^{(\mathrm{s})}_{\mathrm{el},1}}{2s}\chi^2\right], \tag{4.27}$$

which does not differ significantly from the Gaussian distribution with variance $s/\lambda^{(\mathrm{s})}_{\mathrm{el},1}$. This result is accurate whenever $s \ll \lambda^{(\mathrm{s})}_{\mathrm{el},1}$ and $\theta_\mathrm{c} \ll 1$. 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 $\lambda^{(\mathrm{s})}_{\mathrm{el},1}$ 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 $F^{(\mathrm{s})}(s;\chi)$ to sample the angular deflection in artificial soft collisions. Instead, we may use a simpler distribution, $F_\mathrm{a}(s;\chi)$, with the same mean and variance, without appreciably distorting the simulation results. This is so because 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 $F^{(\mathrm{s})}(s;\chi)$ 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 $\lambda^{(\mathrm{h})}_{\mathrm{el}}$) 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 $F^{(\mathrm{s})}(s;\mu)$ are

$$\langle\mu\rangle^{(\mathrm{s})} \equiv \int_0^1 \mu F_\mathrm{a}(s;\mu)\,\mathrm{d}\mu = \frac{1}{2}\left[1-\exp(-s/\lambda^{(\mathrm{s})}_{\mathrm{el},1})\right] \tag{4.28}$$

and

$$\langle\mu^2\rangle^{(\mathrm{s})} \equiv \int_0^1 \mu^2 F_\mathrm{a}(s;\mu)\,\mathrm{d}\mu = \langle\mu\rangle^{(\mathrm{s})} - \frac{1}{6}\left[1-\exp(-s/\lambda^{(\mathrm{s})}_{\mathrm{el},2})\right]. \tag{4.29}$$

The angular deflection in soft scattering events will be generated from a distribution $F_\mathrm{a}(s;\mu)$, which is required to satisfy Eqs. (4.28) and (4.29), but is otherwise arbitrary.

PENELOPE uses the following,

$$F_{\rm a}(s;\mu) = aU_{0,\mu_0}(\mu) + (1-a)U_{\mu_0,1}(\mu), \tag{4.30}$$

where $U_{u,v}(x)$ denotes the normalised uniform distribution in the interval (u,v),

$$U_{u,v}(x) = \begin{cases} 1/(v-u) & \text{if } u \le x \le v, \\ 0 & \text{otherwise.} \end{cases} \tag{4.31}$$

The parameters a and μ_0, obtained from conditions (4.28) and (4.29), are

$$\mu_0 = \frac{2\langle\mu\rangle^{(\rm s)} - 3\langle\mu^2\rangle^{(\rm s)}}{1-2\langle\mu\rangle^{(\rm s)}}, \qquad a = 1 - 2\langle\mu\rangle^{(\rm s)} + \mu_0. \tag{4.32}$$

This simple distribution is flexible enough to reproduce the combinations of first and second moments encountered in the simulations [notice that $\langle\mu\rangle^{(\rm s)}$, 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)]

$$\mu = \mathcal{P}_{\rm el}^{-1}(\xi), \tag{4.33}$$

where

$$\mathcal{P}_{\rm el}(\mu) = \int_0^\mu p_{\rm el}(E;\mu)\,{\rm d}\mu \tag{4.34}$$

is the cumulative distribution function of μ. To get sampled μ-values greater than the cutoff deflection $\mu_{\rm c} = (1-\cos\theta_{\rm c})/2$, we only need to use random numbers ξ restricted to the interval $(\xi_{\rm c},1)$ [see Eq. (3.27)].

The angular distribution of soft events $F_{\rm a}(s;\mu)$, 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 $\lambda_{\rm el}^{(\rm h)}$ between hard elastic events and the cutoff deflection $\mu_{\rm c}$ are related through [see Eqs. (3.26) and (4.9)]

$$\frac{1}{\lambda_{\rm el}^{(\rm h)}} = \frac{1}{\lambda_{\rm el}} \int_{\mu_{\rm c}}^1 p_{\rm el}(\mu)\,{\rm d}\mu. \tag{4.35}$$

This equation can be inverted to give

$$\mu_{\rm c} = \mathcal{P}_{\rm el}^{-1}\left(\xi_{\rm c}\right), \tag{4.36}$$

where

$$\xi_{\rm c} \equiv 1 - \frac{\lambda_{\rm el}}{\lambda_{\rm el}^{(\rm h)}}\,. \tag{4.37}$$

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

$$\left(\lambda_{\mathrm{el},1}^{(\mathrm{s})}\right)^{-1} = \frac{2}{\lambda_{\mathrm{el}}} T_1(\mu_c) \qquad \text{and} \qquad \left(\lambda_{\mathrm{el},2}^{(\mathrm{s})}\right)^{-1} = \frac{6}{\lambda_{\mathrm{el}}} \left[T_1(\mu_c) - T_2(\mu_c)\right] \tag{4.38}$$

with

$$T_1(\mu_c) = \int_0^{\mu_c} \mu p_{\mathrm{el}}(\mu)\, \mathrm{d}\mu \qquad \text{and} \qquad T_2(\mu_c) = \int_0^{\mu_c} \mu^2 p_{\mathrm{el}}(\mu)\, \mathrm{d}\mu. \tag{4.39}$$

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

$$\begin{aligned} T_1(\mu_c) &= \int_0^{\mu_c} \mu p_{\mathrm{MW-I}}(\mu)\, \mathrm{d}\mu \\ &= \begin{cases} (1-B) I_1(\mu_c) & \text{if } \ 0 \le \xi_c < \xi_0 \\ (1-B) I_1(\mu_0) + (\xi_c - \xi_0)\, \mu_0 & \text{if } \ \xi_0 \le \xi_c < \xi_0 + B \\ (1-B) I_1(\mu_c) + B\mu_0 & \text{if } \ \xi_0 + B \le \xi_c \le 1 \end{cases} \end{aligned} \tag{4.40}$$

and

$$\begin{aligned} T_2(\mu_c) &= \int_0^{\mu_c} \mu^2 p_{\mathrm{MW-I}}(\mu)\, \mathrm{d}\mu \\ &= \begin{cases} (1-B) I_2(\mu_c) & \text{if } \ 0 \le \xi_c < \xi_0 \\ (1-B) I_2(\mu_0) + (\xi_c - \xi_0)\, \mu_0^2 & \text{if } \ \xi_0 \le \xi_c < \xi_0 + B \\ (1-B) I_2(\mu_c) + B\mu_0^2 & \text{if } \ \xi_0 + B \le \xi_c \le 1 \end{cases} \end{aligned} \tag{4.41}$$

with

$$I_1(\mu) \equiv A \left[(1+A) \ln\left(\frac{A+\mu}{A}\right) - \frac{(1+A)\mu}{A+\mu} \right] \tag{4.42}$$

and

$$I_2(\mu) \equiv A \left[\frac{(1+A)\mu^2}{A+\mu} - 2 I_1(\mu) \right]. \tag{4.43}$$

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 completely for energy straggling, within the accuracy of multiple-scattering theory, but disregards the correlation between delta-ray emission and energy loss at each step. As a consequence, energetic delta rays can be generated in a step 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 $W_{\rm cc}$ and $W_{\rm cr}$, and consider inelastic collisions with energy loss $W < W_{\rm cc}$ and emission of bremsstrahlung photons with $W < W_{\rm cr}$ 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 $W_{\rm cc}$ and $W_{\rm cr}$ 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 along these lines, we will be able to use larger values of the cutoff energies $W_{\rm cc}$ and $W_{\rm cr}$, 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 $\lambda_{\rm in}^{\rm (h)}$ and $\lambda_{\rm br}^{\rm (h)}$ between hard collisions and hard radiative events, the stopping power $S_{\rm s}$ and the energy straggling parameter $\Omega_{\rm s}^2$ associated with soft interactions. These quantities are given by

$$\lambda_{\rm in}^{\rm (h)}(E) = \left(\mathcal{N} \int_{W_{\rm cc}}^{E} \frac{{\rm d}\sigma_{\rm in}}{{\rm d}W} \, {\rm d}W \right)^{-1} , \tag{4.44}$$

$$\lambda_{\rm br}^{\rm (h)}(E) = \left(\mathcal{N} \int_{W_{\rm cr}}^{E} \frac{{\rm d}\sigma_{\rm br}}{{\rm d}W} \, {\rm d}W \right)^{-1} , \tag{4.45}$$

$$S_{\rm s}(E) = \mathcal{N}\int_0^{W_{\rm cc}} W\frac{{\rm d}\sigma_{\rm in}}{{\rm d}W}\,{\rm d}W + \mathcal{N}\int_0^{W_{\rm cr}} W\frac{{\rm d}\sigma_{\rm br}}{{\rm d}W}\,{\rm d}W \tag{4.46}$$

and

$$\Omega^2_{\rm s}(E) = \mathcal{N}\int_0^{W_{\rm cc}} W^2\frac{{\rm d}\sigma_{\rm in}}{{\rm d}W}\,{\rm d}W + \mathcal{N}\int_0^{W_{\rm cr}} W^2\frac{{\rm d}\sigma_{\rm br}}{{\rm d}W}\,{\rm d}W. \tag{4.47}$$

To prevent $\lambda^{\rm (h)}_{\rm br}(E)$ from vanishing (infrared divergence), in PENELOPE the radiative cutoff energy $W_{\rm cr}$ 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, $S_{\rm s}(E)s$, is much less than the initial energy E so that the DCSs can be assumed to remain 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)

$$\frac{\partial G(s;\omega)}{\partial s} = \mathcal{N}\int_0^{\infty}\left[G(s;\omega-W) - G(s;\omega)\right]\sigma_{\rm s}(E;W)\,{\rm d}W \tag{4.48}$$

with the initial value $G(0;\omega) = \delta(\omega)$. Here, $\sigma_{\rm s}(E;W)$ stands for the DCS for soft stopping interactions, *i.e.*,

$$\sigma_{\rm s}(E;W) \equiv \frac{{\rm d}\sigma_{\rm s}}{{\rm d}W} = \frac{{\rm d}\sigma_{\rm in}}{{\rm d}W}\,\Theta(W_{\rm cc}-W) + \frac{{\rm d}\sigma_{\rm br}}{{\rm d}W}\,\Theta(W_{\rm cr}-W), \tag{4.49}$$

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 (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,

$$\langle\omega^n\rangle \equiv \int_0^{\infty}\omega^n G(s;\omega)\,{\rm d}\omega. \tag{4.50}$$

From Eq. (4.48) it follows that

$$\begin{aligned}\frac{\rm d}{{\rm d}s}\langle\omega^n\rangle &= \mathcal{N}\int_0^{\infty}{\rm d}\omega\int_0^{\infty}{\rm d}W\,\omega^n\left[G(s;\omega-W)-G(s;\omega)\right]\sigma_{\rm s}(E;W)\\ &= \mathcal{N}\left(\int_0^{\infty}{\rm d}\omega'\int_0^{\infty}{\rm d}W\,(\omega'+W)^n G(s;\omega')\sigma_{\rm s}(E;W) - \langle\omega^n\rangle\int_0^{\infty}\sigma_{\rm s}(E;W)\,{\rm d}W\right)\\ &= \sum_{k=1}^{n}\frac{n!}{k!(n-k)!}\langle\omega^{n-k}\rangle\mathcal{N}\int_0^{\infty}W^k\sigma_{\rm s}(E;W)\,{\rm d}W,\end{aligned} \tag{4.51}$$

where use has been made of the fact that $\sigma_{\rm s}(E;W)$ vanishes when $W<0$. In particular, we have

$$\frac{\rm d}{{\rm d}s}\langle\omega\rangle = \mathcal{N}\int_0^{\infty}W\sigma_{\rm s}(E;W)\,{\rm d}W = S_{\rm s}, \tag{4.52}$$

$$\frac{\mathrm{d}}{\mathrm{d}s}\langle\omega^2\rangle = 2\langle\omega\rangle\mathcal{N}\int_0^\infty W\sigma_\mathrm{s}(E;W)\,\mathrm{d}W + \mathcal{N}\int_0^\infty W^2\sigma_\mathrm{s}(E;W)\,\mathrm{d}W$$
$$= 2\langle\omega\rangle S_\mathrm{s} + \Omega_\mathrm{s}^2 \tag{4.53}$$

and, hence,

$$\langle\omega\rangle = S_\mathrm{s}s, \tag{4.54}$$

$$\langle\omega^2\rangle = (S_\mathrm{s}s)^2 + \Omega_\mathrm{s}^2 s. \tag{4.55}$$

The variance of the energy loss distribution is

$$\mathrm{var}(\omega) = \langle\omega^2\rangle - \langle\omega\rangle^2 = \Omega_\mathrm{s}^2 s, \tag{4.56}$$

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 $S_\mathrm{s}s$ and variance $\Omega_\mathrm{s}^2 s$,

$$G(s;\omega) \simeq \frac{1}{\left(2\pi\Omega_\mathrm{s}^2(E)s\right)^{1/2}} \exp\left[-\frac{(\omega - S_\mathrm{s}(E)s)^2}{2\Omega_\mathrm{s}^2(E)s}\right]. \tag{4.57}$$

This result is accurate only if 1) the average energy loss $S_\mathrm{s}(E)s$ is much smaller than E (so that the DCS $\mathrm{d}\sigma_\mathrm{s}/\mathrm{d}W$ is nearly constant along the step) and 2) its standard deviation $[\Omega_\mathrm{s}^2(E)s]^{1/2}$ is much smaller than its mean $S_\mathrm{s}(E)s$ (otherwise there would be a finite probability of negative energy losses), *i.e.*,

$$\left[\Omega_\mathrm{s}^2(E)s\right]^{1/2} \ll S_\mathrm{s}(E)s \ll E. \tag{4.58}$$

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 $s_\mathrm{crit} = \Omega_\mathrm{s}^2/S_\mathrm{s}^2$.

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 $G_\mathrm{a}(s;\omega)$ 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 $\langle\omega\rangle^2 > 9\,\mathrm{var}(\omega)$, we use a truncated Gaussian distribution (normalisation is irrelevant here),

$$G_{\mathrm{a,I}}(s;\omega) = \begin{cases} \exp\left[-\dfrac{(\omega-\langle\omega\rangle)^2}{2(1.015387\sigma)^2}\right] & \text{if } |\omega-\langle\omega\rangle| < 3\,\sigma, \\ 0 & \text{otherwise,} \end{cases} \tag{4.59}$$

where $\sigma = [\mathrm{var}(\omega)]^{1/2}$ 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 $3\,\mathrm{var}(\omega) < \langle\omega\rangle^2 < 9\,\mathrm{var}(\omega)$, the energy loss is sampled from the uniform distribution

$$G_{\mathrm{a,II}}(s;\omega) = U_{\omega_1,\omega_2}(\omega) \tag{4.60}$$

with

$$\omega_1 = \langle\omega\rangle - \sqrt{3}\,\sigma, \qquad \omega_2 = \langle\omega\rangle + \sqrt{3}\,\sigma. \tag{4.61}$$

• Case III. Finally, when $\langle\omega\rangle^2 < 3\,\mathrm{var}(\omega)$, the adopted distribution is an admixture of a delta and a uniform distribution,

$$G_{\mathrm{a,III}}(s;\omega) = a\delta(\omega) + (1-a)U_{0,\omega_0}(\omega) \tag{4.62}$$

with

$$a = \frac{3\mathrm{var}(\omega) - \langle\omega\rangle^2}{3\mathrm{var}(\omega) + 3\langle\omega\rangle^2} \qquad \text{and} \qquad \omega_0 = \frac{3\mathrm{var}(\omega) + 3\langle\omega\rangle^2}{2\langle\omega\rangle}. \tag{4.63}$$

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

$$\omega_{\max} = \begin{cases} \langle\omega\rangle + 3\sigma & \text{in case I,} \\ \omega_2 & \text{in case II,} \\ \omega_0 & \text{in case III.} \end{cases} \tag{4.64}$$

$\omega_{\max}$ 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 $F_{\rm a}(s;\mu)$, Eqs. (4.30)–(4.32), and loses energy ω that is generated from the distribution $G_{\rm a}(s;\omega)$, 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 hinge is simulated only when its position lies within 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, $E - S_{\rm s}t$.

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)]

$$\begin{aligned}\frac{\partial G(s;\omega)}{\partial s} &= \mathcal{N}\int_0^\infty G(s;\omega-W)\,\sigma_{\rm s}(E_0-\omega+W;W)\,{\rm d}W\\ &\quad-\mathcal{N}\int_0^\infty G(s;\omega)\,\sigma_{\rm s}(E_0-\omega;W)\,{\rm d}W,\end{aligned}\tag{4.65}$$

where E_0 denotes the kinetic energy of the particle at the beginning of the step. We wish 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 $G_{\rm a}(s;\omega)$ 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,

$$\sigma_{\rm s}(E_0-\omega;W)\simeq\sigma_{\rm s}(E_0;W)-\left[\frac{\partial\sigma_{\rm s}(E;W)}{\partial E}\right]_{E=E_0}\omega.\tag{4.66}$$

We recall that we are considering only soft energy-loss interactions (inelastic collisions and bremsstrahlung emission) for which the cutoff energies, $W_{\rm cc}$ and $W_{\rm cr}$, 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 $S_{\rm s}(E_0-\omega)$ can then be approximated as

$$S_{\rm s}(E_0-\omega)\equiv\mathcal{N}\int W\sigma_{\rm s}(E_0-\omega;W)\,{\rm d}W\simeq S_{\rm s}(E_0)-S'_{\rm s}(E_0)\,\omega,\tag{4.67}$$

where the prime denotes the derivative with respect to E. Similarly, for the straggling parameter $\Omega_{\rm s}^2(E)$ we have

$$\Omega_{\rm s}^2(E_0-\omega) \equiv \mathcal{N}\int W^2\sigma_{\rm s}(E_0-\omega;W)\,{\rm d}W \simeq \Omega_{\rm s}^2(E_0) - \Omega_{\rm s}^{2\prime}(E_0)\,\omega. \tag{4.68}$$

From Eq. (4.65) it follows that the moments of the multiple-scattering distribution,

$$\langle\omega^n\rangle = \int \omega^n G(s;\omega)\,{\rm d}\omega,$$

satisfy the equations

$$\begin{aligned}\frac{\rm d}{{\rm d}s}\langle\omega^n\rangle \;&=\; \mathcal{N}\int {\rm d}\omega\int {\rm d}W\;\;[(\omega+W)^n G(s;\omega)\sigma_{\rm s}(E_0-\omega;W)]\\ &-\;\mathcal{N}\int {\rm d}\omega\int {\rm d}W\;\omega^n G(s;\omega)\sigma_{\rm s}(E_0-\omega;W)\\ &=\;\mathcal{N}\sum_{k=1}^{n}\frac{n!}{k!(n-k)!}\int {\rm d}\omega\int {\rm d}W\,\omega^{n-k}W^k G(s;\omega)\sigma_{\rm s}(E_0-\omega;W).\end{aligned} \tag{4.69}$$

By inserting the approximation (4.66), we obtain

$$\frac{\rm d}{{\rm d}s}\langle\omega^n\rangle = \sum_{k=1}^{n}\frac{n!}{k!(n-k)!}\left(\left\langle\omega^{n-k}\right\rangle M_k - \left\langle\omega^{n-k+1}\right\rangle M_k'\right), \tag{4.70}$$

where

$$M_k \equiv \mathcal{N}\int W^k\sigma_{\rm s}(E_0;W)\,{\rm d}W \tag{4.71}$$

and

$$M_k' \equiv \mathcal{N}\int W^k\left[\frac{\partial\sigma_{\rm s}(E;W)}{\partial E}\right]_{E=E_0}{\rm d}W = \left[\frac{{\rm d}M_k}{{\rm d}E}\right]_{E=E_0}. \tag{4.72}$$

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

$$\frac{\rm d}{{\rm d}s}\langle\omega\rangle = S_{\rm s}(E_0) - S_{\rm s}'(E_0)\langle\omega\rangle, \tag{4.73}$$

which yields

$$\langle\omega\rangle = \frac{S_{\rm s}(E_0)}{S_{\rm s}'(E_0)}\left\{1-\exp\left[-S_{\rm s}'(E_0)s\right]\right\}. \tag{4.74}$$

The equation for $n=2$ reads,

$$\frac{\rm d}{{\rm d}s}\langle\omega^2\rangle \;=\; \Omega_{\rm s}^2(E_0) + \left[2S_{\rm s}(E_0) - \Omega_{\rm s}^{2\prime}(E_0)\right]\langle\omega\rangle - 2S_{\rm s}'(E_0)\langle\omega^2\rangle, \tag{4.75}$$

and its solution is

$$\begin{aligned}\langle\omega^2\rangle \;&=\; \Omega_{\rm s}^2(E_0)\,\frac{1-\exp[-2S'_{\rm s}(E_0)s]}{2S'_{\rm s}(E_0)}\\ &+\; s\left[2S_{\rm s}(E_0)-\Omega_{\rm s}^{2\prime}(E_0)\right]S_{\rm s}(E_0)\left[\frac{1-\exp[-S'_{\rm s}(E_0)s]}{2S'_{\rm s}(E_0)}\right]^2 .\end{aligned} \tag{4.76}$$

Hence,

$$\begin{aligned}{\rm var}(\omega) &= \langle\omega^2\rangle-\langle\omega\rangle^2\\ &= \Omega_{\rm s}^2(E_0)\,\frac{1-\exp[-2S'_{\rm s}(E_0)s]}{2S'_{\rm s}(E_0)}-2\Omega_{\rm s}^{2\prime}(E_0)S_{\rm s}(E_0)\left[\frac{1-\exp[-S'_{\rm s}(E_0)s]}{2S'_{\rm s}(E_0)}\right]^2 .\end{aligned} \tag{4.77}$$

Since these expressions are derived from the linear approximation, Eq. (4.66), it is consistent to evaluate $\langle\omega\rangle$ and ${\rm var}(\omega)$ from their Taylor expansions to second order,

$$\begin{aligned}\langle\omega\rangle \;&=\; S_{\rm s}(E_0)\,s\left[1-\frac{1}{2}S'_{\rm s}(E_0)\,s+\mathcal{O}(s^2)\right]\\ &\simeq\; S_{\rm s}(E_0)\,s\left\{1-\frac{1}{2}\left[\frac{{\rm d}\ln S_{\rm s}(E)}{{\rm d}E}\right]_{E=E_0}S_{\rm s}(E_0)\,s\right\}\end{aligned} \tag{4.78}$$

and

$$\begin{aligned}{\rm var}(\omega) \;&=\; \Omega_{\rm s}^2(E_0)\,s-\left[\frac{1}{2}\Omega_{\rm s}^{2\prime}(E_0)\,S_{\rm s}(E_0)+\Omega_{\rm s}^2(E_0)\,S'_{\rm s}(E_0)\right]s^2+\mathcal{O}(s^3)\\ &\simeq\; \Omega_{\rm s}^2(E_0)\,s\left\{1-\left[\frac{1}{2}\frac{{\rm d}\ln\Omega_{\rm s}^2(E)}{{\rm d}E}+\frac{{\rm d}\ln S_{\rm s}(E)}{{\rm d}E}\right]_{E=E_0}S_{\rm s}(E_0)\,s\right\},\end{aligned} \tag{4.79}$$

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 $G_{\rm a}(\omega;s)$, 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, $\lambda_{\rm T}^{\rm (h)}$, is given by

$$\left[\lambda_{\rm T}^{\rm (h)}\right]^{-1} = \mathcal{N}\sigma_{\rm T}^{\rm (h)} = \mathcal{N}\left[\sigma_{\rm el}^{\rm (h)} + \sigma_{\rm in}^{\rm (h)} + \sigma_{\rm br}^{\rm (h)} + \sigma_{\rm si}\left(+\sigma_{\rm an}\right)\right] \equiv \Sigma_{\rm h}, \tag{4.80}$$

where $\sigma_{\rm T}^{\rm (h)}$ is the total atomic cross section for hard interactions. We recall that the inverse mean free path, $\Sigma_{\rm h}$, 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

$$p(s) = \Sigma_{\rm h} \exp\left(-\Sigma_{\rm h} s\right). \tag{4.81}$$

In each hard event, one and only one interaction (i="el", "in", "br", "si" or "an") occurs with probability

$$p_{\rm i} = \sigma_{\rm i}^{\rm (h)}/\sigma_{\rm T}^{\rm (h)}. \tag{4.82}$$

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 $\lambda_{\rm T}^{\rm (h)}$ 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 $\lambda_{\rm T}^{\rm (h)}(E)$ remains essentially constant along the step. Then, the mean energy loss in a step is given by

$$\langle \Delta E \rangle = \lambda_{\rm T}^{\rm (h)} S(E), \tag{4.83}$$

where

$$S(E) = S_{\rm in}(E) + S_{\rm br}(E) \tag{4.84}$$

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

$$\lambda_{\rm el}^{\rm (h)}(E) \equiv \max\left\{\lambda_{\rm el}(E), \min\left[C_1\lambda_{\rm el,1}(E), C_2\frac{E}{S(E)}\right]\right\}. \tag{4.85}$$

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), which are 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).

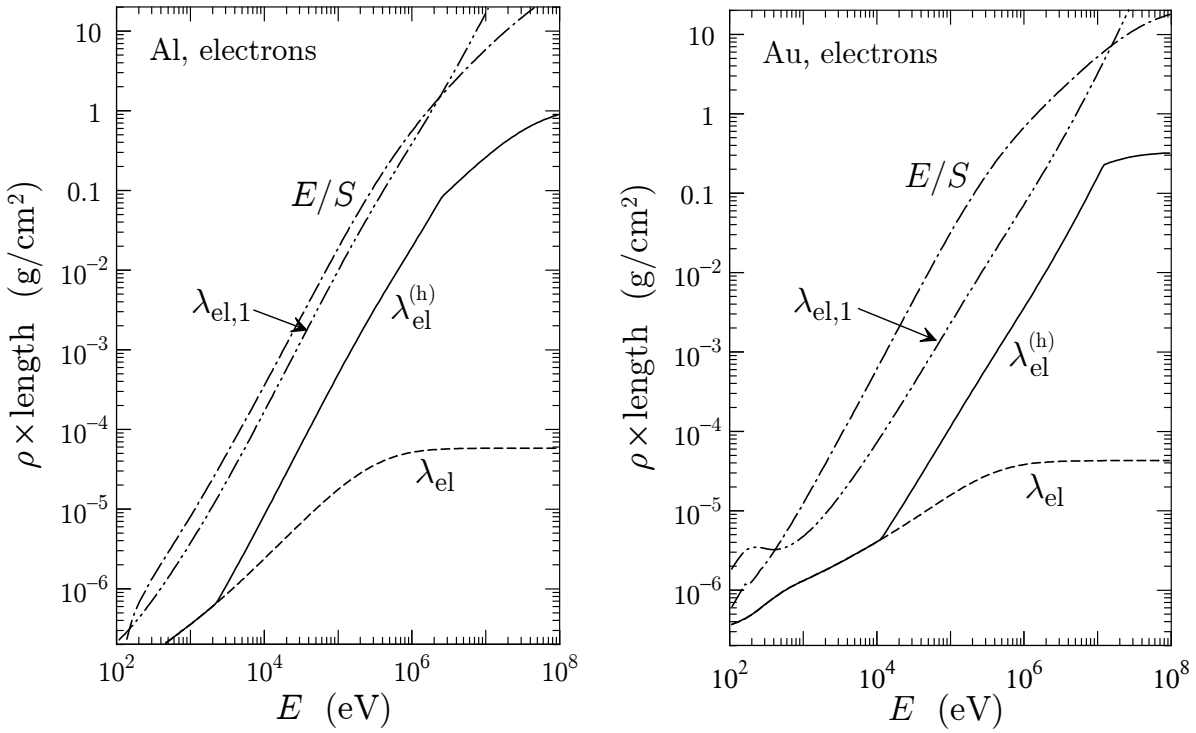

Figure 4.3: Elastic mean free path $\lambda_{\rm el}$, first transport mean free path $\lambda_{\rm el,1}$ and $E/S(E)$ for electrons in aluminium and gold. The solid line represents the mean free path between hard elastic events $\lambda_{\rm el}^{\rm (h)}$ obtained from Eq. (4.85) with $C_1 = C_2 = 0.05$. (Adapted from Baró *et al.*, 1995).

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 $\lambda_{\rm el}$, $\lambda_{\rm el,1}$ and E/S for electrons in aluminium and gold are represented as functions of the kinetic energy. The mean free path $\lambda_{\rm el}^{\rm (h)}$ for hard elastic events, determined from the prescription (4.85) with $C_1 = C_2 = 0.05$ is also plotted. For low energies, $\lambda_{\rm el}^{\rm (h)} = \lambda_{\rm el}$ and the simulation is purely detailed ($\mu_{\rm c} = 0$). For intermediate energies, $\lambda_{\rm el}^{\rm (h)} = C_1\lambda_{\rm el,1}$, whereas $\lambda_{\rm el}^{\rm (h)} = C_2 E/S(E)$ 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 $\lambda_{\rm T}^{\rm (h)}$ 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 $\Sigma_{\rm h}(E)$ 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-to-one correspondence between the kinetic energy E of the electron and the travelled path length s,

$$s = \int_E^{E_0} \frac{\mathrm{d}E'}{S_\mathrm{s}(E')}, \tag{4.86}$$

where E_0 is the initial energy (at $s = 0$) and $S_\mathrm{s}(E)$ is the soft stopping power, Eq. (4.46) [we consider that no hard interactions occur along the step]. Equivalently,

$$\frac{\mathrm{d}s}{\mathrm{d}E} = -\frac{1}{S_\mathrm{s}(E)}. \tag{4.87}$$

Thus, the inverse mean free path Σ_h can be formally considered as a function of the path length s. The probability $p(s)\,\mathrm{d}s$ of having the first hard interaction when the particle has travelled a length in the interval $(s, s + \mathrm{d}s)$ is determined by the equation [cf. Eq. (1.109)]

$$p(s) = \Sigma_\mathrm{h}(s) \int_s^\infty p(s')\,\mathrm{d}s', \tag{4.88}$$

with the normalisation condition,

$$\int_0^\infty p(s)\,\mathrm{d}s = 1. \tag{4.89}$$

Instead of the path length s, it is convenient to consider the dimensionless variable

$$q \equiv \int_E^{E_0} \frac{\Sigma_\mathrm{h}(E')}{S_\mathrm{s}(E')}\mathrm{d}E' = \int_0^s \Sigma_\mathrm{h}(s')\,\mathrm{d}s', \tag{4.90}$$

which varies with energy and

$$\frac{\mathrm{d}q}{\mathrm{d}E} = -\frac{\Sigma_\mathrm{h}(E)}{S_\mathrm{s}(E)}. \tag{4.91}$$

The PDF of q is

$$\pi(q) = p(s)\frac{\mathrm{d}s}{\mathrm{d}q} = p(s)\,\frac{\mathrm{d}s}{\mathrm{d}E}\,\frac{\mathrm{d}E}{\mathrm{d}q} = p(s)\frac{1}{\Sigma_\mathrm{h}(s)}. \tag{4.92}$$

From Eq. (4.88) it follows that $\pi(q)$ satisfies the equation

$$\pi(q) = \int_q^\infty \pi(q')\,\mathrm{d}q'. \tag{4.93}$$

Therefore, q is distributed exponentially,

$$\pi(q) = \exp(-q). \tag{4.94}$$

The PDF of the step length s is obtained by inverting the transformation (4.90),

$$p(s) = \Sigma_\mathrm{h}(s) \exp\left(-\int_0^s \Sigma_\mathrm{h}(s')\,\mathrm{d}s'\right). \tag{4.95}$$

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 $\Sigma_\mathrm{s}(E)$, 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,

$$\omega_\mathrm{max} = \int_0^{s_\mathrm{max}} S_\mathrm{s}(s)\,\mathrm{d}s. \tag{4.96}$$

Let $\Sigma_\mathrm{h,max}$ denote an upper bound for the inverse mean free path of hard events in the swept energy interval, *i.e.*,

$$\Sigma_\mathrm{h,max} > \max\left\{\Sigma_\mathrm{h}(E), E \in (E_0 - \omega_\mathrm{max}, E_0)\right\} \tag{4.97}$$

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

$$\Sigma_\delta(E) = \Sigma_\mathrm{h,max} - \Sigma_\mathrm{h}(E), \tag{4.98}$$

so that the inverse mean free path of the combined process (delta interactions + hard events) equals $\Sigma_\mathrm{h,max}$, 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 $\Sigma_\mathrm{h,max}$, *i.e.*, $s = (-\ln\xi)/\Sigma_\mathrm{h,max}$.

(ii) If $s > s_\mathrm{max}$, move the electron a path length s_max and determine the soft energy loss ω along this path length. Modify the electron energy[2], $E \leftarrow E - \omega$, and assume that a delta interaction occurs at the end of the step.

(iii) If $s < s_\mathrm{max}$, 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 $\xi\Sigma_\mathrm{h,max} < \Sigma_\mathrm{h}(E)$, simulate a hard interaction

  (2) Otherwise, assume that the particle undergoes a delta interaction.

(iv) Return to (i).

[2]In 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.

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 $s_{\max}$.

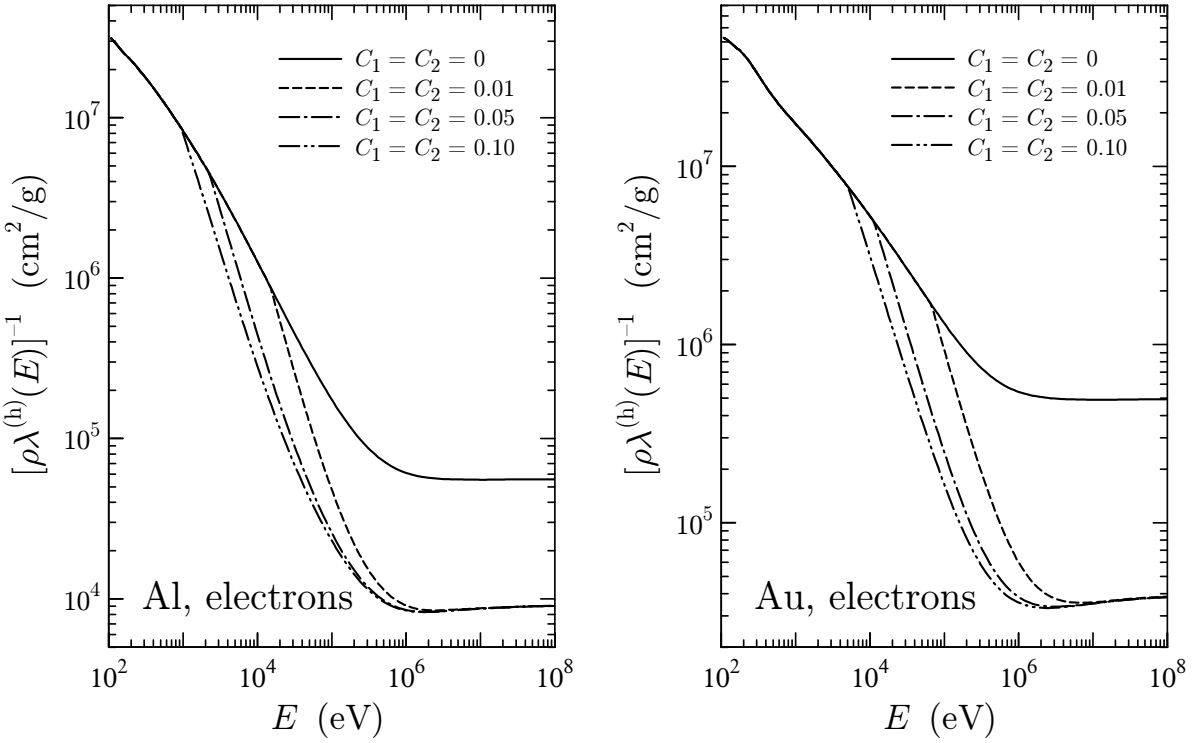

Figure 4.4: Inverse mean free path (interaction probability per unit path length) for hard interactions of electrons in aluminium and gold for the indicated values of the simulation parameters. The plotted curves were calculated with $W_{\mathrm{cc}} = W_{\mathrm{cr}} = 100$ eV. Note that, by varying 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. Hence, at high energies, the value $\lambda^{(\mathrm{h})}(E)$ is determined by the cutoff energies W_{cc} and W_{cr}.

Fortunately, the energy loss generated from the artificial distribution $G_{\mathrm{a}}(\omega; s)$, 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 $\Sigma_{\mathrm{h}}(E)$ 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 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 Mott and Massey, 1965). This was the approach followed by Fano (1954b) 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. In addition, 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 $W < W_{\rm cc}$, from our inelastic scattering model (see Section 3.2), as follows.

We recall that the recoil energy Q is given by (see Appendix A)

$$Q(Q + 2\mathrm{m_e}c^2) = c^2(p^2 + p'^2 - 2pp'\cos\theta), \tag{4.99}$$

where p and p' are the magnitudes of the momentum of the projectile before and after the collision,

$$(cp)^2 = E(E + 2\mathrm{m_e}c^2) \qquad \text{and} \qquad (cp')^2 = (E - W)(E - W + 2\mathrm{m_e}c^2). \tag{4.100}$$

In soft distant interactions with the k-th oscillator, the angular deflection $\mu = (1 - \cos\theta)/2$ and the recoil energy Q are related through

$$Q(Q + 2\mathrm{m_e}c^2) = 4cp\, cp_k\mu + (cp - cp_k)^2, \tag{4.101}$$

where p_k is the momentum of the projectile after the collision, calculated by assuming that the energy loss equals the resonance energy W_k of the oscillator,

$$(cp_k)^2 = (E - W_k)(E - W_k + 2\mathrm{m_e}c^2). \tag{4.102}$$

In the case of inner-shell excitations, W_k should be replaced by the modified resonance energy W'_k, Eq. (3.78) [see the comments about Eqs. (3.83) and (3.126) in Section 3.2]. The scattering DCS for soft distant interactions[3], Eq. (3.81), can then be expressed in terms of the variable μ as

$$\frac{\mathrm{d}\sigma_{\rm dis,l}}{\mathrm{d}\mu} = \frac{2\pi e^4}{\mathrm{m_e}v^2}\sum_k \frac{f_k}{W_k}\left(\int_0^{W_{\rm cc}} p_{\rm dis}(W)\,\mathrm{d}W\right) \times \frac{2\mathrm{m_e}c^2}{4\,cp\,cp_k\mu + (cp - cp_k)^2}\,\frac{4\,cp\,cp_k}{2(Q + \mathrm{m_e}c^2)}\,, \tag{4.103}$$

[3]Distant transverse interactions do not cause appreciable scattering.

with $p_{\rm in}(W)=\delta(W-W_k)$ for the conduction band. Considering that $Q \ll {\rm m_e}c^2$ for the majority of soft distant collisions, we have

$$\frac{{\rm d}\sigma_{\rm dis,l}}{{\rm d}\mu}=\frac{2\pi e^4}{{\rm m_e}v^2}\sum_k \frac{f_k}{W_k}\left(\int_0^{W_{\rm cc}} p_{\rm dis}(W)\,{\rm d}W\right)\frac{1}{R_k+\mu}, \qquad 0<\mu<\mu_{k1}, \tag{4.104}$$

where

$$R_k=\frac{(cp-cp_k)^2}{4\,cp\,cp_k} \tag{4.105}$$

and

$$\mu_{k1}=\mu(Q=Q_k)=\frac{Q_k(Q_k+2{\rm m_e}c^2)-(cp-cp_k)^2}{4cp\,cp_k}. \tag{4.106}$$

Again, in the case of inner shells the cutoff recoil energy Q_k should be replaced by the modified value Q'_k given by Eq. (3.80).

On the other hand, the DCS for soft ($W<W_{\rm cc}$) close collisions is given by [see Eqs. (3.86) and (3.92)]

$$\frac{{\rm d}\sigma^{(\pm)}_{\rm clo}}{{\rm d}W}=\frac{2\pi e^4}{{\rm m_e}v^2}\sum_k f_k\frac{1}{W^2}F^{(\pm)}(E,W). \tag{4.107}$$

The angular deflection and the energy loss are related by (3.134), which implies that

$$W=\frac{E(E+2{\rm m_e}c^2)2(\mu-\mu^2)}{2E(\mu-\mu^2)+{\rm m_e}c^2} \tag{4.108}$$

and

$$\frac{{\rm d}W}{{\rm d}\mu}=\frac{E(E+2{\rm m_e}c^2){\rm m_e}c^2 2(1-2\mu)}{[2E(\mu-\mu^2)+{\rm m_e}c^2]^2}. \tag{4.109}$$

Therefore,

$$\frac{{\rm d}\sigma^{(\pm)}_{\rm clo}}{{\rm d}\mu}=\frac{2\pi e^4}{{\rm m_e}v^2}\sum_k f_k\frac{1}{W^2}F^{(\pm)}(E,W)\frac{{\rm d}W}{{\rm d}\mu}, \qquad \mu_{k2}<\mu<\mu_{k3}, \tag{4.110}$$

where

$$\begin{aligned}\mu_{k2} &= \mu(Q=Q_k)=\frac{Q_k(Q_k+2{\rm m_e}c^2)-(cp-cp_{k2})^2}{4cp\,cp_{k2}},\\ \mu_{k3} &= \mu(Q=W_{\rm cc})=\frac{W_{\rm cc}(W_{\rm cc}+2{\rm m_e}c^2)-(cp-cp_{k3})^2}{4cp\,cp_{k3}},\end{aligned} \tag{4.111}$$

with

$$\begin{aligned}(cp_{k2})^2&=(E-Q_k)(E-Q_k+2{\rm m_e}c^2),\\ (cp_{k3})^2&=(E-W_{\rm cc})(E-W_{\rm cc}+2{\rm m_e}c^2).\end{aligned} \tag{4.112}$$

The angular DCS for soft inelastic interactions is then given by

$$\begin{aligned}\frac{\mathrm{d}\sigma^{(\mathrm{s})}_{\mathrm{in}}}{\mathrm{d}\mu} &= \frac{\mathrm{d}\sigma_{\mathrm{dis,l}}}{\mathrm{d}\mu} + \frac{\mathrm{d}\sigma^{(\pm)}_{\mathrm{clo}}}{\mathrm{d}\mu} \\ &= \frac{2\pi e^4}{\mathrm{m_e}v^2}\sum_k \frac{f_k}{W_k}\left\{\left(\int_0^{W_{\mathrm{cc}}} p_{\mathrm{dis}}(W)\,\mathrm{d}W\right)\frac{1}{R_k+\mu}\right. \\ &\quad \left. + \frac{1}{W^2}F^{(\pm)}(E,W)\frac{\mathrm{d}W}{\mathrm{d}\mu}\right\}, \end{aligned} \tag{4.113}$$

where the summation extends over the oscillators with resonance energy W_k less than W_{cc} (ionisation energy U_k less than W_{cc} in the case of inner shells), 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

$$\left[\lambda^{(\mathrm{s})}_{\mathrm{in}}\right]^{-1} = \mathcal{N}\int_0^{\mu^{(\mathrm{s})}_{\mathrm{max}}} \frac{\mathrm{d}\sigma^{(\mathrm{s})}_{\mathrm{in}}}{\mathrm{d}\mu}\,\mathrm{d}\mu, \tag{4.114}$$

$$\left[\lambda^{(\mathrm{s})}_{\mathrm{in,1}}\right]^{-1} = \mathcal{N}\int_0^{\mu^{(\mathrm{s})}_{\mathrm{max}}} 2\mu\frac{\mathrm{d}\sigma^{(\mathrm{s})}_{\mathrm{in}}}{\mathrm{d}\mu}\,\mathrm{d}\mu \tag{4.115}$$

and

$$\left[\lambda^{(\mathrm{s})}_{\mathrm{in,2}}\right]^{-1} = \mathcal{N}\int_0^{\mu^{(\mathrm{s})}_{\mathrm{max}}} 6(\mu-\mu^2)\frac{\mathrm{d}\sigma^{(\mathrm{s})}_{\mathrm{in}}}{\mathrm{d}\mu}\,\mathrm{d}\mu\,, \tag{4.116}$$

where $\mu^{(\mathrm{s})}_{\mathrm{max}}$ is the maximum angular deflection in soft inelastic interactions.

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 inverse transport mean free paths

$$\left[\lambda^{(\mathrm{s})}_{\mathrm{comb,1}}\right]^{-1} = \left[\lambda^{(\mathrm{s})}_{\mathrm{el,1}}\right]^{-1} + \left[\lambda^{(\mathrm{s})}_{\mathrm{in,1}}\right]^{-1} \tag{4.117}$$

and

$$\left[\lambda^{(\mathrm{s})}_{\mathrm{comb,2}}\right]^{-1} = \left[\lambda^{(\mathrm{s})}_{\mathrm{el,2}}\right]^{-1} + \left[\lambda^{(\mathrm{s})}_{\mathrm{in,2}}\right]^{-1}. \tag{4.118}$$

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 Energy deposition and flight times

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 $\lambda^{(\mathrm{s})}_{\mathrm{comb,1}}$ and $\lambda^{(\mathrm{s})}_{\mathrm{comb,2}}$. The

energy loss ω caused by soft-stopping interactions along the step is sampled from the distribution $G_{\rm a}(s;\omega)$, Eqs. (4.59)–(4.63).

In previous versions of PENELOPE, to keep the program simple (and independent of the geometry), the energy ω was assumed to be deposited at the hinge. Furthermore, to account (at least partially) for the energy dependence of $\lambda^{(\rm s)}_{\rm comb,1}$ and $\lambda^{(\rm s)}_{\rm comb,2}$ we used a trick, the alternate random hinge, which was essentially equivalent to assuming that these quantities vary linearly with energy. It has been observed recently that simulations of electron transport in multilayers of various materials become sensitive to user parameters when step lengths are comparable to the layer thickness because the random-hinge method overestimates energy straggling in steps that end at an interface. To mitigate this deficiency, in the current version of the code the effect of soft energy-loss events is regarded as a continuous slowing down process characterised by a (constant) stopping power $S_{\rm soft}$ that reproduces the sampled value ω of the energy loss. This approach amounts to considering that the energy ω is spread uniformly over the step, which is closer to reality than depositing this energy locally at the hinge. Additionally, it allows us to account for the fact that a step ending at an interface has a reduced length and a smaller energy loss.

PENELOPE describes a step of length s as a sequence of two trajectory segments of respective lengths τ and $s-\tau$ (see Fig. 4.1). The first of these segments terminates at the hinge, where the particle is deflected. Let E denote the energy of the particle at the beginning of the step. In the simulation routines, we first sample the soft energy loss ω along the step, and determine the effective stopping power $S_{\rm soft} = \omega/s$. We then sample the position τ of the hinge, move the particle this distance in the initial direction, and determine the current energy, $E - S_{\rm soft}\tau$. The soft angular deflection is sampled from the distribution (4.30) evaluated at this energy, and the new direction is determined. Finally, the particle flies the remaining distance $s-\tau$ in the new direction. For the sake of compatibility with previous versions of PENELOPE, the full energy loss ω is associated to the hinge, where the energy of the projectile is reduced in that amount; the evaluation of the energy loss along each segment of the trajectory can be performed by the main program. For this purpose, the simulation routines deliver the values of effective stopping power $S_{\rm soft}$ and of the energy E_0 of the particle at the beginning of the segment (*i.e.*, $E_0 = E$ for the first segment, and $E_0 = E - S_{\rm soft}\tau$ for the second segment). If in the process the particle reaches an interface after a truncated segment of length ℓ, the energy lost by the particle along that segment is $S_{\rm soft}\ell$ and the energy at the interface is $E_0 - S_{\rm soft}\ell$.

An advantage of considering that soft energy-loss interactions slow down electrons and positrons with constant stopping power is that the calculation of flight times is almost trivial. Consider an electron with initial energy E, subject to the stopping power $S_{\rm soft}$. The time in which the electron moves along a trajectory segment of length s' is given by

$$t = \int \frac{{\rm d}s}{v} = \int_{E-S_{\rm soft}s'}^{E} \frac{1}{v(E)}\,\frac{{\rm d}E}{S_{\rm soft}}\,.$$

Inserting the expression (A.7) for the velocity $v(E)$, and recalling that $S_{\rm soft}$ is a constant, the integral is elementary,

$$t = \frac{1}{cS_{\rm soft}} \int_{E-S_{\rm soft}s'}^{E} \frac{E+\mathrm{m_e}c^2}{\sqrt{E(E+2\mathrm{m_e}c^2)}}\,\mathrm{d}E$$
$$= \frac{1}{cS_{\rm soft}} \left[\sqrt{E(E+2\mathrm{m_e}c^2)} - \sqrt{(E-S_{\rm soft}s')(E-S_{\rm soft}s'+2\mathrm{m_e}c^2)}\right]. \quad (4.119)$$

This formula is utilised in the simulation program `penmain` (Section 7.2.2) for on-the-fly calculation of the age of electrons and positrons.

The CSDA with constant stopping power also facilitates the calculation of the average distribution of fluence with respect to energy, $\overline{\Phi}(E)\,\Delta E$ [see Eq. (5.39)]. This quantity is built by tallying the path length travelled by a particle while its energy is in the interval $(E, E+\Delta E)$. Assume that a particle with initial energy E_0 flies a distance s'. At the end of the flight its energy is $E_1 = E_0 - S_{\rm soft}s'$. The particle contributes to $\overline{\Phi}(E)\,\Delta E$ only when the energy intervals $(E, E+\Delta E)$ and (E_1, E_0) overlap, and the contribution is equal to $\delta E/S_{\rm soft}$, where δE is the length of the intersection of the two intervals.

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, 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, when mixed simulation is effective, consists of two segments and a hinge.

Simulation with PENELOPE is controlled by the constants C_1 and C_2 [see Eq. (4.85)] and the cutoff energies $W_{\rm cc}$ and $W_{\rm cr}$. Hereafter, these four quantities will be referred to as simulation parameters. The parameter C_1, which determines the mean free path $\lambda_{\rm el}^{(\rm h)}$ 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 $\lambda_{\rm el}^{(\rm h)}$. 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 $W_{\rm cc}$ and $W_{\rm cr}$ 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 quite insensitive to the adopted values of $W_{\rm cc}$ and $W_{\rm cr}$ 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 hinge, in which the particle changes its direction of movement, combined with a continuous slowing down process with constant stopping power $S_{\rm soft}$ giving a prescribed total energy loss at the end of the step. When $W_{\rm cc}$ 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, 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. (3.146)]. To test the accuracy of mixed algorithms, and to also perform studies of low-energy electron and positron transport (with, say, $E < 100$ keV), it may be convenient to run 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 $W_{\rm cr}$ value selected by the user is negative, in which case the program sets $W_{\rm cr} = 10$ eV, disregards soft bremsstrahlung events and simulates hard events (with $W > 10$ eV) in a detailed way. Generation of the angular deflection in the hinges is discontinued when the simulation of elastic and inelastic scattering becomes detailed (*i.e.*, when $\lambda_{\rm el}^{(\rm h)} = \lambda_{\rm el}$, $W_{\rm cc} = 0$).

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 the simulation to be resumed 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 out 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 $4\lambda_{\rm T}^{(\rm h)}$. When the user-selected value is larger, the code sets $s_{\max} = 4\lambda_{\rm T}^{(\rm h)}$; in this case, about 2% 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 $\lambda_{\rm T}^{(\rm h)}$ is correctly accounted for. Incidentally, limiting the step length is also necessary to perform simulation of electron/positron transport in external static electromagnetic fields (see Appendix D).

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 $E_{\rm abs}$; 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 upper bound $\Sigma_{\mathrm{h,max}}$ for the inverse mean free path of hard events, Eq. (4.97). 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

$$s = -\ln \xi / \Sigma_{\mathrm{h,max}}. \tag{4.120}$$

If $s > s_{\max}$, truncate the step by setting $s = s_{\max}$.

(iv) Sample the energy loss ω due to soft stopping interactions along the step s from the distribution $G_{\mathrm{a}}(s;\omega)$, Eqs. (4.59)–(4.63), and determine the soft stopping power $S_{\mathrm{soft}} = \omega/s$.

(v) Generate the length $\tau = s\xi$ of the segment to the hinge. Let the particle advance this distance in the direction $\hat{\mathbf{d}}$: $\mathbf{r} \leftarrow \mathbf{r} + \tau\hat{\mathbf{d}}$.

(vi) If the track has crossed an interface:
Stop the particle at the intersection of the trajectory with the (first) crossed interface (*i.e.*, redefine $\mathbf{r}$ as equal to the position of this point), set τ equal to the travelled distance, and reduce the kinetic energy: $E \leftarrow E - S_{\mathrm{soft}}\tau$.
Go to (ii) to continue the simulation in the new material, or go to (xiii) if the new material is the outer vacuum.

(vii) Reduce the kinetic energy: $E \leftarrow E - S_{\mathrm{soft}}\tau$.

(viii) Simulate the hinge: Sample the polar angular deflection $\mu = (1-\cos\theta)/2$ from the distribution $F_{\mathrm{a}}(s;\mu)$, 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 angles (as described in Section 1.4.4) to obtain the new direction: $\hat{\mathbf{d}} \leftarrow \mathcal{R}(\theta,\phi)\hat{\mathbf{d}}$.
Deposit the energy ω at $\mathbf{r}$.
Go to (xiii) if $E < E_{\mathrm{abs}}$.

(ix) Let the particle advance the distance $s - \tau$ in the direction $\hat{\mathbf{d}}$: $\mathbf{r} \leftarrow \mathbf{r} + (s-\tau)\hat{\mathbf{d}}$.

(x) Do as in (vi).

(xi) Reduce the kinetic energy: $E \leftarrow E - S_{\mathrm{soft}}(s-\tau)$. If in step (iii) the step length is truncated, *i.e.*, if $s = s_{\max}$, simulate a delta interaction and go to (ii).

(xii) Simulate the hard event:
Sample the kind of interaction according to the point probabilities,

$$p_{\rm el}=\frac{\mathcal{N}\sigma_{\rm el}^{\rm (h)}}{\Sigma_{\rm h,max}},\quad p_{\rm in}=\frac{\mathcal{N}\sigma_{\rm in}^{\rm (h)}}{\Sigma_{\rm h,max}},\quad p_{\rm br}=\frac{\mathcal{N}\sigma_{\rm br}^{\rm (h)}}{\Sigma_{\rm h,max}},\quad p_{\rm si}=\frac{\mathcal{N}\sigma_{\rm si}}{\Sigma_{\rm h,max}},$$

$$p_\delta=\frac{\Sigma_\delta}{\Sigma_{\rm h,max}},\quad \text{and}\quad p_{\rm an}=\frac{\mathcal{N}\sigma_{\rm an}}{\Sigma_{\rm h,max}}\quad \text{in the case of positrons.} \tag{4.121}$$

If the event is a delta interaction, return to (ii).
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: $\hat{\mathbf{d}}\leftarrow\mathcal{R}(\theta,\phi)\hat{\mathbf{d}}$.
Deposit the energy loss W at $\mathbf{r}$, and reduce the kinetic energy: $E\leftarrow E-W$.
If, as a result of the interaction, a secondary particle is emitted in a direction $\hat{\mathbf{d}}_{\rm s}$, with energy $E_{\rm s}>E_{\rm abs}$, store its initial state $(\mathbf{r},E_{\rm s},\hat{\mathbf{d}}_{\rm s})$.
Go to (ii) if $E>E_{\rm abs}$.

(xiii) If the particle is in a material medium, deposit its residual energy on the spot.

(xiv) Simulate the secondary particles 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 $E_{\rm abs}$, C_1, $W_{\rm cc}$ and $s_{\rm max}$, because C_2 is not effective (see Fig. 4.3) and radiative emission is unimportant (hard bremsstrahlung events occur very seldom and, therefore, $W_{\rm cr}$ has no influence). The value of the parameter $s_{\rm max}$ is important to ensure reliability of the results; a safe recipe is to set $s_{\rm max}$ equal to one tenth of the "expected track length" or less. Since the values of $E_{\rm abs}$ and $W_{\rm cc}$ 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.

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 performed with a main program that tracks particles in an infinite medium. Electrons started off from the origin of coordinates moving in the direction of the z axis. During generation of each electron track, we scored the energy deposited at different "depths" (z-coordinate) to get the "depth-dose" distribution. Simulation of a track was discontinued when the electron

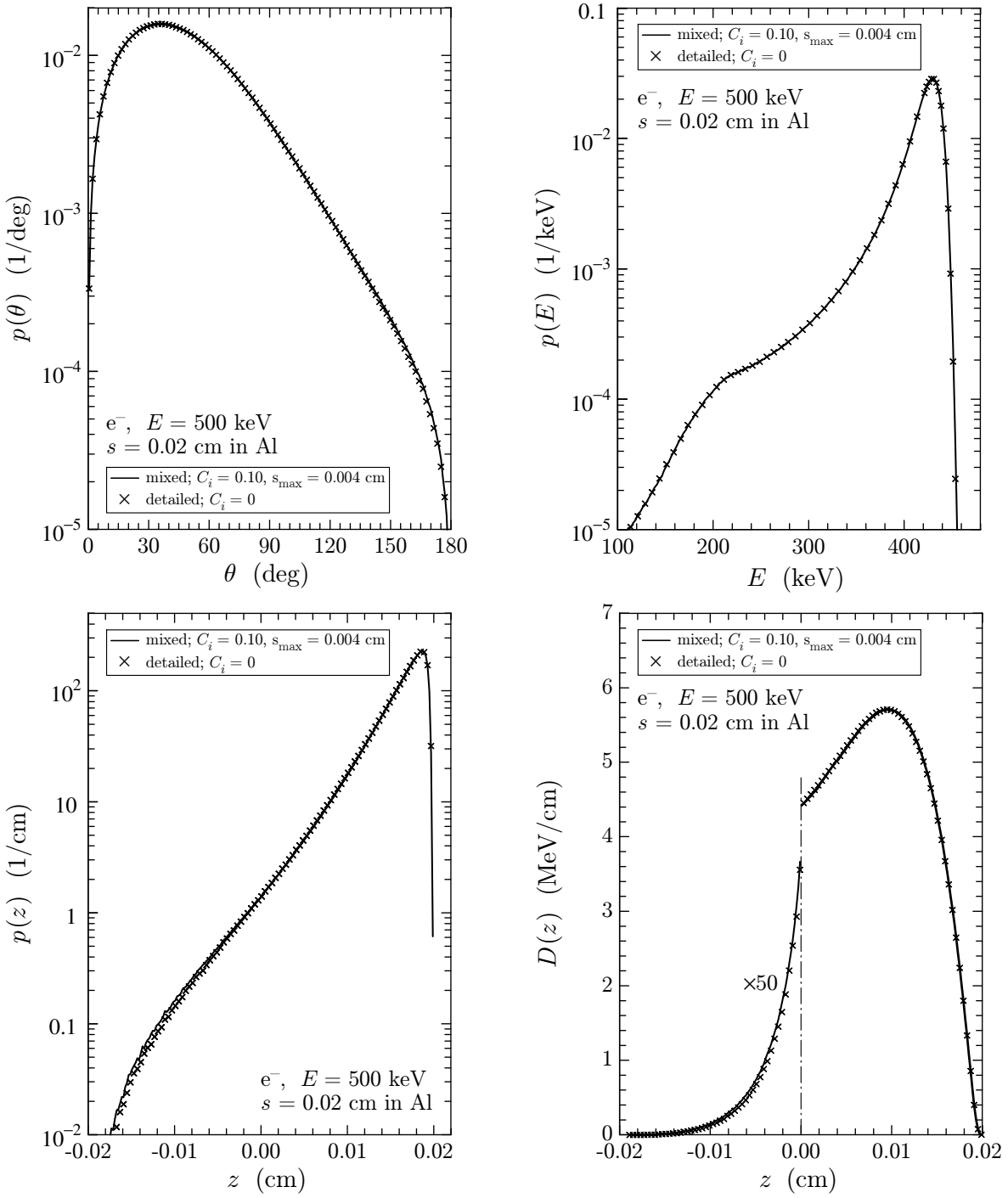

Figure 4.5: Results from the simulations of 500 keV electrons in aluminium described in the text. Crosses, detailed simulation; continuous curves, mixed simulation. $p(\theta)$ and $p(E)$ are the PDFs of the direction of motion (specified by the polar angle θ) and the kinetic energy E of the electrons, after travelling the prescribed 200 μm. $p(z)$ is the PDF of the z-coordinate of the final electron position, 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).

had travelled a path length s equal to 0.02 cm, 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 $s = 0.02$ cm 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 $E_{\rm abs} = 10$ keV and $s_{\rm max} = 40$ μm. Detailed simulation was performed by setting $C_1 = C_2 = 0$, $W_{\rm cc} = 0$ eV and $W_{\rm cr} = -100$ eV. Notice that when the user enters a negative value of the cutoff energy loss for radiative events, PENELOPE sets $W_{\rm cr} = 10$ eV, disregards the emission of soft bremsstrahlung photons with $W \leq 10$ eV (which represents a negligible fraction of the stopping power) and simulates bremsstrahlung events with $W > 10$ eV as usual, that is, in a detailed way. The mixed simulation results shown in Fig. 4.5 were generated with $C_1 = C_2 = 0.10$, $W_{\rm cc} = 2$ keV, $W_{\rm cr} = -100$ (*i.e.*, radiative events were described as in the detailed simulation), and $s_{\rm max} = 0.004$ cm.

In the detailed simulation, 100 million electron tracks were generated by running the program on a computer a with an Intel Core i7-3770 at 3.40 GHz for about 30 hours, which corresponds to a simulation speed of 925 tracks/s. The average numbers of elastic, inelastic and bremsstrahlung interactions that had to be simulated to produce each detailed track were 1297, 1181 and 0.03, respectively. On the same computer, the mixed simulation generated 100 Million tracks in about 37 minutes, which represents a simulation speed of 44,766 tracks/s, 48 times faster than that of detailed simulation. The reason for this higher speed is that, on average, there were only 4.71 hard elastic collisions, 3.86 hard inelastic collisions, 0.03 hard bremsstrahlung events and 3.31 delta interactions along each track. It is worth recalling that detailed simulation is nominally exact; the results are affected only by (small) statistical uncertainties. From Fig. 4.5 we conclude that, in this case, the mixed algorithm provides results that are essentially equivalent to those from the detailed simulation. Further calculations with different values of C_1 showed that the mixed algorithm is fairly stable under variations of this parameter over the accepted range (0,0.2). In most practical cases, where the track length of charged particles is not restricted, the gain in speed obtained from mixed simulation is much larger than the moderate gain found in the present study.

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 far from trivial and difficult to implement. 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 $s_{\rm max}$.

Chapter 5

Radiation fields and dosimetry

Monte Carlo simulation finds important applications in radiation dosimetry and metrology, which may require calculations of local and global characteristics of radiation fields and of the energy imparted by radiation to matter. In this Chapter we present basic concepts and quantities of common use in radiation dosimetry, and we consider their calculation by Monte Carlo methods. In previous Chapters we have introduced various macroscopic quantities (inverse mean free paths, stopping powers, energy straggling parameters, ...) that are defined as integrals of the interaction cross sections. Here the emphasis is on transport characteristics, that is, on quantities that involve the emission and transport of secondary radiation released in the interactions of energetic particles.

We essentially follow the terminology of the ICRU Report 85 (2011) on fundamental quantities and units for ionizing radiation. However, the sequence of definitions of radiometric quantities presented here is opposite to that adopted in the ICRU report. We start from the microscopic properties of the radiation field and the differential cross sections of the various interaction mechanisms, and we obtain macroscopic (average) quantities by integration. The discrete nature of matter and radiation fields (considered respectively as ensembles of individual atoms and energetic particles) is thus incorporated in a natural way.

Monte Carlo simulation is also based on a microscopic picture of radiation and matter. Simulation provides the most detailed description of radiation fields and of the spatial distributions of absorbed dose in material structures. It can be employed for calculating intermediate dosimetric quantities (such as linear energy absorption, photon mass energy-absorption coefficients, and linear energy transfers of charged particles) for different energies and materials. Programming a Monte Carlo calculation of these quantities helps to uncover the assumptions, and subtleties, underlying their definitions. Simulation also allows approximations to be avoided, such as the continuous-slowing down approximation for charged particles, which are frequently adopted in conventional calculations (see Kellerer *et al.*, 1992). Conversely, the so-called track-length estimators utilise macroscopic quantities for improving the efficiency of Monte Carlo calculations of absorbed dose and secondary particle emission (Sections 5.2.3 and 6.5).

5.1 Radiation fields

A *radiation field* is an ensemble of energetic particles of various types (photons, electrons, positrons, protons, neutrons, alphas, *etc.*). Each particle moves in a direction specified by the unit vector $\mathbf{\Omega}$ with radiant energy E, which is the energy of the particle excluding its rest energy. That is, for massive particles, E is the kinetic energy and, in the case of photons, E is simply the energy. In general, particles can be in different "inner" states: photons can have various polarisations, electrons and positrons can be in different spin-polarisation states, *etc.* In the following we shall assume that radiation fields are unpolarised and, consequently, that all definitions and equations include an implicit average over possible inner states of the particles. Because most radiometric quantities are additive, definitions will be given for radiation fields with particles of one type only. For mixed fields, with particles of different types, the definitions contain an implicit summation over the different types of particles.

A complete description of the radiation field is provided by the *particle density*, $n(t, \mathbf{r}, E, \mathbf{\Omega})$. The quantity $n(t, \mathbf{r}, E, \mathbf{\Omega})\,\mathrm{d}\mathbf{r}\,\mathrm{d}E\,\mathrm{d}\mathbf{\Omega}$ is the number of particles in the volume element $\mathrm{d}\mathbf{r}$ at $\mathbf{r}$ that move with directions within the solid angle element $\mathrm{d}\mathbf{\Omega}$ about the direction $\mathbf{\Omega}$ and have energies in the interval $(E, E + \mathrm{d}E)$ at time t. A classical particle of mass $M > 0$ that moves freely with velocity $\mathbf{v}_1$ is described by the particle density

$$n(t, \mathbf{r}, E, \mathbf{\Omega}) = \delta(\mathbf{r} - \mathbf{r}_1^0 - \mathbf{v}_1 t)\,\delta(\mathbf{\Omega} - \hat{\mathbf{v}}_1)\,\delta(E - E_1), \tag{5.1}$$

where $\mathbf{r}_1^0$ is the initial position vector (at $t = 0$), E_1 is the kinetic energy of the particle, and $\delta(x)$ is the Dirac delta distribution (*e.g.*, Schiff, 1968). We recall that

$$\delta(\mathbf{r} - \mathbf{r}') = \delta(x - x')\,\delta(y - y')\,\delta(z - z') \tag{5.2a}$$

and

$$\delta(\mathbf{\Omega} - \mathbf{\Omega}') = \delta(\cos\theta - \cos\theta')\,\delta(\phi - \phi'), \tag{5.2b}$$

where θ and ϕ are the polar and azimuthal angles of the unit vector $\mathbf{\Omega}$. The particle density of a swarm of N particles with initial positions $\mathbf{r}_i^0$, velocities $\mathbf{v}_i$, and energies E_i is

$$n(t, \mathbf{r}, E, \mathbf{\Omega}) = \sum_{i=1}^{N} \delta(\mathbf{r} - \mathbf{r}_i^0 - \mathbf{v}_i t)\,\delta(\mathbf{\Omega} - \hat{\mathbf{v}}_i)\,\delta(E - E_i). \tag{5.3}$$

In the case of classical particles, their instantaneous positions and velocities can be determined with arbitrarily high accuracy. The evolution of radiation fields, however, is governed by the laws of quantum mechanics, which imply that particles cannot be assigned definite values of position *and* momentum. Indeed, for non-relativistic particles, results from measurements of these two quantities yield random values with standard deviations satisfying Heisenberg's uncertainty relation,

$$\Delta r_i\,\Delta p_i \geq \frac{\hbar}{2} \qquad (i = x, y, z). \tag{5.4}$$

Thus, a particle with well-defined momentum has a poorly defined position. Interactions with the atoms of the medium can be regarded as position measurements (Schiff, 1968), which determine the position vector $\mathbf{r}$ of the particle with an uncertainty Δr_1 of the order of the atomic radius. The Thomas-Fermi model (Condon and Odabaşi, 1980) leads to the approximate expression $Z^{-1/3}a_0$ for the atomic radius, where a_0 is the Bohr radius. After the interaction, the components of the linear momentum have uncertainties $\Delta p_i \sim \hbar/(2\Delta r_i) = \hbar Z^{1/3}/(2a_0)$. In transport studies, $\mathbf{r}$ and $\mathbf{p}$ are considered as *independent* variables, that is, quantum correlations between position and momentum are disregarded. This simplification is acceptable only when the momentum of the particle is much larger than its uncertainty, *i.e.*, $p_i \gg \Delta p_i$. Hence, the trajectory picture underlying transport calculations and Monte Carlo simulation is valid for electrons and positrons with kinetic energies much higher than $(\Delta p_i)^2/(2\mathrm{m_e}) \sim 0.4Z^{2/3}E_\mathrm{h}$, where $E_\mathrm{h} = 27.2114$ eV is the Hartree energy.

5.1.1 Current density and flux density

The *angular current density* of a radiation field is defined by

$$\mathbf{j}(t,\mathbf{r},E,\mathbf{\Omega}) \equiv \mathbf{\Omega}\, v(E)\, n(t,\mathbf{r},E,\mathbf{\Omega}), \tag{5.5}$$

where $v(E)$ is the speed of particles with energy E. To clarify the physical meaning of the angular current density, we recall that an *oriented* surface element of area $\mathrm{d}A$ with normal unit vector $\hat{\mathbf{n}}$ is represented by the vector $\mathrm{d}\mathbf{A} = \hat{\mathbf{n}}\,\mathrm{d}A$. Let us consider a surface element $\mathrm{d}\mathbf{A}$ at the position $\mathbf{r}$ and a particle that at the time t is at the point $\mathbf{r}_1$ moving with velocity $\mathbf{v}_1 = v_1\mathbf{\Omega}_1$; in a time interval $\mathrm{d}t$, the particle will cross the surface element $\mathrm{d}\mathbf{A}$ at $\mathbf{r}$ only if the point $\mathbf{r}_1$ is within the space region swept by the surface $\mathrm{d}\mathbf{A}$ when it is translated a distance $-\mathbf{v}_1\mathrm{d}t$ (see Fig. 5.1); the volume of this region is $|\mathbf{v}_1\cdot\mathrm{d}\mathbf{A}|\mathrm{d}t = |\hat{\mathbf{n}}\cdot\mathbf{\Omega}_1|\, v_1\mathrm{d}t\,\mathrm{d}A$. Hence, the quantity

$$dN \equiv \mathbf{j}(t,\mathbf{r},E,\Omega)\cdot\mathrm{d}\mathbf{A}\,\mathrm{d}E\,\mathrm{d}\mathbf{\Omega}\,\mathrm{d}t = v(E)\, n(t,\mathbf{r},E,\mathbf{\Omega})\,(\hat{\mathbf{n}}\cdot\mathbf{\Omega})\,\mathrm{d}A\,\mathrm{d}E\,\mathrm{d}\mathbf{\Omega}\,\mathrm{d}t \tag{5.6}$$

is the net number of particles in the ranges $\mathrm{d}E$ and $\mathrm{d}\mathbf{\Omega}$ that cross the surface element $\mathrm{d}\mathbf{A}$ placed at $\mathbf{r}$ during the time interval $\mathrm{d}t$. Note that particles moving in directions $\mathbf{\Omega}$ such that $\hat{\mathbf{n}}\cdot\mathbf{\Omega} > 0$ (< 0) give positive (negative) contributions.

In most practical cases, the radiation field is present only during a finite time interval, *i.e.*, $n(t,\mathbf{r},E,\mathbf{\Omega}) = 0$ when $t \to \pm\infty$, and one is usually interested in total quantities, integrated over time. Let us consider a finite volume $\mathcal{V}$ limited by a closed surface $\mathcal{S}$; an element of this surface is represented by a vector $\mathrm{d}\mathbf{A} \equiv \hat{\mathbf{n}}\,\mathrm{d}A$ where $\hat{\mathbf{n}}$ is the outward normal to the surface. The total number of particles that enter the volume $\mathcal{V}$ through its limiting surface is

$$\begin{aligned} N_\mathrm{in}(\mathcal{S}) &= -\int_0^\infty \mathrm{d}E \int \mathrm{d}\mathbf{\Omega} \int_{-\infty}^\infty \mathrm{d}t \int_\mathcal{S} \mathrm{d}\mathbf{A}\cdot\mathbf{j}(t,\mathbf{r},E,\mathbf{\Omega})\Theta(-\hat{\mathbf{n}}\cdot\mathbf{\Omega}) \\ &= -\int_0^\infty \mathrm{d}E \int \mathrm{d}\mathbf{\Omega} \int_{-\infty}^\infty \mathrm{d}t \int_\mathcal{S} \mathrm{d}A\,(\hat{\mathbf{n}}\cdot\mathbf{\Omega})v(E)\, n(t,\mathbf{r},E,\mathbf{\Omega})\Theta(-\hat{\mathbf{n}}\cdot\mathbf{\Omega}), \end{aligned} \tag{5.7}$$

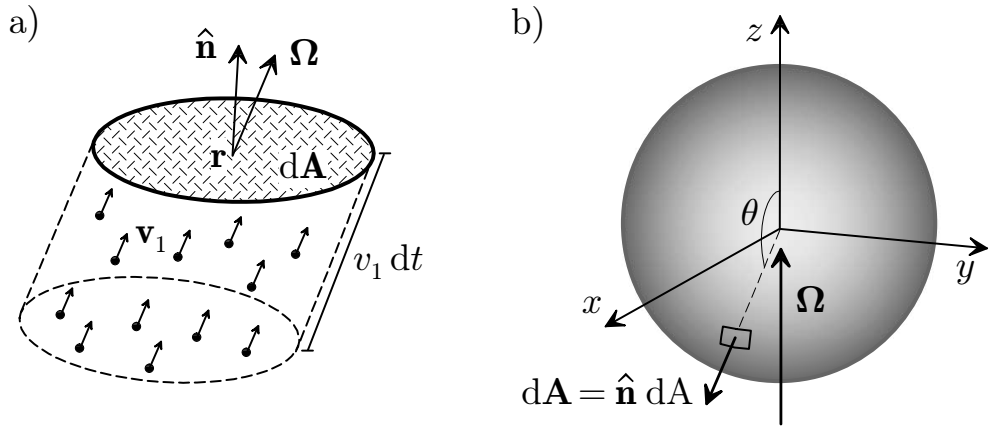

Figure 5.1: Surfaces in a radiation field. a) A surface element d$\mathbf{A}$ in a field of particles moving with the same velocity $\mathbf{v}_1$. In a time interval dt the surface element is crossed by the particles that initially are within the volume that would be swept by the surface if it were moving with velocity $-\mathbf{v}_1$. b) Schematic diagram of the coordinate system used to calculate the integral of Eq. (5.8) over the surface of the sphere.

where we have introduced the unit step function $\Theta(x)$ (= 1 if $x > 0$, = 0 otherwise) to count only particles that enter the volume (*i.e.*, with directions $\boldsymbol{\Omega}$ such that $\hat{\mathbf{n}}\cdot\boldsymbol{\Omega} < 0$). If we remove this step function, particles that enter and leave the volume would not contribute, and the integral would give the net number of particles absorbed in $\mathcal{V}$ (*i.e.*, the number of particles entering that are absorbed minus the number of particles generated in the volume). When $\mathcal{S}$ is a sphere of small radius r_s, the surface integral in (5.7) can be easily evaluated by using spherical polar coordinates with the origin at the centre of the sphere and the polar axis along the direction of $\boldsymbol{\Omega}$ (see Fig. 5.1), so that $\hat{\mathbf{n}}\cdot\boldsymbol{\Omega} = \cos\theta$. We have

$$\begin{aligned} -\int_{\mathcal{S}} \mathrm{d}A\,(\hat{\mathbf{n}}\cdot\boldsymbol{\Omega})\,n(t,\mathbf{r},E,\boldsymbol{\Omega})\Theta(-\hat{\mathbf{n}}\cdot\boldsymbol{\Omega}) &= 2\pi r_\mathrm{s}^2 \int_{\pi/2}^{\pi} \mathrm{d}\theta\,\sin\theta\,(-\cos\theta)\,n(t,\mathbf{r},E,\boldsymbol{\Omega}) \\ &= \pi r_\mathrm{s}^2\,n(t,\mathbf{r},E,\boldsymbol{\Omega}). \end{aligned} \tag{5.8}$$

Hence

$$N_\mathrm{in}(\mathrm{sphere}) = \pi r_\mathrm{s}^2 \int_0^\infty \mathrm{d}E \int \mathrm{d}\boldsymbol{\Omega} \int_{-\infty}^{\infty} \mathrm{d}t\,v(E)\,n(t,\mathbf{r},E,\boldsymbol{\Omega}), \tag{5.9}$$

where πr_s^2 is the cross-sectional area of the sphere.

Analogously, the total radiant energy that enters the volume $\mathcal{V}$ through its surface is given by

$$R_\mathrm{in}(\mathcal{S}) = -\int_0^\infty \mathrm{d}E \int \mathrm{d}\boldsymbol{\Omega} \int_{-\infty}^{\infty} \mathrm{d}t \int_{\mathcal{S}} \mathrm{d}\mathbf{A}\cdot\mathbf{j}(t,\mathbf{r},E,\boldsymbol{\Omega})\,E\,\Theta(-\hat{\mathbf{n}}\cdot\boldsymbol{\Omega}). \tag{5.10}$$

In the case of a small sphere with radius r_s,

$$R_\mathrm{in}(\mathrm{sphere}) = \pi r_\mathrm{s}^2 \int_0^\infty \mathrm{d}E \int \mathrm{d}\boldsymbol{\Omega} \int_{-\infty}^{\infty} \mathrm{d}t\,v(E)\,n(t,\mathbf{r},E,\boldsymbol{\Omega})\,E. \tag{5.11}$$

The angular current density is awkward to handle because of its vector nature. A more convenient quantity is the *angular flux density*, which is defined as the magnitude of the angular current density,

$$\Phi(t, \mathbf{r}, E, \mathbf{\Omega}) \equiv v(E)\, n(t, \mathbf{r}, E, \mathbf{\Omega}). \tag{5.12}$$

Notice that $\Phi(t, \mathbf{r}, E, \mathbf{\Omega})\, \mathrm{d}E\, \mathrm{d}\mathbf{\Omega}\, \mathrm{d}A\, \mathrm{d}t$ is the number of particles in the ranges $\mathrm{d}E$ and $\mathrm{d}\mathbf{\Omega}$ that in the time interval $\mathrm{d}t$ cross a small surface element $\mathrm{d}\mathbf{A}$ placed at $\mathbf{r}$ and perpendicular to $\mathbf{\Omega}$, *i.e.*, $\mathrm{d}\mathbf{A} = \mathbf{\Omega}\, \mathrm{d}A$. The SI unit of angular flux density is $(\mathrm{J{\cdot}m^2{\cdot}s})^{-1}$. The integral of the angular flux density over directions is the *flux density*,

$$\Phi(t, \mathbf{r}, E) \equiv \int \mathrm{d}\mathbf{\Omega}\, \Phi(t, \mathbf{r}, E, \mathbf{\Omega}). \tag{5.13}$$

That is, $\Phi(t, \mathbf{r}, E)\, \mathrm{d}A\, \mathrm{d}E\, \mathrm{d}t$ gives the number of particles with energies in $(E, E + \mathrm{d}E)$ that cross a small surface of area $\mathrm{d}A$, placed at r and perpendicular to the direction of motion *of each particle*, during the time interval $\mathrm{d}t$. Additionally, according to Eq. (5.9), $\Phi(t, \mathbf{r}, E)\, \mathrm{d}A\, \mathrm{d}E\, \mathrm{d}t$ is the number of particles incident on a small sphere of cross-sectional area $\mathrm{d}A$, centred at $\mathbf{r}$, with energies in $(E, E + \mathrm{d}E)$ during time $\mathrm{d}t$.

5.1.2 Radiometric quantities

The angular flux density, $\Phi(t, \mathbf{r}, E, \mathbf{\Omega})$, and the angular current density, $\mathbf{j}(t, \mathbf{r}, E, \mathbf{\Omega}) = \mathbf{\Omega}\, \Phi(t, \mathbf{r}, E, \mathbf{\Omega})$, provide the most detailed description of a radiation field. Other radiometric quantities can be expressed as integrals of these fundamental functions over appropriate ranges of their arguments. As mentioned above, we follow the terminology of the ICRU Report 85 (2011) for radiometric quantities. However, we adopt a more explicit notation, where the dependence of a quantity on the variables t, $\mathbf{r}$, E, and $\mathbf{\Omega}$ (when applicable) is indicated in the list of arguments.

In the ICRU Report 85, the angular flux density, (5.12), is denoted by $\dot{\Phi}_{\mathbf{\Omega},E}$ and is called the *distribution of the scalar particle radiance with respect to energy*. The quantity

$$\Psi(t, \mathbf{r}, E, \mathbf{\Omega}) \equiv E\Phi(t, \mathbf{r}, E, \mathbf{\Omega}) \tag{5.14}$$

[$\dot{\Psi}_{\mathbf{\Omega},E}$ in the ICRU notation] is the *distribution of the scalar energy radiance with respect to energy*. The ICRU Report 85 also considers vector quantities, which are expressed as the product of a scalar function and the direction vector $\mathbf{\Omega}$. Thus, from the scalar quantities (5.12) and (5.14) we can define the *distribution of the vector particle radiance with respect to energy*,

$$\mathbf{\Omega}\ \Phi(t, \mathbf{r}, E, \mathbf{\Omega}) = \mathbf{j}(t, \mathbf{r}, E, \mathbf{\Omega}), \tag{5.15}$$

and the *distribution of the vector energy radiance with respect to energy*,

$$\mathbf{\Omega}\, \Psi(t, \mathbf{r}, E, \mathbf{\Omega}) = \mathbf{\Omega}\, E\Phi(t, \mathbf{r}, E, \mathbf{\Omega}). \tag{5.16}$$

Although vector quantities are useful in transport theory, they are of limited interest in radiation dosimetry because most radiation effects are independent of the particle direction.

The *fluence* $\Phi(\mathbf{r})$ at the point $\mathbf{r}$ is defined as

$$\Phi(\mathbf{r}) \equiv \frac{\mathrm{d}N_{\mathrm{in}}}{\mathrm{d}A}, \tag{5.17}$$

where $\mathrm{d}N_{\mathrm{in}}$ is the total number of particles incident on a small sphere $\mathcal{S}$ of cross-sectional area $\mathrm{d}A$, centred at $\mathbf{r}$. The unit of fluence in the SI is m^{-2}. Equation (5.9) implies that

$$\begin{aligned}\Phi(\mathbf{r}) &= \int_{-\infty}^{\infty}\mathrm{d}t\int_{0}^{\infty}\mathrm{d}E\int\mathrm{d}\boldsymbol{\Omega}\,\Phi(t,\mathbf{r},E,\boldsymbol{\Omega})\\ &= \int_{-\infty}^{\infty}\mathrm{d}t\int_{0}^{\infty}\mathrm{d}E\int\mathrm{d}\boldsymbol{\Omega}\,v(E)\,n(t,\mathbf{r},E,\boldsymbol{\Omega}).\end{aligned} \tag{5.18}$$

The *average fluence* in a finite volume $\mathcal{V}$ is

$$\overline{\Phi} \equiv \frac{1}{\mathcal{V}}\int_{\mathcal{V}}\mathrm{d}\mathbf{r}\,\Phi(\mathbf{r}). \tag{5.19}$$

That is,

$$\overline{\Phi} = \frac{1}{\mathcal{V}}\int_{\mathcal{V}}\mathrm{d}\mathbf{r}\int_{0}^{\infty}\mathrm{d}E\int\mathrm{d}\boldsymbol{\Omega}\int_{-\infty}^{\infty}\mathrm{d}t\,v(E)n(t,\mathbf{r},E,\boldsymbol{\Omega}). \tag{5.20}$$

To reveal the physical significance of $\overline{\Phi}$, we consider a field of free, non-interacting classical particles with density [cf. Eq. (5.1)]

$$n(t,\mathbf{r},E,\boldsymbol{\Omega}) = \sum_i \delta(\mathbf{r}-\mathbf{r}_i^0-\mathbf{v}_i t)\,\delta(\boldsymbol{\Omega}-\hat{\mathbf{v}}_i)\,\delta(E-E_i), \tag{5.21}$$

where $\mathbf{r}_i^0$, $\mathbf{v}_i$ and E_i are, respectively, the initial position vector (at $t=0$), the velocity and the energy of the i-th particle. The average fluence in a volume $\mathcal{V}$ is

$$\begin{aligned}\overline{\Phi} &= \frac{1}{\mathcal{V}}\int_{\mathcal{V}}\mathrm{d}\mathbf{r}\int_{0}^{\infty}\mathrm{d}E\int\mathrm{d}\boldsymbol{\Omega}\int_{-\infty}^{\infty}\mathrm{d}t\,v(E)\sum_i\delta(\mathbf{r}-\mathbf{r}_i^0-\mathbf{v}_i t)\,\delta(\boldsymbol{\Omega}-\hat{\mathbf{v}}_i)\,\delta(E-E_i)\\ &= \frac{1}{\mathcal{V}}\sum_i\int_{\mathcal{V}}\mathrm{d}\mathbf{r}\int_{-\infty}^{\infty}\mathrm{d}t\,v_i\,\delta(\mathbf{r}-\mathbf{r}_i^0+\mathbf{v}_i t)\,.\end{aligned} \tag{5.22}$$

The integral over $\mathbf{r}$ is now trivial: the delta function $\delta(\mathbf{r}-\mathbf{r}_i^0+\mathbf{v}_i t)$ simply implies that a particle contributes only when its trajectory intersects the volume $\mathcal{V}$. Furthermore, the product $\mathrm{d}t\,v_i$ is the path length $\mathrm{d}s$ that particle i travels *within the volume* $\mathcal{V}$ during the time $\mathrm{d}t$. Hence,

$$\overline{\Phi} = \frac{1}{\mathcal{V}}\sum_i(\text{path length of particle } i \text{ in } \mathcal{V}). \tag{5.23}$$

That is, the average fluence gives the total path length of particles per unit volume. It has dimensions of $(\text{surface})^{-1}$ and is expressed in m^{-2} or a multiple of it. The equality (5.23) is valid also for real radiation fields, provided only that particles follow classical

trajectories, $\mathbf{r}_i(t) = \mathbf{r}_i^0 + \mathbf{v}_i\, t$, between consecutive interactions. Expression (5.23) can be used to estimate the average fluence in Monte Carlo simulations. Normally, the efficiency of the simulation decreases when the detector volume $\mathcal{V}$ is reduced, because fewer and fewer particles contribute to the fluence score. That is, Monte Carlo codes are only able to provide average fluences in *finite* volumes.

The *energy fluence* is defined by

$$\begin{aligned}\Psi(\mathbf{r}) &\equiv \int_{-\infty}^{\infty} \mathrm{d}t \int_0^{\infty} \mathrm{d}E \int \mathrm{d}\boldsymbol{\Omega}\, E\, \Phi(t, \mathbf{r}, E, \boldsymbol{\Omega}) \\ &= \int_0^{\infty} \mathrm{d}E \int \mathrm{d}\boldsymbol{\Omega} \int_{-\infty}^{\infty} \mathrm{d}t\, E\, v(E)\, n(t, \mathbf{r}, E, \boldsymbol{\Omega}). \end{aligned} \tag{5.24}$$

Note that

$$\Psi(\mathbf{r}) = \frac{\mathrm{d}R_{\mathrm{in}}}{\mathrm{d}A}, \tag{5.25}$$

where $\mathrm{d}R_{\mathrm{in}}$ is the total radiant energy of particles incident on a small sphere of cross-sectional area $\mathrm{d}A$, centred at $\mathbf{r}$. The *average energy fluence* in a finite volume $\mathcal{V}$ is

$$\overline{\Psi} \equiv \frac{1}{\mathcal{V}} \int_{\mathcal{V}} \mathrm{d}\mathbf{r}\, \Psi(\mathbf{r}). \tag{5.26}$$

That is,

$$\overline{\Psi} = \frac{1}{\mathcal{V}} \int_{\mathcal{V}} \mathrm{d}\mathbf{r} \int_0^{\infty} \mathrm{d}E \int \mathrm{d}\boldsymbol{\Omega} \int_{-\infty}^{\infty} \mathrm{d}t\, E\, v(E)\, n(t, \mathbf{r}, E, \boldsymbol{\Omega}). \tag{5.27}$$

A derivation parallel to that of Eq. (5.23) leads to

$$\overline{\Psi} = \frac{1}{\mathcal{V}} \sum_{ik} E_{ik} \times (\text{path length of particle } i \text{ with energy } E_{ik} \text{ in } \mathcal{V}), \tag{5.28}$$

where the subscript k denotes the different energies acquired by a particle along its trajectory.

The *fluence rate*, $\Phi(t, \mathbf{r})$, and the *energy-fluence rate* $\Psi(t, \mathbf{r})$ are defined as

$$\Phi(t, \mathbf{r}) \equiv \int \mathrm{d}\boldsymbol{\Omega} \int_0^{\infty} \mathrm{d}E\, \Phi(t, \mathbf{r}, E, \boldsymbol{\Omega}) = \int \mathrm{d}\boldsymbol{\Omega} \int_0^{\infty} \mathrm{d}E\, v(E)\, n(t, \mathbf{r}, E, \boldsymbol{\Omega})\,, \tag{5.29}$$

and

$$\Psi(t, \mathbf{r}) \equiv \int \mathrm{d}\boldsymbol{\Omega} \int_0^{\infty} \mathrm{d}E\, \Psi(t, \mathbf{r}, E, \boldsymbol{\Omega}) = \int \mathrm{d}\boldsymbol{\Omega} \int_0^{\infty} \mathrm{d}E\, E\, v(E)\, n(t, \mathbf{r}, E, \boldsymbol{\Omega}), \tag{5.30}$$

respectively. The physical significance of these quantities is made clear by considering a small sphere of cross-sectional area $\mathrm{d}A$ centred at $\mathbf{r}$. $\Phi(t, \mathbf{r})\, \mathrm{d}A$ and $\Psi(t, \mathbf{r})\, \mathrm{d}A$ are then, respectively, the number of particles and the radiant energy incident on the sphere per unit time.

For a monoenergetic radiation field composed of particles with radiant energy E_0,

$$n(t, \mathbf{r}, E, \mathbf{\Omega}) = n(t, \mathbf{r}, \mathbf{\Omega})\, \delta(E - E_0), \tag{5.31}$$

the fluence rate and the energy-fluence rate are, respectively,

$$\Phi(t, \mathbf{r}) = v(E_0)\, \overline{n}(t, \mathbf{r}) \qquad \text{and} \qquad \Psi(t, \mathbf{r}) = E_0\, v(E_0)\, \overline{n}(t, \mathbf{r})\,, \tag{5.32}$$

where

$$\overline{n}(t, \mathbf{r}) = \int \mathrm{d}\mathbf{\Omega}\, n(t, \mathbf{r}, \mathbf{\Omega}) \tag{5.33}$$

is the particle number density (*i.e.*, the number of particles per unit volume).

The *particle radiance* $\Phi(t, \mathbf{r}, \mathbf{\Omega})$ is defined by

$$\Phi(t, \mathbf{r}, \mathbf{\Omega}) \equiv \int_0^\infty \mathrm{d}E\, \Phi(t, \mathbf{r}, E, \mathbf{\Omega}) = \int_0^\infty \mathrm{d}E\, v(E)\, n(t, \mathbf{r}, E, \mathbf{\Omega}), \tag{5.34}$$

and the *energy radiance* is

$$\Psi(t, \mathbf{r}, \mathbf{\Omega}) \equiv \int_0^\infty \mathrm{d}E\, \Psi(t, \mathbf{r}, E, \mathbf{\Omega}) = \int_0^\infty \mathrm{d}E\, E\, v(E)\, n(t, \mathbf{r}, E, \mathbf{\Omega}). \tag{5.35}$$

The quantities $\Phi(t, \mathbf{r}, \mathbf{\Omega})\, \mathrm{d}A$ and $\Psi(t, \mathbf{r}, \mathbf{\Omega})\, \mathrm{d}A$ are, respectively, the number of particles and the radiant energy that enter a small sphere of cross sectional area $\mathrm{d}A$ per unit time and per unit solid angle in the direction $\mathbf{\Omega}$.

The *distribution of fluence with respect to energy* is defined as

$$\Phi(\mathbf{r}, E) \equiv \int_{-\infty}^\infty \mathrm{d}t \int \mathrm{d}\mathbf{\Omega}\, \Phi(t, \mathbf{r}, E, \mathbf{\Omega}) = \int_{-\infty}^\infty \mathrm{d}t \int \mathrm{d}\mathbf{\Omega}\, v(E)\, n(t, \mathbf{r}, E, \mathbf{\Omega}), \tag{5.36}$$

and the *distribution of energy fluence with respect to energy* is

$$\Psi(\mathbf{r}, E) \equiv \int_{-\infty}^\infty \mathrm{d}t \int \mathrm{d}\mathbf{\Omega}\, \Psi(t, \mathbf{r}, E, \mathbf{\Omega}) = \int_{-\infty}^\infty \mathrm{d}t \int \mathrm{d}\mathbf{\Omega}\, E\, v(E)\, n(t, \mathbf{r}, E, \mathbf{\Omega})\,. \tag{5.37}$$

The quantities $\Phi(\mathbf{r}, E)\, \mathrm{d}E\, \mathrm{d}A$ and $\Psi(\mathbf{r}, E)\, \mathrm{d}E\, \mathrm{d}A$ are, respectively, the number of particles and the radiant energy that enter a small sphere of cross sectional area $\mathrm{d}A$ with energies in the interval $(E, E + \mathrm{d}E)$.

The distribution of fluence with respect to energy $\Phi(\mathbf{r}, E)$ is used in dosimetry to calculate important quantities such as the absorbed dose and the kerma (see below). In a Monte Carlo simulation we can only determine the *average distribution of fluence with respect to energy* in a finite volume $\mathcal{V}$,

$$\overline{\Phi}(E) \equiv \frac{1}{\mathcal{V}} \int_{\mathcal{V}} \mathrm{d}\mathbf{r}\, \Phi(\mathbf{r}, E) = \frac{1}{\mathcal{V}} \int_{\mathcal{V}} \mathrm{d}\mathbf{r} \int_{-\infty}^\infty \mathrm{d}t \int \mathrm{d}\mathbf{\Omega}\, v(E)\, n(t, \mathbf{r}, E, \mathbf{\Omega}). \tag{5.38}$$

In the limit of small volumes, $\overline{\Phi}(E)$ reduces to $\Phi(\mathbf{r}, E)$. Since $\overline{\Phi}(E)$ is a function of the continuous variable E, it is generated as a histogram (see Section 1.5). Arguments similar to those employed in the derivation of Eq. (5.23) show that

$$\overline{\Phi}(E)\,\Delta E \simeq \int_E^{E+\Delta E} \overline{\Phi}(E')\,\mathrm{d}E' = \frac{1}{\mathcal{V}} \sum_i \left(\begin{array}{l} \text{path length of particle } i \text{ while it is} \\ \text{inside } \mathcal{V} \text{ and its radiant energy } E_i \\ \text{is in the interval } (E, E+\Delta E) \end{array} \right). \tag{5.39}$$

Therefore, $\overline{\Phi}(E)\,\Delta E$ is the total path length per unit volume of field particles having radiant energies in the interval $(E, E+\Delta E)$. In class II simulation of electrons and positrons (and other charged particles), the kinetic energy of the transported particle varies continuously along each trajectory segment and the contributions to $\overline{\Phi}(E)\,\Delta E$ should be evaluated as described in Section 4.3.3.

For intense fields, the fluence, $\Phi(\mathbf{r})$, and the distribution of fluence with respect to energy, $\Phi(\mathbf{r}, E)$, are continuous functions of their arguments. In this case,

$$\lim_{\mathcal{V}\to 0} \overline{\Phi} = \Phi(\mathbf{r})\,, \tag{5.40}$$

and

$$\lim_{\mathcal{V}\to 0} \overline{\Phi}(E)\,\Delta E = \Phi(\mathbf{r}, E)\,\Delta E\,, \tag{5.41}$$

where $\mathbf{r}$ is the position vector of a point in the small volume $\mathcal{V}$.

The *flux* $N_{\mathcal{S}}(t)$ of particles across a surface $\mathcal{S}$ is defined as the net number of particles that cross the surface per unit time,

$$\begin{aligned} N_{\mathcal{S}}(t) &\equiv \int_0^\infty \mathrm{d}E \int \mathrm{d}\mathbf{\Omega} \int_{\mathcal{S}} \mathrm{d}A\,(\hat{\mathbf{n}}\cdot\mathbf{\Omega})\Phi(t, \mathbf{r}, E, \mathbf{\Omega}) \\ &= \int_0^\infty \mathrm{d}E \int \mathrm{d}\mathbf{\Omega} \int_{\mathcal{S}} \mathrm{d}A\,(\hat{\mathbf{n}}\cdot\mathbf{\Omega}) v(E)\, n(t, \mathbf{r}, E, \mathbf{\Omega}), \end{aligned} \tag{5.42}$$

where $\hat{\mathbf{n}}$ is the unit vector normal to the surface. The *energy flux* $R_t(\mathcal{S})$ is the net radiant energy that flows across the surface per unit time,

$$R_{\mathcal{S}}(t) \equiv \int_0^\infty \mathrm{d}E \int \mathrm{d}\mathbf{\Omega} \int_{\mathcal{S}} \mathrm{d}A\,(\hat{\mathbf{n}}\cdot\mathbf{\Omega}) E\, v(E)\, n(t, \mathbf{r}, E, \mathbf{\Omega}). \tag{5.43}$$

The number of particles and the radiant energy that cross the surface during the time interval (t_1, t_2) are

$$N_{\mathcal{S}} = \int_{t_1}^{t_2} \mathrm{d}t\, N_{\mathcal{S}}(t) \qquad \text{and} \qquad R_{\mathcal{S}} = \int_{t_1}^{t_2} \mathrm{d}t\, R_{\mathcal{S}}(t), \tag{5.44}$$

respectively.

5.2 Monte Carlo simulation and dosimetry

The fundamental quantity in radiation dosimetry is the *absorbed dose*, $D(\mathbf{r})$, which is defined as the mean energy absorbed (*i.e.*, deposited onto matter) per unit mass at $\mathbf{r}$ (ICRU Report 85, 2011). Considering a small volume element, $\mathrm{d}\mathbf{r}$ around the point $\mathbf{r}$ and the average energy, $\mathrm{d}\overline{\varepsilon}$, absorbed by the matter in $\mathrm{d}\mathbf{r}$, we have

$$D(\mathbf{r}) \equiv \frac{\mathrm{d}\overline{\varepsilon}}{\mathrm{d}m} = \frac{1}{\rho(\mathbf{r})}\,\frac{\mathrm{d}\overline{\varepsilon}}{\mathrm{d}\mathbf{r}}, \tag{5.45}$$

where $\mathrm{d}m$ is the mass contained in $\mathrm{d}\mathbf{r}$, and $\rho(\mathbf{r}) = \mathrm{d}m/\mathrm{d}\mathbf{r}$ is the mass density. In the SI, the special name for the unit of absorbed dose is the gray, 1 Gy = 1 J/kg. When radiant energies of particles are given in eV, absorbed doses are frequently expressed in eV/g,

$$1\ \mathrm{eV/g} = 1.602\,176\ \times 10^{-16}\ \mathrm{Gy}\,, \tag{5.46}$$

or in some multiple or submultiple, such as keV/μg. Although $D(\mathbf{r})$ is defined as a function of position, measurements are usually performed with dosimeters having finite sensitive volumes, $\mathcal{V}$. After appropriate transformations, the output of these devices gives the absorbed dose in the volume $\mathcal{V}$,

$$D = \frac{1}{M}\int_{\mathcal{V}} D(\mathbf{r})\,\rho(\mathbf{r})\,\mathrm{d}\mathbf{r} = \frac{\overline{\varepsilon}}{M}, \tag{5.47}$$

where

$$M = \int_{\mathcal{V}} \rho(\mathbf{r})\,\mathrm{d}\mathbf{r} \tag{5.48}$$

is the total mass contained in $\mathcal{V}$, and $\overline{\varepsilon}$ is the mean energy absorbed by that mass.

Because of the discrete nature of radiation fields and the randomness of interactions, the concept of absorbed dose is useful only when the number of interactions within the sensitive volume of the dosimeter, $\mathrm{d}\mathbf{r}$ or $\mathcal{V}$, is "statistically sufficient", *i.e.*, so large that the mean energy absorbed, $\overline{\varepsilon}$, is well defined. In the case of weak fields (with small fluences) or small sensitive volumes, repeated measurements of the energy ϵ absorbed in a given volume under identical conditions may give widely fluctuating results. Under these conditions, it is more appropriate to consider the *specific energy absorbed* [1] by the matter in a volume containing a total mass M,

$$z \equiv \frac{\varepsilon}{M}\,. \tag{5.49}$$

Note that z is a stochastic quantity, and is meaningful even when the number of interactions with the mass M is small. Evidently, $D = \overline{z}$ and, in the limit of small volumes, the absorbed dose $D(\mathbf{r})$ is equal to the mean specific energy absorbed at $\mathbf{r}$.

[1] With the ICRU conventions, the term *specific* indicates *per unit mass*, and the quantity z is named the specific energy imparted. We prefer to call it "specific energy absorbed", which emphasizes the fact that it excludes the energy that leaves the sensitive volume $\mathcal{V}$ of the dosimeter in the form of secondary radiations. For instance, consider the photoelectric absorption of a photon of energy E followed by the emission of a characteristic x ray of energy E_{x} from the target atom. When that x ray leaves the volume $\mathcal{V}$, the energy imparted is E, while the energy absorbed is only $E - E_{\mathrm{x}}$.

We have already mentioned that Monte Carlo simulation offers the most direct method for computing the absorbed dose in finite volumes. In Monte Carlo calculations the tracking of particles P is discontinued when their radiant energy E falls below the adopted absorption energy $E_{\mathrm{abs},P}$. The absorption energies must be finite because the number of particles in the shower induced by a primary particle, and the simulation time spent to generate a complete shower (primary particle and all its descendants), increase without limit when $E_{\mathrm{abs},P}$ decreases. Normally, the spatial distribution of absorbed dose is tallied using volume bins with a certain "thickness" ℓ_{b} (*e.g.*, the edge length for cubic voxels). The absorption energy, $E_{\mathrm{abs},P}$, should be selected so that particles P with radiant energy equal to $E_{\mathrm{abs},P}$ deposit all of their energy within distances that are much less than ℓ_{b}. Thus, particles with energy less than $E_{\mathrm{abs},P}$ are absorbed inside the same bin where they are released, and distortion of local absorbed dose caused by the finite values of the absorption energies is expected to be small. That is to say, in Monte Carlo calculations we may consider that particles with energies below $E_{\mathrm{abs},P}$ are absorbed on the spot.

The central problem in classical dosimetry is to obtain the absorbed dose $D(\mathbf{r})$ from knowledge of the distributions of fluence with respect to energy, $\Phi_P(\mathbf{r}, E)$, of the particles P present in the radiation field. The fluence distribution can be regarded as a surrogate of dose; it provides a complete description of the radiation field, from which we can calculate the absorbed dose in any material medium. It is employed, for instance, to transform the reading of an ionisation chamber into the absorbed dose in the medium in the absence of the chamber.

Nowadays, Monte Carlo simulation allows the absorbed dose in finite volume bins to be tallied directly from the simulated particle histories at practically no cost; a result of this ease is that not all Monte Carlo users are familiar with the approximations underlying the practical transformation from the fluence distribution to absorbed dose. In the following we introduce intermediate quantities for the calculation of the absorbed dose from the distribution of fluence with respect to energy, and describe the calculation of these quantities by means of Monte Carlo simulation. These quantities are also useful in some Monte Carlo simulations where the calculation of the fluence is more efficient than the straight evaluation of the absorbed dose (see Section 5.2.3).

For simplicity, we assume that the radiation field contains only photons, electrons and positrons, which will be indicated by the respective labels $P = \gamma$, $-$, and $+$. We also consider that the only relevant interactions are those included in the PENELOPE simulation algorithm. The formulation may be readily generalised to other radiation (neutrons, protons, *etc.*) and to include other interaction mechanisms (*e.g.*, photonuclear interactions).

Usually, the distribution of fluence with respect to energy is known only for particles P with radiant energies higher than a certain cutoff energy, Δ_P. In calculations of the absorbed dose, particles with sub-cutoff energies (*i.e.*, with $E < \Delta_P$) are assumed to be absorbed on the spot. We would like to point out that, when the fluence distribution is generated by Monte Carlo simulation, the cutoffs Δ_P do not need to be equal to the

absorption energies $E_{\mathrm{abs},P}$ adopted in the simulation (see below). Let $\Phi_{\mathrm{sr},P}(\mathbf{r}, E)$ denote the fluence distribution with respect to the energy of particles P from the radiation source (*e.g.*, from an external beam or from nuclear decays within the material system), which may extend to energies E below Δ_P. The total distribution of fluence of particles P is

$$\Phi_P(\mathbf{r}, E) = \Phi_{\mathrm{sr},P}(\mathbf{r}, E) + \Phi_{\mathrm{sh},P}(\mathbf{r}, E)\,, \tag{5.50}$$

where $\Phi_{\mathrm{sh},P}(\mathbf{r}, E)$ is the contribution of particles P in the showers originated by the source particles. We assume that $\Phi_{\mathrm{sh},P}(\mathbf{r}, E)$ is known only for $E > \Delta_P$.

We recall that $\Phi_P(\mathbf{r}, E)\,\mathrm{d}E$ is the path length of particles P with energies in the interval $(E, E + \mathrm{d}E)$ per unit volume. Hence, $\rho^{-1}\Phi_P(\mathbf{r}, E)\,\mathrm{d}E$ is the path length of these particles per unit mass. To obtain the absorbed dose, we only have to multiply this quantity by the mean energy that is effectively absorbed (converted into atomic excitations and sub-cutoff particles) per unit path length of particles P.

It is thus natural to consider the *linear energy absorption*[2] (LEA) for particles P of radiant energy E, $A_{\Delta,P}(E)$, defined as the average energy that is transferred to the material per unit path length, and is not re-emitted as particles P' having radiant energy higher than the corresponding cutoff $\Delta_{P'}$. The LEA can be given in units of eV/cm, or a multiple of it. Although a similar quantity, the linear energy transfer or LET, is used only for charged particles (ICRU Report 85, 2011), the concept is also applicable to uncharged particles. With this generalisation, the contribution to the absorbed dose $D(\mathbf{r})$ of particles P with energies in the interval $(E, E + \mathrm{d}E)$ is given by the product $A_{\Delta,P}(E)\,\rho^{-1}\,\Phi_P(\mathbf{r}, E)\,\mathrm{d}E$. Note that the LEA depends on the values of the cutoff energies $\Delta_{P'}$ of all radiation P' that can be released following the interactions of particles P; the subscript Δ indicates the set of cutoff energies. Except for a slight correction (the track-end term, D_{end}, see below), the absorbed dose can now be expressed as

$$D(\mathbf{r}) = \frac{1}{\rho(\mathbf{r})} \sum_P \int A_{\Delta,P}(E)\,\Phi_P(\mathbf{r}, E)\,\mathrm{d}E\,. \tag{5.51}$$

Because the LEA is essentially proportional to the mass density of the medium, it is convenient to consider the *mass energy absorption* (MEA), $B_{\Delta,P}(E)$, defined by

$$B_{\Delta,P}(E) \equiv \frac{1}{\rho(\mathbf{r})}\,A_{\Delta,P}(E)\,, \tag{5.52}$$

which is largely independent of ρ. Since the product of density and length is the mass thickness (usually given in g·cm^{-2}), the MEA is the mean energy absorbed per unit mass thickness traversed by particles P of energy E. The MEA can be expressed in units of eV·cm^2/g or one of its multiples.

[2]With the ICRU conventions, the term *linear* means *per unit length*.

5.2.1 Monte Carlo calculation of the linear energy absorption

The LEA can be calculated from simple simulations with PENELOPE, or with other Monte Carlo codes, of showers induced by single interactions of parent particles P of energy E in an infinite medium. In these simulations, the value of the absorption energy for electrons should be small enough to ensure that electrons with $E < E_{\mathrm{abs},-}$ radiate a negligible fraction of their energy as bremsstrahlung photons or x rays with energies larger than Δ_γ. We set $E_{\mathrm{abs},-}$ by requiring that the radiative yield (see Section 3.3.2.1) of electrons of that energy is less than 10^{-4}. In PENELOPE a positron that reaches the absorption energy is assumed to deposit its kinetic energy on the spot and then annihilates as if it were at rest. Because the energy distribution of annihilation photons depends on the kinetic energy of the positron, we set $E_{\mathrm{abs},+} = \min\{10^{-3}E, 1\ \mathrm{keV}, E_{\mathrm{abs},-}\}$, to reduce the impact of this simplification in PENELOPE. The absorption energy of photons, $E_{\mathrm{abs},\gamma}$, is set equal to 50 eV, the lowest energy allowed by the program.

The energy transferred in each interaction of particles P with radiant energy E is

$$w = E - E' - Q\,, \tag{5.53}$$

where E' is the energy of the scattered particle, which has to be replaced by zero when the particle is absorbed (*i.e.*, when $E' < \Delta_P$), and Q is the change in the rest energies of the elementary particles involved in the interaction (including the nucleus if interactions such as nuclear photoabsorption are considered). For instance, in the case of electron-positron pair production by photons Q is positive (each event increases the rest energy by $2\mathrm{m_e}c^2$). On the contrary, positron annihilation gives a negative Q (the rest energy decreases by $2\mathrm{m_e}c^2$). When the interaction causes the emission of secondary particles from the target atom, PENELOPE stores them in the secondary stack (see Chapter 7).

5.2.1.1 LEA of photons

The inverse mean free path (or attenuation coefficient) for photons of energy E is

$$\mu_\gamma(E) = \mathcal{N}\sigma_{\gamma,\mathrm{T}}(E), \tag{5.54}$$

where $\mathcal{N}$ is the number of atoms or molecules per unit volume, Eq. (1.102), and

$$\sigma_{\gamma,T}(E) = \sigma_{\mathrm{Ra}}(E) + \sigma_{\mathrm{Co}}(E) + \sigma_{\mathrm{ph}}(E) + \sigma_{\mathrm{pp}}(E) \tag{5.55}$$

is the total cross section (see Chapter 2). We recall that $\mu_\gamma(E)$ is the interaction probability (or the average number of interactions) per unit path length. Therefore, the LEA of photons can be expressed as

$$A_{\Delta,\gamma}(E) = F_\gamma(E) E \mathcal{N}\sigma_{\gamma,\mathrm{T}}(E), \tag{5.56}$$

where $F_\gamma(E)$ is the average fraction of the photon energy that is converted into atomic excitation and sub-cutoff radiation in each interaction.

The quantity $F_\gamma(E)E$ is well suited for calculation through Monte Carlo simulation. To obtain it, we simulate showers that start with a *single interaction* of a parent photon of energy E and score the energy G_i absorbed in each shower. After the interaction of the i-th parent photon, we set $G_i = w$ [see Eq. (5.53)], the energy transferred in the interaction, and proceed to follow the particles P' that are left in the stack. When the interaction of the primary photon is Rayleigh scattering, $w = 0$ and no secondary particles are produced. In the case of Compton scattering, $w = E - E'$ if the energy E' of the scattered photon is larger than Δ_γ, and $w = E$ otherwise. If a particle has energy $E_{P'}$ higher than $\Delta_{P'}$, we update the absorbed-energy counter (*i.e.*, G_i is replaced by $G_i - E_{P'}$), and the simulation of P' is discontinued to avoid double counting, because these particles are assumed to contribute to the fluence distribution $\Phi_{\mathrm{sh},P'}(\mathbf{r}, E')$. However, if P' has sub-cutoff radiant energy but is able to release other particles P'' with energies above the corresponding cutoff, $\Delta_{P''}$, we have to track the particle P', and send to the stack all its descendants P'' with energies higher than $\Delta_{P''}$. After simulating a large number N of showers, the average energy absorbed in interactions of the parent photons is obtained as

$$F_\gamma(E)\,E = \frac{1}{N}\sum_{i=1}^{N} G_i\,. \tag{5.57}$$

5.2.1.2 LEA of electrons and positrons

Electrons and positrons ($P = \pm$) lose energy by way of inelastic collisions ("in") and bremsstrahlung emission ("br"). The total energy-loss DCS is (see Chapter 3)

$$\frac{\mathrm{d}\sigma_{P,\mathrm{T}}(E)}{\mathrm{d}W} = \frac{\mathrm{d}\sigma^P_{\mathrm{in}}(E)}{\mathrm{d}W} + \frac{\mathrm{d}\sigma^P_{\mathrm{br}}(E)}{\mathrm{d}W}\,. \tag{5.58}$$

The LEA for electrons and positrons (and other charged particles) can be expressed as

$$A_{\Delta,P}(E) = S_P(E) - \frac{\mathrm{d}E_{\mathrm{rel}}}{\mathrm{d}s}\,, \tag{5.59}$$

where

$$S_P(E) = \mathcal{N}\int_0^E \frac{\mathrm{d}\sigma_{P,\mathrm{T}}(E)}{\mathrm{d}W}\,W\,\mathrm{d}W \tag{5.60}$$

is the total (collision plus radiative) stopping power, and $\mathrm{d}E_{\mathrm{rel}}/\mathrm{d}s$ is the average energy released as particles P' with energies higher than $\Delta_{P'}$ per unit path length. Since the total cross section for bremsstrahlung emission is infinite (see Section 3.3.2), caution must be exercised when computing $A_{\Delta,P}(E)$ by Monte Carlo simulation. The easiest strategy is to separate the contribution of soft stopping interactions with energy transfers less than the cutoff energy Δ_P, which cause divergence of the total cross section, but give a finite contribution to the stopping power. We write

$$A_{\Delta,P}(E) = \mathcal{N}\int_0^{\Delta_P} \frac{\mathrm{d}\sigma_{P,\mathrm{T}}(E)}{\mathrm{d}W}\,W\,\mathrm{d}W + F_P(E)E\,\rho^{-1}\mathcal{N}\sigma_{P,T}(E)\,, \tag{5.61}$$

where

$$\sigma_{-,\mathrm{T}} = \int_{\Delta_-}^{E} \frac{\mathrm{d}\sigma_{-,\mathrm{T}}(E)}{\mathrm{d}W}\,\mathrm{d}W\,, \tag{5.62a}$$

and

$$\sigma_{+,\mathrm{T}} = \int_{\Delta_+}^{E} \frac{\mathrm{d}\sigma_{+,\mathrm{T}}(E)}{\mathrm{d}W}\,\mathrm{d}W + \sigma_{\mathrm{an}}(E)\,, \tag{5.62b}$$

for electrons and positrons, respectively. The integral on the right-hand sides of Eqs. (5.62) is the total cross section for hard stopping interactions, which is finite, and $\sigma_{\mathrm{an}}(E)$ is the cross section for positron annihilation. The quantity $F_P(E)E$ is the average energy that is effectively absorbed (*i.e.*, deposited in the form of atomic excitations and sub-cutoff radiations) in a hard interaction.

The average energy absorbed in each interaction, $F_P(E)\,E$, can be readily calculated by simulation. As in the case of photons, we generate showers that start with a *single hard interaction* of a primary electron or positron having kinetic energy E. After the interaction of the i-th primary particle, we set $G_i = w$, the energy transferred in the interaction [Eq. (5.53)], discontinue the simulation of the primary particle, and proceed to follow the particles P' in the stack. In the event that after the interaction the energy E of the primary is less than Δ_P, it is added to the energy-absorption counter, *i.e.*, we replace G_i with $G_i + E$. When the energy $E_{P'}$ of a particle P' in the stack is larger than $\Delta_{P'}$, we replace G_i with $G_i - E_{P'}$ and discontinue the simulation of P' to avoid double counting. When P' has sub-cutoff radiant energy but is able to release other particles P'' with energies higher than $\Delta_{P''}$, we track P' and send to the stack all its descendants P'' with energies higher than $\Delta_{P''}$. After simulating a large number N of showers, we have

$$F_P(E)\,E = \frac{1}{N}\sum_{i=1}^{N} G_i\,. \tag{5.63}$$

The program `LEA.f` calculates the LEA for photons, electrons, and positrons using PENELOPE and the simulation schemes described above. The quantity effectively calculated is the average deposited energy, $F_P(E)\,E$, in an interaction (hard interaction in the case of charged particles). For each energy E of the parent particle, the program generates a number N of showers that is large enough to give the LEA value with a statistical uncertainty less than the adopted tolerance, typically 0.1% (1σ level). The computer time needed to calculate the LEA increases with the radiant energy of the particle; the calculation for photons of energies about 10 MeV takes a few minutes on an Intel Core i7 processor.

For electrons and positrons, the program `LEA.f` also calculates the *restricted stopping power*, defined by

$$S_{\Delta,P}(E) \equiv \mathcal{N}\int_0^{\Delta_P} \frac{\mathrm{d}\sigma^P_{\mathrm{in}}(E)}{\mathrm{d}W}\,W\,\mathrm{d}W + \mathcal{N}\int_0^{\Delta_\gamma} \frac{\mathrm{d}\sigma^P_{\mathrm{br}}(E)}{\mathrm{d}W}\,W\,\mathrm{d}W\,, \tag{5.64}$$

which gives the energy loss per unit path length due to inelastic collisions with $W < \Delta_P$ and to the emission of bremsstrahlung photons with $W < \Delta_\gamma$. The LEA is larger

than the restricted stopping power, because in $S_{\Delta,P}(E)$ we exclude the contribution of interactions with energy losses in excess of Δ_P, while in $A_{\Delta,P}(E)$ we exclude only the energies of particles released with $E_P > \Delta_P$ (a part of the transferred energy remains on the spot as atomic excitations and particles with sub-cutoff energies). Figure 5.2 displays the MEA for electrons in water and copper, calculated with the cutoffs $\Delta_\pm = 1$ keV and $\Delta_\gamma = 0$. For comparison, we have also plotted the restricted mass stopping power, $S_{\Delta,-}(E)/\rho$, and the mass stopping power, $S_-(E)/\rho$. The ratio $B_{\Delta,-}/(S_{\Delta,-}/\rho)$ equals unity at low energies and increases monotonically with the energy E of the electron, reaching a saturation value at energies that are larger than both the ionisation energies of the atoms and the cutoff energies Δ_P. The saturation values are 1.015 for water and 1.025 for copper.

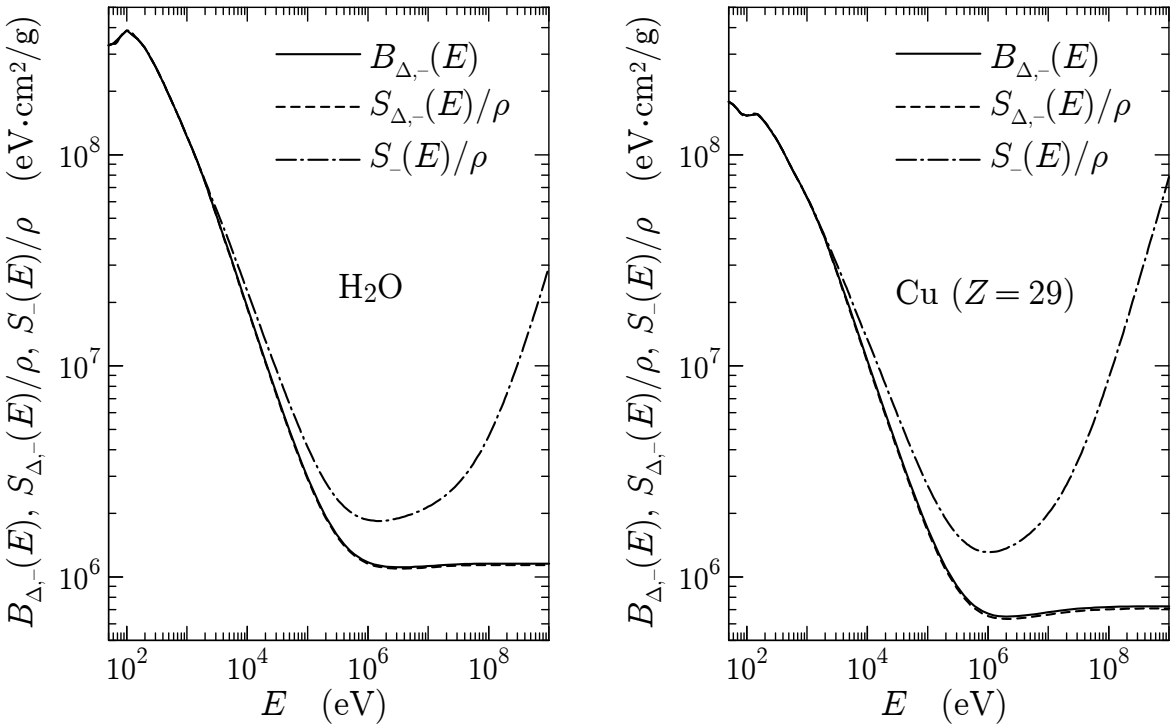

Figure 5.2: Mass energy absorption, $B_{\Delta,-}(E)$, restricted mass stopping power, $S_{\Delta,-}(E)/\rho$, and mass stopping power, $S_-(E)/\rho$, of electrons in water and copper, calculated with $\Delta_- = \Delta_+ = 1$ keV and $\Delta_\gamma = 0$. For projectiles with energies less than about 10 keV, the MEA practically coincides with the restricted mass stopping power.

5.2.2 Absorbed dose

The absorbed dose can be expressed as

$$D(\mathbf{r}) = \frac{1}{\rho(\mathbf{r})} \sum_P \int_{\Delta_P}^{\infty} A_{\Delta,P}(E)\, \Phi_P(\mathbf{r}, E)\, \mathrm{d}E + D_{\mathrm{end}} \tag{5.65}$$

where D_{end} is the contribution of track ends, *i.e.*, of particles with energies E higher than Δ_P that after an interaction emerge with radiant energy below the cutoff, and of source particles with sub-cutoff energies. Since particles with $E < \Delta_P$ are assumed to be absorbed on the spot (*i.e.*, at the first interaction), the contribution of source particles with energies below the cutoff is

$$D_{\text{end, below}} = \frac{1}{\rho(\mathbf{r})} \sum_P \int_0^{\Delta_P} \mathcal{N} \sigma_{P,\text{T}}(E)\, E\, \Phi_{P,\text{sr}}(\mathbf{r}, E)\, \text{d}E\,. \tag{5.66}$$

Particles with energy E higher than the cutoff Δ_P are absorbed after an interaction in which the particle emerges with radiant energy $E' = E - W$ below Δ_P, where, as usual, W is the energy transferred in the interaction. Let $\text{d}\sigma_P(E)/\text{d}W$ denote the energy-transfer DCS, differential only in W, of particles P with radiant energy E. For photons, only Compton scattering needs to be considered, because all other interaction processes either do not alter the photon energy (Rayleigh scattering) or imply the absorption of the incident photon (photoelectric effect and pair production). That is,

$$\frac{\text{d}\sigma_\gamma(E)}{\text{d}W} = \frac{\text{d}\sigma_{\text{Co}}(E)}{\text{d}E'}\,. \tag{5.67}$$

Since the energy W transferred in the interaction where P is absorbed is already counted in the first term of expression (5.65), only the remainder $E - W$ contributes to the track-end term. Therefore, the track-end contribution of particles with energies above their cutoffs is given by

$$D_{\text{end, above}} = \frac{1}{\rho(\mathbf{r})} \sum_P \int_{\Delta_P}^{\infty} \left\{ \int_{E-\Delta_P}^{E} \mathcal{N} \frac{\text{d}\sigma_P(E)}{\text{d}W} (E - W)\, \text{d}W \right\} \Phi_P(\mathbf{r}, E)\, \text{d}E\,. \tag{5.68}$$

Note that particles P' that may be emitted with energies $E > \Delta_{P'}$ as the result of an interaction where P is absorbed, are also accounted for by the first term on the right-hand side of Eq. (5.65).

In simulations where charged particles are tracked using mixed (class-II) algorithms, soft interactions also cause the gradual absorption of particles with energies slightly above the cutoff. To describe the absorption through soft interactions we can utilise the CSDA, because the average effect of these interactions is appropriately represented by the restricted stopping power, $S_{\Delta,P}(E)$ [Eq. (5.64)]. The CSDA can be regarded as the limit of a discrete interaction process in which the particles lose a fixed energy, $W_{\text{s},P}$, at each interaction. The corresponding energy-loss DCS is assumed to be

$$\frac{\text{d}\sigma_{\text{s},P}(E)}{\text{d}W} = \sigma_{\text{s},P}(E)\, \delta(W - W_{\text{s},P})\,, \tag{5.69}$$

where the total cross section $\sigma_{\text{s},P}(E)$ is determined by requiring that the process gives the correct stopping power, *i.e.*,

$$\sigma_{\text{s},P}(E) = \frac{S_{\Delta,P}(E)}{\mathcal{N} W_{\text{s},P}}\,. \tag{5.70}$$

The CSDA is obtained as the limit $W_{\mathrm{s},P} = 0$ of this interaction process. When only this process is active, the track end contribution to the absorbed dose is [cf. Eq. (5.68)]

$$D^{\mathrm{soft}}_{\mathrm{end,\ above}} = \frac{1}{\rho(\mathbf{r})} \sum_P \int_{\Delta_P}^{\infty} \left\{ \int_{E-\Delta_P}^{E} \mathcal{N}\, \sigma_{\mathrm{s},P}(E)\, (E - W)\ \delta(W - W_{\mathrm{s},P})\, \mathrm{d}W \right\} \Phi_P(\mathbf{r}, E)\, \mathrm{d}E.$$

Because the delta distribution implies that we have non-vanishing contributions only when $W_{\mathrm{s},P}$ lies in the interval $(E - \Delta_P, E)$, we can write

$$\begin{aligned} D^{\mathrm{soft}}_{\mathrm{end,\ above}} &= \frac{1}{\rho(\mathbf{r})} \sum_P \int_{\Delta_P}^{\infty} \Big\{ \mathcal{N} \sigma_{\mathrm{s},P}(E)\, (E - W_{\mathrm{s},P}) \\ &\qquad \times \Theta(W_{\mathrm{s},P} - E + \Delta_P)\, \Theta(E - W_{\mathrm{s},P}) \Big\}\, \Phi_P(\mathbf{r}, E)\, \mathrm{d}E \\ &= \frac{1}{\rho(\mathbf{r})} \sum_P \mathcal{N} \int_{\Delta_P}^{\Delta_P + W_{\mathrm{s},P}} \sigma_{\mathrm{s},P}(E)\, (E - W_{\mathrm{s},P})\, \Phi_P(\mathbf{r}, E)\, \mathrm{d}E\,, \end{aligned} \tag{5.71}$$

where $\Theta(x)$ is the unit step function (= 1 if $x > 0$, and 0 otherwise). For small values of $W_{\mathrm{s},P}$, we have

$$\begin{aligned} D^{\mathrm{soft}}_{\mathrm{end,\ above}} &\simeq \frac{1}{\rho(\mathbf{r})} \sum_P \mathcal{N} \sigma_{\mathrm{s},P}(\Delta_P)\, \Delta_P\, \Phi_P(\mathbf{r}, \Delta_P) \int_{\Delta_P}^{\Delta_P + W_{\mathrm{s},P}} \mathrm{d}E \\ &= \frac{1}{\rho(\mathbf{r})} \sum_P \mathcal{N} \sigma_{\mathrm{s},P}(\Delta_P)\, \Delta_P\, \Phi_P(\mathbf{r}, \Delta_P)\, W_{\mathrm{s},P}\,. \end{aligned} \tag{5.72}$$

In the limit $W_{\mathrm{s},P} \to 0$, we obtain the exact CSDA result,

$$D^{\mathrm{soft}}_{\mathrm{end,\ above}} = \frac{1}{\rho(\mathbf{r})} \sum_P \Delta_P\, S_P(\Delta_P)\, \Phi_P(\mathbf{r}, \Delta_P)\,, \tag{5.73}$$

where we have used the identity (5.70), and the fact that at $E = \Delta_P$ the restricted stopping power equals the total stopping power, $S_{\Delta,P}(\Delta_P) = S_P(\Delta_P)$. The result (5.73) implies that the product $S_P(\Delta_P)\, \Phi_P(\mathbf{r}, \Delta_P)$ is the number of particles that reach the energy value Δ_P per unit volume as a result of soft interactions.

Summarising, the contribution of track ends to the absorbed dose is

$$D_{\mathrm{end}} = D_{\mathrm{end,\ above}} + D_{\mathrm{end,\ below}} + D^{\mathrm{soft}}_{\mathrm{end,\ above}}\,, \tag{5.74}$$

where the last term (end-track contribution of soft interactions) is to be included only for charged particles and when soft interactions are described with the CSDA. Recalling

that $\mathcal{N} = N_{\rm A}\rho/A_{\rm M}$, the absorbed dose at $\mathbf{r}$ can be evaluated as

$$\begin{aligned} D(\mathbf{r}) \;=\; & \frac{1}{\rho(\mathbf{r})} \sum_P \int_{\Delta_P}^{\infty} A_{\Delta,P}(E)\, \Phi_P(\mathbf{r},E)\, {\rm d}E \\ & + \frac{N_{\rm A}}{A_{\rm M}} \sum_P \int_0^{\Delta_P} E\, \sigma_{P,{\rm T}}(E)\, \Phi_{P,{\rm sr}}(\mathbf{r},E)\, {\rm d}E \\ & + \frac{N_{\rm A}}{A_{\rm M}} \sum_P \int_{\Delta_P}^{\infty} \left\{ \int_{E-\Delta_P}^{E} (E-W)\, \frac{{\rm d}\sigma_{P,{\rm T}}(E)}{{\rm d}W}\, {\rm d}W \right\} \Phi_P(\mathbf{r},E)\, {\rm d}E \\ & + \frac{1}{\rho(\mathbf{r})} \sum_P \Delta_P\, S_P(E)\, \Phi_P(\mathbf{r},\Delta_P)\,. \end{aligned} \tag{5.75}$$

In calculations of the average absorbed dose in finite volumes, the cutoff energies Δ_P should be selected so that particles leaving and entering the volume of interest with energies less than Δ_P carry, on average, the same total radiant energies. When this condition holds, we say that there is *radiation equilibrium* for particles P with sub-cutoff energies. Only under these circumstances does Eq. (5.75) yield the correct absorbed dose. This is not the case, *e.g.*, in the vicinity of a material-vacuum interface, where there is a net flux of secondary electrons and photons from material to vacuum.

5.2.3 Track length estimator of the absorbed dose

In PENELOPE the absorbed dose is normally obtained by scoring the energy deposited by the interactions within the volume $\mathcal{V}$ of interest. For high-energy photons, this method may require very large simulation times to determine the absorbed dose in thin foils or small volumes, where photon interactions are very rare. To increase the efficiency of the simulation, we may use the variance-reduction technique of interaction forcing (see Section 1.6.1), which is easy to implement, and is nominally exact. An alternative solution is provided by the so-called *track length estimator*, which essentially consists of evaluating the distribution of particle fluence with respect to energy, and using Eq. (5.65) to obtain the absorbed dose. Since every particle that enters the volume contributes to the fluence, the track length estimator may be more efficient than the usual method of scoring the energy deposited in individual interactions.

The track length estimator is implemented as follows. Let M denote the total mass contained in the volume $\mathcal{V}$,

$$M = \int_{\mathcal{V}} \rho(\mathbf{r})\, {\rm d}\mathbf{r}\,. \tag{5.76}$$

When a particle P of radiant energy E travels a path length s within the volume $\mathcal{V}$, we score the quantity $A_{\Delta,P}(E)s$, where $A_{\Delta,P}(E)$ is the LEA. The absorbed dose in $\mathcal{V}$, obtained from the simulation of N complete showers, is [cf. Eq. (5.65)]

$$D = \frac{1}{M\,N} \sum_{i=1}^{N} \left(\sum_j A_{\Delta,P}(E_j)\, s_j \right) + D_{\rm end}\,, \tag{5.77}$$

where the first summation is over the generated showers, and the second is over the steps, j, of the particles of the i-th shower that intersect the volume $\mathcal{V}$. The second term on the right-hand side is the contribution of track ends,

$$D_{\text{end}} = \frac{1}{M\,N} \sum_{k} E_k \,, \tag{5.78}$$

where the summation is over all particles k that reach energies E_k below the corresponding absorption energy $E_{\text{abs},P}$ (*i.e.*, particles that are effectively absorbed) within $\mathcal{V}$. Note that the LEA and the mass density are local quantities; the estimator (5.77) is applicable even when the volume $\mathcal{V}$ contains zones of different compositions or densities. It should be observed that the mass M in these formulas has to be evaluated separately. Usually, simulation codes deliver the product DM.

A practical difficulty of the track length estimator is that it requires the calculation of the LEA prior to the simulation run. To ensure consistency, the cutoffs Δ_P used to determine the LEA must be identical to the absorption energies adopted in the simulation. It should be kept in mind that absorbed doses calculated with the track length estimator are accurate only when there is complete equilibrium of particles with sub-cutoff energies.

5.3 Dosimetry of photon fields

We will now introduce several quantities of practical interest in the dosimetry of photon beams. We consider a photon field described by its distribution of fluence with respect to energy, $\Phi_\gamma(\mathbf{r}, E)$, and we wish to determine the absorbed dose $D(\mathbf{r})$ from the fluence distribution.

The *energy-transfer coefficient*, $\mu_{\text{tr}}(E)$, of photons of energy E is defined as the average fraction of energy released in the form of kinetic energy of charged particles per unit path length of the photons. The charged particles that are set in motion by the interactions of photons are photoelectrons, Compton electrons, electron-positron pairs (and triplets), and Auger electrons. In photoelectric and Compton events (and also in triplet production), the interaction causes the ionisation of the target atom, which is left with a vacancy in the active electron shell. A vacancy in an inner shell migrates to outer shells through a cascade of radiative and non-radiative transitions (see Section 2.6). The total kinetic energy of the released charged particles is assumed to be the difference between the transferred energy w, Eq. (5.53), and the energies of the x rays emitted in the course of the relaxation cascade. Note that, in the case of pair and triplet production, the rest energy of the particles produced, $Q = 2\text{m}_\text{e}c^2$, does not contribute to $\mu_{\text{tr}}(E)$.

The quantities needed to calculate the energy-transfer coefficient are the average total energy X_i of x rays emitted in the relaxation of an atom with a vacancy in the i-th shell, and the average energy $\langle E' \rangle_i$ of the scattered photon in Compton events involving

electrons of the i-th shell. The energy X_i released as characteristic x rays can be readily evaluated from the atomic transition probabilities in the EADL (Perkins *et al.*, 1991). On the other hand,

$$\langle E' \rangle_i = \frac{1}{\sigma_{\mathrm{Co},i}(E)} \int_0^{E-U_i} E' \, \frac{\mathrm{d}\sigma_{\mathrm{Co},i}(E)}{\mathrm{d}E'} \, \mathrm{d}E', \tag{5.79}$$

where U_i is the ionisation energy of the active shell (only energy transfers $W = E - E'$ higher than U_i can cause ionisation), and $\mathrm{d}\sigma_{\mathrm{Co},i}(E)/\mathrm{d}E'$ is the DCS for Compton events with electrons of the active shell, differential only in the energy of the scattered photon. This DCS can be obtained by numerical integration of the DCS (2.28) over directions of the scattered photon,

$$\frac{\mathrm{d}\sigma_{\mathrm{Co},i}(E)}{\mathrm{d}E'} = \int \frac{\mathrm{d}^2\sigma_{\mathrm{Co},i}(E)}{\mathrm{d}E' \, \mathrm{d}\Omega} \, \mathrm{d}\Omega \, . \tag{5.80}$$

The energy-transfer coefficient is given by

$$\begin{aligned} \mu_{\mathrm{tr}} \, = \, \frac{1}{E} \, \frac{N_{\mathrm{A}} \rho}{A_{\mathrm{M}}} \Bigg\{ \sum_i \Big[& \sigma_{\mathrm{ph},i} \, \left(E - X_i\right) + \sigma_{\mathrm{Co},i} \, \left(E - \langle E' \rangle_i - X_i\right) \\ & + \sigma_{\mathrm{triplet},i} \, \left(E - 2\mathrm{m_e}c^2 - X_i\right) \Big] + \sigma_{\mathrm{pair}} \, \left(E - 2\mathrm{m_e}c^2\right) \Bigg\} , \end{aligned} \tag{5.81}$$

where the summation runs over the shells of the target atom. The cross sections $\sigma_{\mathrm{ph},i}$, $\sigma_{\mathrm{Co},i}$, $\sigma_{\mathrm{triplet},i}$, and σ_{pair} correspond, respectively, to photoelectric absorption and Compton scattering by electrons of the i-th shell, triplet production in the field of these electrons, and nuclear-field pair production. It is convenient to express μ_{tr} in terms of the partial attenuation coefficients (see Section 2.5),

$$\mu_{\mathrm{tr}} \, = \, \sum_i \Big[f_{\mathrm{ph},i} \, \mu_{\mathrm{ph},i} + f_{\mathrm{Co},i} \, \mu_{\mathrm{Co},i} + f_{\mathrm{triplet},i} \, \mu_{\mathrm{triplet},i} \Big] + f_{\mathrm{pair}} \, \mu_{\mathrm{pair}} \, , \tag{5.82}$$

where

$$\begin{aligned} f_{\mathrm{ph},i} = 1 - \frac{X_i}{E}, \qquad f_{\mathrm{Co},i} = 1 - \frac{\langle E' \rangle_i + X_i}{E}, \\ f_{\mathrm{triplet},i} = 1 - \frac{2\mathrm{m_e}c^2 + X_i}{E}, \qquad \text{and} \qquad f_{\mathrm{pair}} = 1 - \frac{2\mathrm{m_e}c^2}{E} \end{aligned} \tag{5.83}$$

are the average fractions of the photon energy E that are converted into kinetic energy of charged particles released in the corresponding interactions. The weighted average f of these quantities is defined by the equality

$$\mu_{\mathrm{tr}} = f \, \mu \, , \tag{5.84}$$

where μ is the total attenuation coefficient, $\mu = \mu_{\mathrm{ph}} + \mu_{\mathrm{Ra}} + \mu_{\mathrm{Co}} + \mu_{\mathrm{pp}}$, including the contribution from Rayleigh scattering, which has $f_{\mathrm{Ra}} = 0$. The factor f gives the average

fraction of the photon energy that is released as kinetic energy of charged particles in individual interactions of photons.

To remove the dependence on the density ρ of the material, it is customary to define the *mass energy-transfer coefficient* as

$$\frac{\mu_{\rm tr}}{\rho} = f\,\frac{\mu}{\rho}\,, \tag{5.85}$$

where μ/ρ is the total mass attenuation coefficient, Eq. (2.102). Usually, $\mu_{\rm tr}$ and $\mu_{\rm tr}/\rho$ are given in units of cm^{-1} and cm^2/g, respectively.

The definitions (5.81) to (5.85) apply to elemental materials, and involve only characteristics of the target atom. In the case of homogeneous mixtures and compounds we can adopt the usual additivity approximation. Let us consider a compound material containing elements of atomic numbers Z_j and atomic weights $A_{\rm w}(Z_j)$, with corresponding fractions by weight w_j. If a "molecule" contains n_j atoms of the element Z_j, the molar mass of the compound is $A_{\rm M} = \sum_j n_j A_{\rm w}(Z_j)$ and $n_j = w_j A_{\rm M}/A_{\rm w}(Z_j)$. According to the additivity approximation, the mass energy-transfer coefficient of the compound is

$$\frac{\mu_{\rm tr}}{\rho} = \frac{N_{\rm A}}{A_{\rm M}} \sum_j n_j \left\{ \sum_i \left[\sigma^{(j)}_{{\rm ph},i}\, f^{(j)}_{{\rm ph},i} + \sigma^{(j)}_{{\rm Co},i}\, f^{(j)}_{{\rm Co},i} + \sigma^{(j)}_{{\rm triplet},i}\, f^{(j)}_{{\rm triplet},i} \right] + \sigma^{(j)}_{\rm pair}\, \sigma^{(j)}_{\rm pair} \right\}\,, \tag{5.86}$$

where the superscript (j) refers to the element Z_j. We can also write

$$\frac{\mu_{\rm tr}}{\rho} = \sum_j w_j \frac{\mu^{(j)}_{\rm tr}}{\rho} = \sum_j w_j f^{(j)}\,\frac{\mu^{(j)}}{\rho} = f\,\frac{\mu}{\rho}\,. \tag{5.87}$$

The quantity *kerma*, K, is defined as (and the name is an acronym for) the sum of the kinetic energies of all charged particles released by photons per unit mass (ICRU Report 85, 2011). From the definition of the distribution of fluence with respect to energy $\Phi_\gamma(\mathbf{r}, E)$ [see Eqs. (5.36) and (5.39)], the kerma can be calculated as

$$K(\mathbf{r}) = \int_0^\infty \frac{\mu_{\rm tr}(E)}{\rho}\, E\, \Phi_\gamma(\mathbf{r}, E)\, \mathrm{d}E = \int_0^\infty f\,\frac{\mu(E)}{\rho}\, E\, \Phi_\gamma(\mathbf{r}, E)\, \mathrm{d}E\,. \tag{5.88}$$

The kerma is a good approximation to the absorbed dose when there is complete charged-particle equilibrium *and* the charged particles released by the photons deposit all of their kinetic energies in the material. The latter condition does not hold when a part of the kinetic energy of charged particles is re-emitted in radiative events, *i.e.*, as bremsstrahlung photons and x rays. In this case, we need to account for the fact that this radiation reduces the absorbed dose at $\mathbf{r}$ *and* increases the photon fluence (that is, it contributes to the absorbed dose elsewhere).

To facilitate the direct calculation of the absorbed dose from the fluence distribution, we introduce the *mass energy-absorption coefficient*, defined by

$$\frac{\mu_{\rm en}}{\rho} = (1-g_{\rm ph})\sum_j w_j \sum_i f^{(j)}_{{\rm ph},i}\,\frac{\mu^{(j)}_{{\rm ph},i}}{\rho} + (1-g_{\rm Co})\sum_j w^j \sum_i f^{(j)}_{{\rm Co},i}\,\frac{\mu^{(j)}_{{\rm Co},i}}{\rho}$$
$$+ (1-g_{\rm triplet})\sum_j w^j \sum_i f^{(j)}_{{\rm triplet},i}\,\frac{\mu^{(j)}_{{\rm triplet},i}}{\rho} + (1-g_{\rm pair})\sum_j w_j f^{(j)}_{\rm pair}\,\frac{\mu^{(j)}_{\rm pair}}{\rho}\,. \quad (5.89)$$

The quantities $g_{\rm ph}$, $g_{\rm Co}$, $g_{\rm triplet}$ and $g_{\rm pair}$ are, respectively, the average fractions of the kinetic energy of primary charged particles released in photoelectric, Compton, triplet- and pair-production interactions that are emitted as bremsstrahlung photons and x rays during the slowing down of the charged particles. A weighted average g fraction is defined by

$$\frac{\mu_{\rm en}}{\rho} = (1-g)\,\frac{\mu_{\rm tr}}{\rho} = (1-g)f\,\frac{\mu}{\rho}\,. \quad (5.90)$$

Note that the product $\mu_{\rm en}\,E$ is equal to the LEA, $A_{\Delta,\gamma}(E)$, calculated with the cutoffs $\Delta_\gamma = 0$ and $\Delta_- = \Delta_+ = \infty$.

The absorbed dose at $\mathbf{r}$ can then be evaluated as

$$D(\mathbf{r}) = \int_0^\infty \frac{\mu_{\rm en}(E)}{\rho}\,E\,\Phi_\gamma(\mathbf{r},E)\,{\rm d}E = \int_0^\infty (1-g)f\,\frac{\mu(E)}{\rho}\,E\,\Phi_\gamma(\mathbf{r},E)\,{\rm d}E\,. \quad (5.91)$$

This expression gives the correct absorbed dose provided only that there is complete charged-particle equilibrium. Because the g fractions depend on the slowing down of the primary charged particles, as well as on the production of secondary delta rays, they cannot be deduced from atomic characteristics and have to be evaluated for each material under consideration.

The simplest method to estimate the g factors is to use the CSDA, which leads to relatively simple expressions that can be evaluated by a single numerical quadrature. As an example, let us consider the calculation of the factor $g_{\rm Co}$ for Compton scattering. The distribution of the kinetic energy $E_{\rm e}$ of Compton electrons released in interactions of photons with electrons of the i-th shell is

$$P_{{\rm Co},i}(E_{\rm e}) = \frac{1}{\sigma_{{\rm Co},i}(E)}\,\frac{{\rm d}\sigma_{{\rm Co},i}(E)}{{\rm d}E'}\,, \quad (5.92)$$

where $E' = E - E_{\rm e} - U_i$ is the energy of the scattered photon. The bremsstrahlung yield $Y_{\rm br}(E_{\rm e})$, Eq. (3.167),

$$Y_{\rm br}(E_{\rm e}) = \frac{1}{E_{\rm e}}\int_0^{E_{\rm e}} S_{\rm br}(E'_{\rm e})\,\frac{{\rm d}E'_{\rm e}}{S(E'_{\rm e})}\,, \quad (5.93)$$

is the average fraction of the energy $E_{\rm e}$ of an electron that is emitted as bremsstrahlung photons along the electron trajectory, *i.e.*, from its start with energy E_e until it comes

to rest. In analogy to the bremsstrahlung yield, we introduce the x-ray yield, defined as the average fractional energy emitted as x rays resulting from impact ionisations by a charged particle in the course of its slowing down,

$$Y_{\rm si}(E_{\rm e}) = \frac{1}{E_{\rm e}} \int_0^{E_{\rm e}} \left[\mathcal{N} \sum_j n_j \sum_i \sigma^{(j)}_{{\rm si},k}(E'_{\rm e})\; X^{(j)}_k \right] \frac{{\rm d}E'_{\rm e}}{S(E'_{\rm e})}$$

$$= \frac{1}{E_{\rm e}}\, N_{\rm A}\rho \int_0^{E_{\rm e}} \left[\sum_j \frac{w_j}{A_{\rm m}(Z_j)} \sum_k \sigma^{(j)}_{{\rm si},k}(E'_{\rm e})\; X^{(j)}_k \right] \frac{{\rm d}E'_{\rm e}}{S(E'_{\rm e})}\,, \qquad (5.94)$$

where $\sigma^{(j)}_{{\rm si},k}(E_{\rm e})$ is the cross section for ionisation of the k-th shell of the element Z_j by impact of electrons with kinetic energy $E_{\rm e}$. The average fraction of the energy of Compton electrons that is radiated can be expressed as

$$g_{\rm Co} = \frac{\sum_j w_j \sum_i \sigma^{(j)}_{{\rm Co},i} \int_0^E P^{(j)}_{{\rm Co},i}(E_{\rm e})\, E_{\rm e}\, Y_{\rm tot}(E_{\rm e})\, {\rm d}E_{\rm e}}{\sum_j w_j \sum_i \sigma^{(j)}_{{\rm Co},i} \int_0^E P^{(j)}_{{\rm Co},i}(E_{\rm e})\, E_{\rm e}\, {\rm d}E_{\rm e}}\,, \qquad (5.95)$$

where

$$Y_{\rm tot}(E_{\rm e}) = Y_{\rm br}(E_{\rm e}) + Y_{\rm si}(E_{\rm e}) \qquad (5.96)$$

is the total radiation yield, including bremsstrahlung and x-ray emission.

Seltzer (1993) employed an elaborate scheme to determine mass energy-absorption coefficients including the effects of energy straggling and delta-ray production, which were described using the results from Monte Carlo simulations. Tabulations of μ/ρ and $\mu_{\rm en}/\rho$ evaluated from Seltzer's approach, for the elements from hydrogen to uranium ($Z = 1$ to 92) and for a list of compounds and mixtures, for photon energies between 1 keV and 20 MeV, are available at `www.nist.gov/pml/data/xraycoef/index.cfm`. A practical alternative is to obtain the mass energy-absorption coefficients from simulations with PENELOPE. Although Monte Carlo simulation requires substantial computation times, it has the advantage of including consistently all interactions and is applicable to any material.

The program `LEA.f` can be used to calculate the mass energy-absorption coefficient with PENELOPE. As indicated above, $\mu_{\rm en} = A_{\Delta,\gamma}(E)/E$, when the cutoff energies are $\Delta_\gamma = 0$ and $\Delta_\pm = \infty$. Comparing the definitions (5.56) and (5.90), we see that $F_\gamma = (1-g)f$. However, the program `LEA.F` does not calculate the mass-energy transfer coefficient, $\mu_{\rm tr}$, and the associated fraction f, Eq. (5.85). A similar program, named `mutren.f`, has been written to calculate both $\mu_{\rm tr}$ and $\mu_{\rm en}$. The calculation of $\mu_{\rm en}$ is performed using the same scheme as in `LEA.f`, and f is obtained as the average fraction of the photon energy that is transferred to charged particles in an interaction, in accordance with Eq. (5.90). Figure 5.3 displays mass energy-absorption coefficients and mass attenuation coefficients for photons in liquid water, air, copper and lead calculated with `mutren.f`. The average fractions $(1-g)f$ and f for water and lead (as representatives of low- and high-Z materials) are shown in Fig. 5.4.

The results obtained from our programs are found to differ slightly from those in the NIST tables, because of the following differences between the underlying physical models and assumptions. Firstly, PENELOPE uses DCSs for Compton scattering obtained from the impulse approximation, while the NIST calculations are based on DCSs calculated from the Waller-Hartree theory, which are known to underestimate the energy transfer for photons with energies less than about 100 keV (see Ribberfors, 1983; Brusa *et al.*, 1996). Secondly, the NIST tables were evaluated by considering all of the x rays emitted through the whole atomic relaxation cascade (even those with very small energies), while PENELOPE only follows characteristic x rays emitted from vacancies in K, L, M and N subshells with binding energies higher than 50 eV. Finally, PENELOPE disregards all photons emitted with energies less than the absorption energy $E_{\mathrm{abs},\gamma}$ and, consequently, the results from `mutren.f` are slightly dependent on the value of $E_{\mathrm{abs},\gamma}$. Notice, however, that Eq. (5.91) with the mass energy-absorption coefficients given by `mutren.f` or `LEA.f` should reproduce the absorbed doses obtained from PENELOPE simulations with the absorption energies $E_{\mathrm{abs},P} = \Delta_P$, provided only that charged-particle equilibrium is complete.

5.4 Dosimetry of charged-particle fields

Let us now consider the dosimetry of charged-particle fields, under the assumption that radiative energy losses (bremsstrahlung emission and x-ray production) are negligible. The analysis applies to electrons with moderate kinetic energies and to heavy charged particles (protons, alphas, ...), but not to positrons, because these eventually annihilate with the emission of energetic photons. For the sake of concreteness, we assume an electron field, described by its distribution of fluence with respect to energy, $\Phi_-(\mathbf{r}, E)$. We also consider that there is complete equilibrium of electrons with kinetic energies below a certain cutoff Δ and, accordingly, $\Phi_-(\mathbf{r}, E)$ has to be specified only for $E > \Delta$.

Inelastic interactions of a "primary" electron of energy E may release "secondary" electrons with energies higher than Δ (delta rays), which become part of the electron field. That is, the energies of delta rays are not considered to be directly deposited in the material. Following the recommendations in the ICRU Report 85, we consider the *linear energy transfer* (LET) or *restricted linear electronic stopping power*, L_Δ, of the material, which is defined as the mean energy loss per unit path length due to inelastic collisions of an electron of energy E, excluding the kinetic energies of all the delta rays released by the electron. That is

$$L_\Delta(E) = S_{\mathrm{in}}(E) - \frac{\mathrm{d}E_\Delta}{\mathrm{d}s}\,, \tag{5.97}$$

where $S_{\mathrm{in}}(E) = \langle W \rangle \mathcal{N} \sigma^-_{\mathrm{in}}(E)$, Eq. (3.101), is the collision stopping power, and $\mathrm{d}E_\Delta/\mathrm{d}s$ is the average sum of the kinetic energies of delta rays released per unit path length of the electron. The cutoff Δ is usually expressed in eV, and L_Δ in eV/cm, or some convenient multiple (*e.g.*, MeV/cm). A related quantity is the *restricted collision stopping power*,

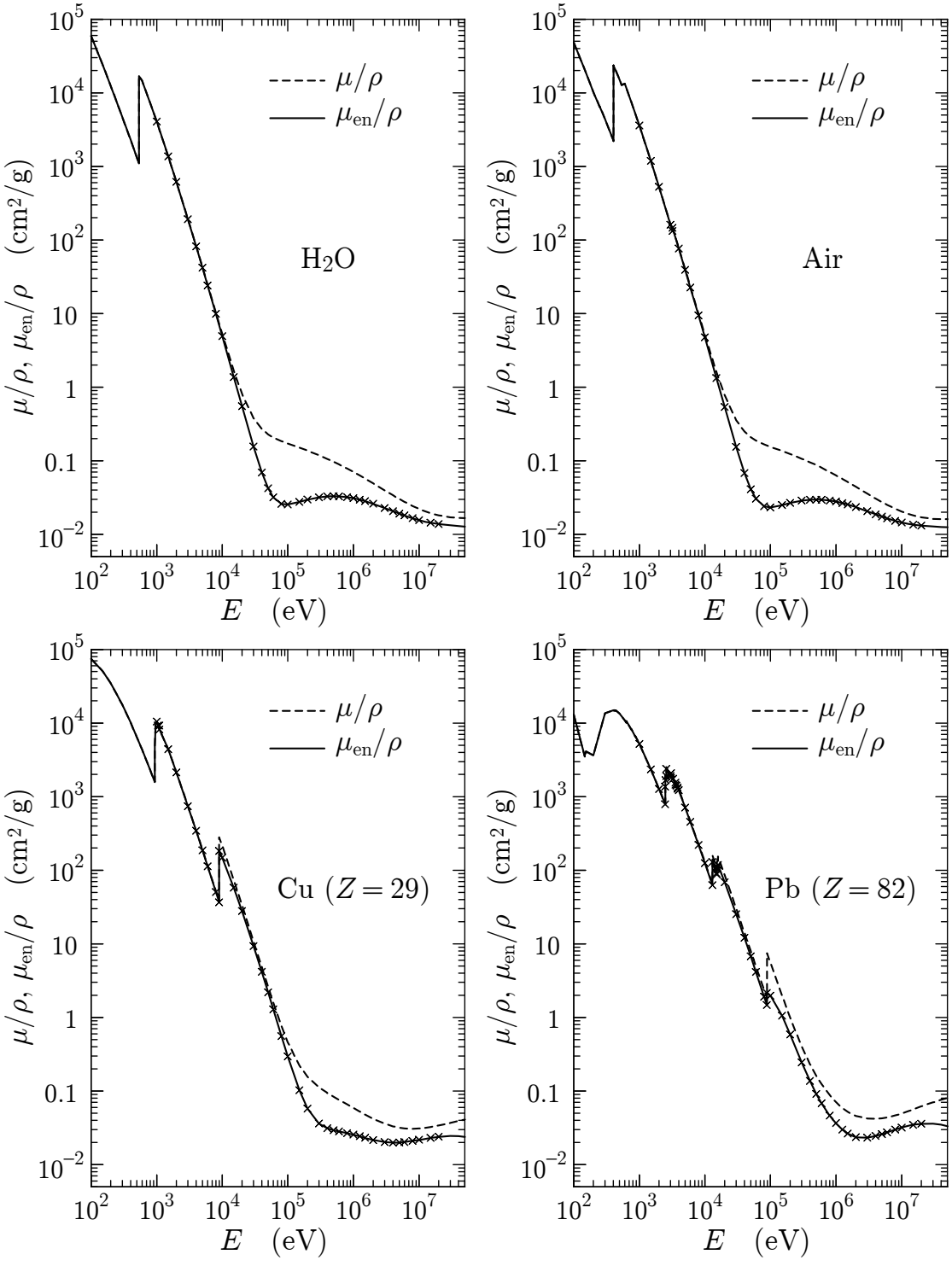

Figure 5.3: Mass energy-absorption coefficients, $\mu_{en}(E)/\rho$, and mass attenuation coefficients, μ/ρ, of water, air, copper and lead as functions of the photon energy. The solid curves represent values of $\mu_{en}(E)$ generated by the program `mutren.f` (*i.e.*, from simulations with PENELOPE) with the cutoff energies $\Delta_\gamma = 0$ and $\Delta_\pm = \infty$. Crosses are values from the NIST database, which were calculated with Seltzer's (1993) method.

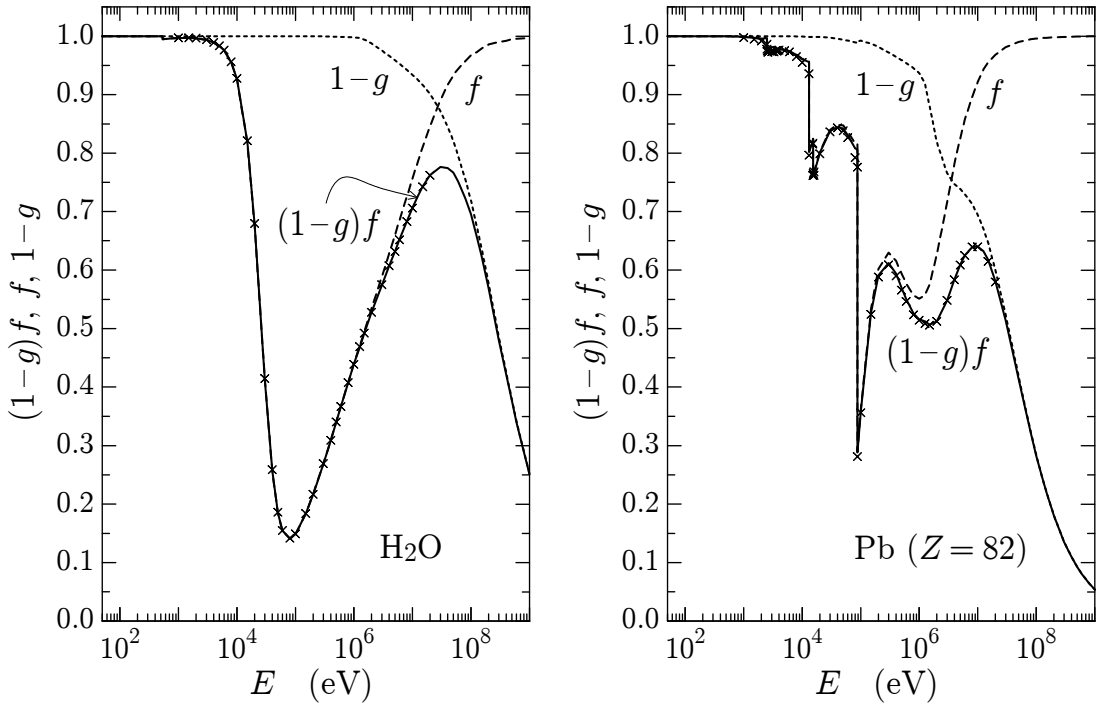

Figure 5.4: Average fraction of energy transferred to charged particles, f, and average fraction of energy absorbed, $(1-g)f$, in photon interactions, as functions of the photon energy. Curves represent values obtained from simulations with PENELOPE. Crosses are values of $(1-g)f$ derived from the NIST database.

defined by

$$S_\Delta(E) = \mathcal{N} \int_0^\Delta W \frac{\mathrm{d}\sigma_{\mathrm{in}}^-(E)}{\mathrm{d}W}\,\mathrm{d}W\,, \tag{5.98}$$

which excludes all energy losses in excess of Δ. Note that $L_\Delta(E)$ is larger than $S_\Delta(E)$, because the sum of kinetic energies of delta rays is smaller than the energy W transferred in the interaction, which is partially spent in overcoming the binding energy of the electrons in the target atom (after the interaction, the atom remains in an excited state). The difference between $L_\Delta(E)$ and $S_\Delta(E)$ is small for large values of Δ, but it becomes significant when Δ is comparable to the ionisation energies of the atom. Note, however, that $L_\Delta(\Delta) = S_{\mathrm{in}}(\Delta)$, *i.e.*, the two quantities coincide for $E = \Delta$. As the LET excludes the actual energies released as delta rays, it is more meaningful than the restricted stopping power.

In general, the LET differs from the LEA, because the LEA accounts for energies released by secondary photons, which are not considered in the definition of the LET. The two quantities coincide only when the total radiative yield (bremsstrahlung plus x rays) is negligible. When this is the case, the LET for electrons can be calculated by running the program `LEA.f` with $\Delta_- = \Delta$ and $\Delta_\gamma = \Delta_+ = 0$.

The quantity *cema*, C, is defined as (and the name is an acronym for) converted energy per unit mass. It is given by

$$C(\mathbf{r}) = \int_0^\infty \frac{S_{\rm in}(E)}{\rho}\, \Phi_{\rm p}(\mathbf{r}, E)\, {\rm d}E\,, \tag{5.99}$$

where $\Phi_{\rm p}(\mathbf{r}, E)$ is the distribution of fluence with respect to energy of primary electrons. That is, the cema disregards the energy transported by secondary electrons.

A more practical quantity is the *restricted cema*, C_Δ, which is the energy deposited by electrons (or other charged particles) per unit mass, excluding the kinetic energies of delta rays. Since electrons with sub-cutoff energies are assumed to be absorbed on the spot, the definition of restricted cema includes a track-end term (Kellerer *et al.*, 1992),

$$C_\Delta(\mathbf{r}) = \int_\Delta^\infty \frac{L_\Delta(E)}{\rho}\, \Phi_-(\mathbf{r}, E)\, {\rm d}E + C_{\rm end}\,. \tag{5.100}$$

From arguments similar to those leading to Eq. (5.75), we have

$$\begin{aligned} C_{\rm end} \;=\; & \frac{N_{\rm A}}{A_{\rm M}} \int_\Delta^\infty \left\{ \int_{E-\Delta}^{E} (E - W)\, \frac{{\rm d}\sigma_{\rm in}^-(E)}{{\rm d}W}\, {\rm d}W \right\} \Phi_-(\mathbf{r}, E)\, {\rm d}E \\ & + \frac{N_{\rm A}}{A_{\rm M}}\, \Delta\, S_{\rm in}(\Delta)\, \Phi_-(\mathbf{r}, \Delta)\,. \end{aligned} \tag{5.101}$$

This result is more general than the expression given by Kellerer *et al.*, which was derived from the CSDA.

The cema can be considered as the limit of the restricted cema when *all* secondary electrons are assumed to be absorbed on the spot. Under these circumstances, $L_\Delta(E) = S_{\rm in}(E)$, and $\Phi_-(\mathbf{r}, E)$ reduces to the distribution of fluence of primary electrons, $\Phi_{\rm p}(\mathbf{r}, t)$. The restricted cema approaches the absorbed dose from electron fields when sub-cutoff electron equilibrium exists and radiative energy losses are negligible. Evidently, the restricted cema provides a better approximation to the absorbed dose than the cema, which assumes equilibrium of secondary electrons irrespective of their energies.

Chapter 6

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, *etc.*). 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 track transported particles. 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 PENGEOM, 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, 1988a,b,c), ITS3 (Halbleib *et al.*, 1992), EGS4

(Nelson *et al.*, 1985), GEANT3 (Brun *et al.*, 1987), 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.*, 2006)] have recourse to a combination of condensed (class I) and mixed (class II) simulation schemes, 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 (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 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 referenced above.

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 2014 of PENGEOM is able to simulate very complex material systems, with up to 5,000 bodies and 10,000 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 geometrical 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. Almansa *et al.* (2016) improved the robustness of the tracking subroutines by introducing fuzzy quadric surfaces (see also Salvat, 2015). Almansa *et al.* have developed a Java application named PENGEOM.JAR which allows the definition of the geometry and its visualisation using the same tracking algorithms as in the Windows viewers. PENGEOM.JAR is not part of the PENELOPE distribution package.

In what follows, and in the computer programs, lengths are assumed to be given in cm. Notice, however, that PENGEOM operates with arbitrary units; the subroutines

perform correctly provided only that the input step lengths and the geometry parameters are expressed in the same units. The position and direction of movement of a particle are referred to the laboratory coordinate system, a right-handed 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.

6.1 Surfaces and their transformations

Given a function $F(\mathbf{r})$, assumed to be continuous and differentiable, the equation $F(\mathbf{r}) = 0$ defines a surface in implicit form. A surface divides the space into two exclusive regions that are identified by the sign of $F(\mathbf{r})$, the *surface side pointer*, SP. A point with coordinates $\mathbf{r} = (x, y, z)$ is said to be inside the surface if $F(\mathbf{r}) \leq 0$ (SP $= -1$), and outside it if $F(\mathbf{r}) > 0$ (SP $= +1$). The surface itself [*i.e.*, the set of points such that $F(\mathbf{r}) = 0$] is the boundary of the two regions. Note that the equation $-F(\mathbf{r}) = 0$ defines the same surface, but with the inside and outside interchanged. Consequently, one must be careful with the global sign of the surface function $F(\mathbf{r})$. Let $\mathbf{r}_0$ be a point on the surface [*i.e.*, $F(\mathbf{r}_0) = 0$]. The gradient of the function $F(\mathbf{r})$ at $\mathbf{r}_0$,

$$\nabla F(\mathbf{r}_0) = \left(\frac{\partial F}{\partial x}, \frac{\partial F}{\partial y}, \frac{\partial F}{\partial z} \right)_{\mathbf{r}=\mathbf{r}_0}, \tag{6.1}$$

is a vector perpendicular to the surface at $\mathbf{r}_0$ that points towards the outside.

To facilitate the definition of the material system and tracking of particles we will use simple surfaces, and occasionally transform them by means of scalings along the directions of the coordinate axes, translations, and rotations. To describe these transformations, we shall adopt the active point of view: the reference frame remains fixed and only the space points (or their position vectors) are scaled, translated or rotated.

A scaling transformation $\mathcal{S}(\boldsymbol{\alpha})$, with scaling factors $\boldsymbol{\alpha} = (\alpha_x, \alpha_y, \alpha_z)$, transforms the vector $\mathbf{r} = (x, y, z)$ into a vector

$$\mathbf{r}' = \mathcal{S}(\boldsymbol{\alpha})\, \mathbf{r} = (\alpha_x x, \alpha_y y, \alpha_z z) = (x', y', z'), \tag{6.2}$$

or, in matrix form,

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} \alpha_x & 0 & 0 \\ 0 & \alpha_y & 0 \\ 0 & 0 & \alpha_z \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}, \tag{6.3}$$

where the vectors $\mathbf{r}$ and $\mathbf{r}'$ are considered as one-column matrices. The inverse of this transformation is the scaling with the reciprocal factors,

$$\mathcal{S}^{-1}(\boldsymbol{\alpha}) = \mathcal{S}(1/\alpha_x, 1/\alpha_y, 1/\alpha_z). \tag{6.4}$$

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

$$\mathcal{T}(\mathbf{t})\, \mathbf{r} = \mathbf{r} + \mathbf{t} = (x + t_x, y + t_y, z + t_z). \tag{6.5}$$

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 axes[1]: 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 6.1 displays the effect of a rotation, defined by its Euler angles, on an asymmetric object, an arrow pointing initially along the z axis with an arm 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 arm. It is clear that, by selecting appropriate values of the Euler angles, we can set the object in any given orientation, that is, we can describe any rotation. Since a rotation of angle 2π represents a complete turn, the rotation remains the same when we add an integer multiple of 2π to either of the Euler angles. Hence, different sets of Euler angles, all of them differing by integer multiples of 2π, represent the same transformation. In formal studies of the rotation group, it is common to associate a rotation with a single set of Euler angles, whose values are restricted to their "natural" intervals: $\omega \in (0, 2\pi)$, $\theta \in (0, \pi)$ and $\phi \in (0, 2\pi)$.

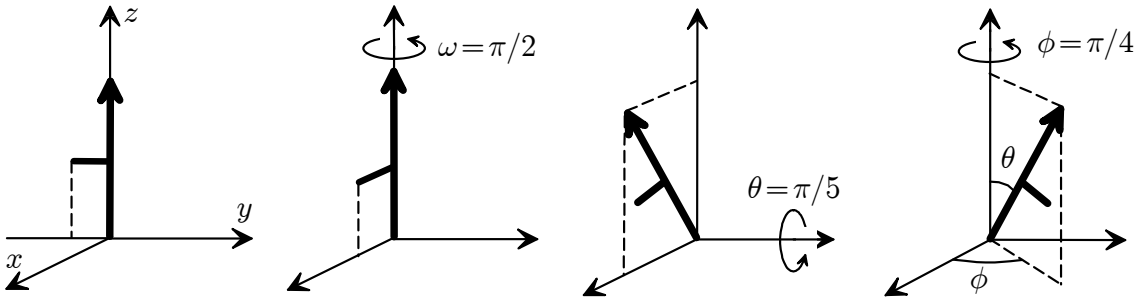

Figure 6.1: Rotation of an asymmetric object, an arrow with an arm, as defined by the Euler angles ω, θ, and ϕ. The orientation of the axes is the same in all the diagrams.

The rotation $\mathcal{R}(\omega, \theta, \phi)$ transforms the vector $\mathbf{r} = (x, y, z)$ into a vector

$$\mathbf{r}' = \mathcal{R}(\omega, \theta, \phi)\, \mathbf{r} = (x', y', z'), \tag{6.6}$$

whose components are given by

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \mathcal{R}(\omega, \theta, \phi) \begin{pmatrix} x \\ y \\ z \end{pmatrix}, \tag{6.7}$$

[1]This definition of the Euler angles is the one usually adopted in quantum mechanics (Edmonds, 1960).

where $\mathcal{R}(\omega,\theta,\phi)$ is the rotation matrix. To obtain the explicit form of this matrix, we recall that rotations of angle ϕ about the z axis and of angle θ about the y axis are described by the matrices

$$\mathcal{R}(\phi\hat{\mathbf{z}}) = \begin{pmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \text{and} \qquad \mathcal{R}(\theta\hat{\mathbf{y}}) = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix}, \tag{6.8}$$

respectively. Hence,

$$\begin{aligned} \mathcal{R}(\omega,\theta,\phi) &= \mathcal{R}(\phi\hat{\mathbf{z}})\,\mathcal{R}(\theta\hat{\mathbf{y}})\,\mathcal{R}(\omega\hat{\mathbf{z}}) \\ &= \begin{pmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} \cos\omega & -\sin\omega & 0 \\ \sin\omega & \cos\omega & 0 \\ 0 & 0 & 1 \end{pmatrix} = \end{aligned}$$

$$\begin{pmatrix} \cos\phi\cos\theta\cos\omega - \sin\phi\sin\omega & -\cos\phi\cos\theta\sin\omega - \sin\phi\cos\omega & \cos\phi\sin\theta \\ \sin\phi\cos\theta\cos\omega + \cos\phi\sin\omega & -\sin\phi\cos\theta\sin\omega + \cos\phi\cos\omega & \sin\phi\sin\theta \\ -\sin\theta\cos\omega & \sin\theta\sin\omega & \cos\theta \end{pmatrix}. \tag{6.9}$$

The inverse of the rotation $\mathcal{R}(\omega,\theta,\phi)$ is $\mathcal{R}(-\phi,-\theta,-\omega)$ and, because rotation matrices are orthogonal, its matrix is the transpose of $\mathcal{R}(\omega,\theta,\phi)$, *i.e.*,

$$\mathcal{R}^{-1}(\omega,\theta,\phi) = \mathcal{R}(-\phi,-\theta,-\omega) = \mathcal{R}(-\omega\hat{\mathbf{z}})\,\mathcal{R}(-\theta\hat{\mathbf{y}})\,\mathcal{R}(-\phi\hat{\mathbf{z}}) = \mathcal{R}^{\mathrm{T}}(\omega,\theta,\phi). \tag{6.10}$$

Let us now consider transformations $\mathcal{C} = \mathcal{T}(\mathbf{t})\,\mathcal{R}(\omega,\theta,\phi)\,\mathcal{S}(\boldsymbol{\alpha})$ that are products of a scaling $\mathcal{S}(\boldsymbol{\alpha})$, a rotation $\mathcal{R}(\omega,\theta,\phi)$ and a translation $\mathcal{T}(\mathbf{t})$. $\mathcal{C}$ transforms a point $\mathbf{r}$ into

$$\mathbf{r}' = \mathcal{C}\mathbf{r} = \mathcal{T}(\mathbf{t})\,\mathcal{R}(\omega,\theta,\phi)\,\mathcal{S}(\boldsymbol{\alpha})\,\mathbf{r} \tag{6.11}$$

or, in matrix form,

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \mathcal{R}(\omega,\theta,\phi) \begin{pmatrix} \alpha_x & 0 & 0 \\ 0 & \alpha_y & 0 \\ 0 & 0 & \alpha_z \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}. \tag{6.12}$$

Notice that the order of the factors does matter; the product of the same factors in reverse order $\mathcal{D} = \mathcal{S}(\boldsymbol{\alpha})\,\mathcal{R}(\omega,\theta,\phi)\,\mathcal{T}(\mathbf{t})$ transforms $\mathbf{r}$ into a point $\mathbf{r}'' = \mathcal{D}\mathbf{r}$ with coordinates

$$\begin{pmatrix} x'' \\ y'' \\ z'' \end{pmatrix} = \begin{pmatrix} \alpha_x & 0 & 0 \\ 0 & \alpha_y & 0 \\ 0 & 0 & \alpha_z \end{pmatrix} \mathcal{R}(\omega,\theta,\phi) \begin{pmatrix} x + t_x \\ y + t_y \\ z + t_z \end{pmatrix}. \tag{6.13}$$

Evidently, in general, $\mathbf{r}'' \neq \mathbf{r}'$.

Given a surface $F(\mathbf{r}) = 0$, we can generate a new surface by applying a scaling $\mathcal{S}(\boldsymbol{\alpha})$ followed by a rotation $\mathcal{R}(\omega, \theta, \phi)$ and a translation $\mathcal{T}(\mathbf{t})$ (we shall always adopt this order). The implicit equation of the transformed surface is

$$G(\mathbf{r}) = F\left[\mathcal{S}^{-1}(\boldsymbol{\alpha})\,\mathcal{R}^{-1}(\omega, \theta, \phi)\,\mathcal{T}^{-1}(\mathbf{t})\,\mathbf{r}\right] = 0, \tag{6.14}$$

which simply expresses the fact that $G(\mathbf{r})$ equals the value of the original function at the point $\mathbf{r}' = \mathcal{S}^{-1}(\boldsymbol{\alpha})\,\mathcal{R}^{-1}(\omega, \theta, \phi)\,\mathcal{T}^{-1}(\mathbf{t})\,\mathbf{r}$ that transforms into $\mathbf{r}$ [*i.e.*, $\mathbf{r} = \mathcal{T}(\mathbf{t})\,\mathcal{R}(\omega, \theta, \phi)\,\mathcal{S}(\boldsymbol{\alpha})\mathbf{r}'$]. As an example, Fig. 6.2 shows schematically the transformation of a sphere into an asymmetric ellipsoid.

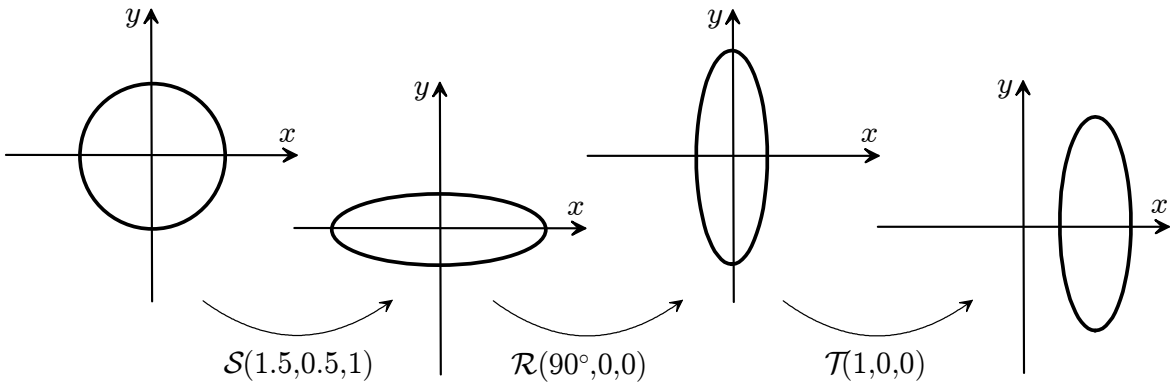

Figure 6.2: Example of a surface transformation: a sphere, $F(\mathbf{r}) = x^2 + y^2 + z^2 - 1 = 0$, is transformed into an ellipsoid. Note the order in which scaling, rotation and translation are applied. Of course, the transformation that leads to the final surface is not unique: a scaling $\mathcal{S}(0.5, 1.5, 1)$ followed by a translation $\mathcal{T}(1, 0, 0)$ would have the same effect.

6.2 Constructive quadric geometry and ray tracing

The material systems considered in PENGEOM consist of a number of homogeneous bodies B_k determined by their limiting surfaces S_i and compositions (materials). Each surface S_i is specified by giving its equation $F_i(\mathbf{r}) = 0$, and the volume of elementary bodies can be defined simply by giving the side pointers SP_i of all the surfaces S_i that limit that volume. However, this is not always possible or convenient for the user. It is more expedient to assume that bodies are defined in "ascending" exclusive order so that previously defined bodies effectively delimit the new ones; this implies that overlaps between different bodies are not permitted. Figure 6.3 shows a simple geometry consisting of 2 bodies B_k ($k = 1, 2$), a sphere with a cylindrical inset, limited by four surfaces S_i ($i = 1$ to 4). In this case, the cylinder is defined first (as the volume limited by surfaces 1, 2, and 3) so that it automatically limits the volume of the hollow sphere (limited by surface 4, and the cylinder). It is impossible to define the hollow sphere as a single body by means of only its limiting surfaces because the surfaces 1 and 2 of the cylinder do not limit body 2 (the hollow sphere) in all their extensions.

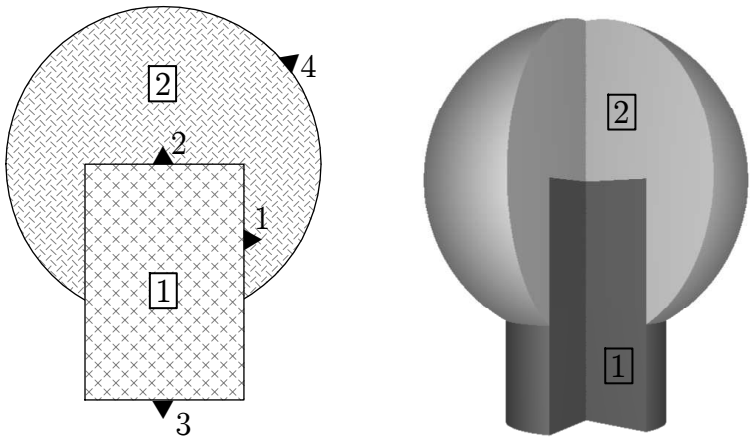

Figure 6.3: Left: Schematic representation of a simple geometry, a sphere with a cylindrical inset. The solid triangles indicate the outside of the surfaces (SP = +1); number labels in squares indicate bodies. Right: Three-dimensional view with an excluded wedge to show the inner structure. (From Salvat, 2015).

The "location" of a particle is specified by giving *its position coordinates* $\mathbf{r}$ *and the body* B_k where it is moving. Given the initial position of a particle, $\mathbf{r}_0$, to determine the body that contains it we should calculate the SPs of all limiting surfaces and explore the set of bodies in ascending order to find the one with the right SPs. For complex systems, with a large number of limiting surfaces, this blind search may be quite lengthy. The grouping of bodies into modules (see Section 6.2.3) serves to reduce the number of surfaces that need to be considered to locate a point.

A particle trajectory is simulated as a sequence of connected straight free flights, each of which ends with an interaction of the particle or with a hinge. When the particle reaches an interface (*i.e.*, a surface limiting two bodies of different compositions), the simulation has to be halted and restarted with the interaction cross sections of the medium beyond the interface. The most basic geometry operation, which is to be performed millions of times in the course of a single simulation, is the calculation of intersections of particle track segments with limiting surfaces. Assume that a particle starts a free flight of length s_0 from a point $\mathbf{r}_0$ in body B_k moving in the direction $\hat{\mathbf{d}}$. We wish to determine whether the track segment intersects any of the surfaces $F(\mathbf{r}) = 0$ which limit the body where the particle is moving. The intersections of the ray $\mathbf{r}_0 + s\hat{\mathbf{d}}$ with the surface occur at distances s from $\mathbf{r}_0$ that are solutions of the following "master" equation

$$f(s) \equiv F(\mathbf{r}_0 + s\hat{\mathbf{d}}) = 0, \tag{6.15}$$

where $f(s)$ is the value of the surface function along the ray. In the course of a free flight of length s_0, the particle will cross the surface only if this equation has a root s such that $0 < s \leq s_0$. Note that the SP of a surface at the particle position changes when the particle crosses the surface. By keeping track of the SPs of the surfaces at the current position of the particle, the body where the particle is moving is identified unambiguously.

6.2.1 Quadric surfaces

To simplify the calculation of interface crossings, it is convenient to use surfaces expressed by simple analytical functions such that the master equation (6.15) can be solved analytically. In the PENGEOM package, all limiting surfaces are assumed to be quadrics given by the implicit equation

$$\begin{aligned} F(\mathbf{r}) \;=\; & A_{\rm xx}x^2 + A_{\rm xy}xy + A_{\rm xz}xz + A_{\rm yy}y^2 + A_{\rm yz}yz + A_{\rm zz}z^2 \\ & + A_{\rm x}x + A_{\rm y}y + A_{\rm z}z + A_0 = 0, \end{aligned} \tag{6.16}$$

which includes planes, spheres, cylinders, cones, ellipsoids, paraboloids, hyperboloids, *etc.* It is useful to express the generic quadric equation (6.16) in matrix form,

$$F(\mathbf{r}) = \mathbf{r}^{\rm T}\mathcal{A}\,\mathbf{r} + \mathbf{A}^{\rm T}\mathbf{r} + A_0 = 0, \tag{6.17}$$

where

$$\mathcal{A} \equiv \begin{pmatrix} A_{\rm xx} & \frac{1}{2}A_{\rm xy} & \frac{1}{2}A_{\rm xz} \\ \frac{1}{2}A_{\rm xy} & A_{\rm yy} & \frac{1}{2}A_{\rm yz} \\ \frac{1}{2}A_{\rm xz} & \frac{1}{2}A_{\rm yz} & A_{\rm zz} \end{pmatrix} \tag{6.18}$$

is a symmetric matrix. As usual, vectors such as $\mathbf{r}$ and $\mathbf{A} \equiv (A_{\rm x}, A_{\rm y}, A_{\rm z})$ are considered here as one-column matrices. $\mathbf{r}^{\rm T}$ is a one-row matrix, and $\mathbf{A}^{\rm T}\mathbf{r}$ is the familiar dot product $\mathbf{A}\cdot\mathbf{r}$. The gradient of the quadric surface function is the vector

$$\nabla F(\mathbf{r}) = 2\mathcal{A}\mathbf{r} + \mathbf{A}\,. \tag{6.19}$$

The advantage of using quadric surfaces is that Eq. (6.15) for the ray intersections is quadratic,

$$f(s) = as^2 + bs + c = 0 \tag{6.20}$$

with

$$a = \hat{\mathbf{d}}^{\rm T}\mathcal{A}\hat{\mathbf{d}}\,, \qquad b = 2\hat{\mathbf{d}}^{\rm T}\mathcal{A}\mathbf{r}_0 + \hat{\mathbf{d}}^{\rm T}\mathbf{A} = \hat{\mathbf{d}}^{\rm T}\nabla F(\mathbf{r_0})\,, \qquad c = F(\mathbf{r_0})\,, \tag{6.21}$$

and its roots are

$$s = \frac{-b \pm \sqrt{\Delta}}{2a} \qquad \text{with} \qquad \Delta \equiv b^2 - 4ac. \tag{6.22}$$

If the discriminant Δ is positive, there are two real roots and the ray intersects the surface twice; if $\Delta = 0$, there is a real root of multiplicity two and the ray grazes the surface; finally, if Δ is negative, there are no real roots and the ray misses the surface. When $\mathcal{A} = 0$, the surface is a plane and there is only one root, $s = -c/b$. Of course, we need to consider only intersections ahead of the ray, which correspond to $s > 0$.

Quadric surfaces are flexible enough to model many man-made material structures, and to approximate more general surfaces. It is worth mentioning that surfaces defined by cubic or four-degree polynomials in x, y and z also allow the analytical calculation

of ray intersections because the associated function $f(s)$ is a cubic or fourth-degree polynomial in s. However, the calculation is more complicated than for quadric surfaces. Since lengthy geometrical operations may severely impair the efficiency of Monte Carlo simulation, only quadric surfaces are considered in PENGEOM.

Although Eq. (6.16) contains 10 parameters, a translation to a central position, followed by a rotation to a standard orientation and a scaling to normalise the remaining parameters, transform a real quadric surface into one of 10 possible reduced forms. These are given by the expression

$$F_{\rm r}(\mathbf{r}) = I_1 x^2 + I_2 y^2 + I_3 z^2 + I_4 z + I_5 = 0, \tag{6.23}$$

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.*, $F_{\rm r}(-x,-y,z) = F_{\rm r}(x,y,z)$. The real reduced quadrics are given in Table 6.1 and are displayed in Fig. 6.4. We do not consider imaginary surfaces (*e.g.*, the imaginary ellipsoid $x^2+y^2+z^2+1=0$), which are of no interest here. Additionally, we have omitted quadrics that represent two planes in various orientations (such as crossing planes $x^2 - y^2 = 0$, parallel planes $z^2 - 1 = 0$, and double planes $z^2 = 0$) because they are more conveniently defined by using single planes, which do not require taking square roots to calculate ray crossings.

In practice, limiting surfaces are frequently described in "geometrical" form and it may be inconvenient to obtain the corresponding quadric parameters. Let us try with a simple example: obtain the parameters of a circular cylinder of radius R such that its symmetry axis goes through the origin and it is parallel to the vector $\mathbf{u} = (1,1,1)$. 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. 6.4), and the set of geometrical transformations that produce the desired surface. Starting from the corresponding reduced form, a quadric is obtained by applying the following transformations (in the quoted order).

(i) A scaling along the directions of the axes, defined by the scaling factors[2] `X-SCALE`$=\alpha_x$, `Y-SCALE`$=\alpha_y$ and `Z-SCALE`$=\alpha_z$. The equation of the scaled quadric is

$$F_{\rm s}(\mathbf{r}) = I_1 \left(\frac{x}{\alpha_x}\right)^2 + I_2 \left(\frac{y}{\alpha_y}\right)^2 + I_3 \left(\frac{z}{\alpha_z}\right)^2 + I_4 \frac{z}{\alpha_z} + I_5 = 0. \tag{6.24}$$

For instance, this transforms the reduced sphere into an ellipsoid with semi-axes equal to the scaling factors.

(ii) A rotation, $\mathcal{R}(\omega,\theta,\phi)$, defined through the Euler angles `OMEGA`$=\omega$, `THETA`$=\theta$ and `PHI`$=\phi$. 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.

[2]Keywords used to denote the various parameters in the geometry definition file are written in typewriter font, *e.g.*, `X-SCALE`). See Section 6.3.

(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).

Thus, a general quadric surface 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 real quadric surface can be expressed in this way. The cylinder mentioned in the above example is defined by

$$
\begin{aligned}
&I_1 = 1,\ \ I_2 = 1,\ \ I_3 = 0,\ \ I_4 = 0,\ \ I_5 = -1,\\
&\texttt{X-SCALE} = \texttt{Y-SCALE} = R,\ \ \texttt{Z-SCALE} = 1,\\
&\texttt{OMEGA} = 0,\ \ \texttt{THETA} = \arccos(1/\sqrt{3}) = 54.73 \text{ deg},\ \ \texttt{PHI} = \arccos(1/\sqrt{2}) = 45 \text{ deg},\\
&\texttt{X-SHIFT} = \texttt{Y-SHIFT} = \texttt{Z-SHIFT} = 0,
\end{aligned}
$$

where the rotation angles θ and ϕ are determined from the relations $\cos\theta = \hat{\mathbf{z}}\cdot\mathbf{u}/|\mathbf{u}|$ and $\cos\phi\,\sin\theta = \hat{\mathbf{x}}\cdot\mathbf{u}/|\mathbf{u}|$ with $\mathbf{u} = (1,1,1)$.

Table 6.1: 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 and y.						

The implicit equation of the quadric is obtained as follows. The equation (6.24) of the scaled quadric can be written in matrix form [cf. Eq. (6.17)],

$$\mathbf{r}^{\mathrm{T}}\mathcal{A}\,\mathbf{r} + \mathbf{A}^{\mathrm{T}}\mathbf{r} + A_0 = 0, \tag{6.25}$$

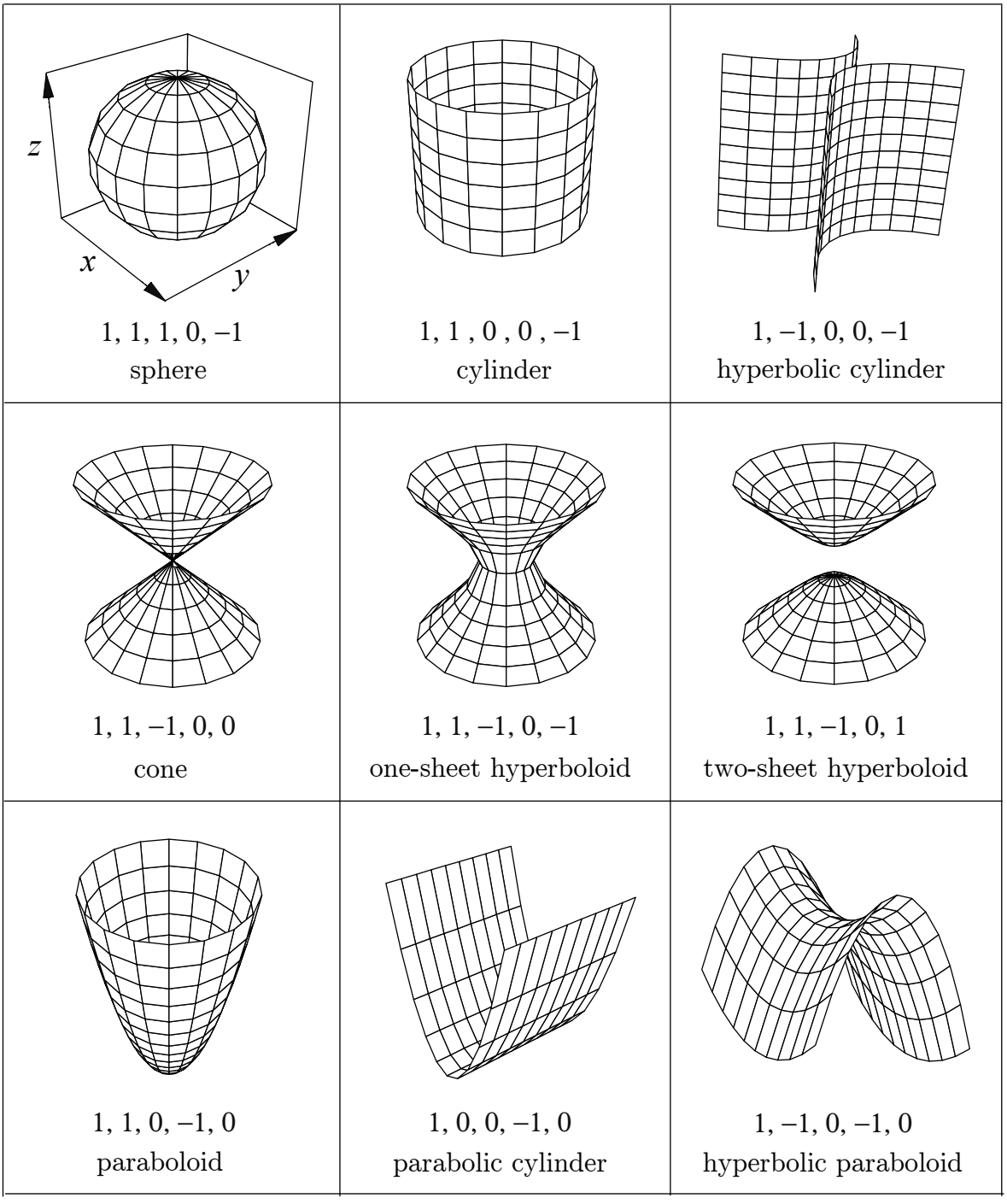

Figure 6.4: Non-planar reduced quadric surfaces and their indices [see Eq. (6.23)]. In all cases, the perspective is the same as for the sphere.

with

$$\mathcal{A} = \begin{pmatrix} I_1/\alpha_x^2 & 0 & 0 \\ 0 & I_2/\alpha_x^2 & 0 \\ 0 & 0 & I_3/\alpha_z^2 \end{pmatrix} \qquad \mathbf{A} = (0, 0, I_4/\alpha_z)\ , \qquad A_0 = I_5\,. \tag{6.26}$$

The equation for the rotated and shifted quadric is [see Eq. (6.14)]

$$(\mathbf{r} - \mathbf{t})^{\mathrm{T}} \mathcal{R}\mathcal{A}\mathcal{R}^{\mathrm{T}}\,(\mathbf{r} - \mathbf{t}) + (\mathcal{R}\mathbf{A})^{\mathrm{T}}(\mathbf{r} - \mathbf{t}) + A_0 = 0, \tag{6.27}$$

which can be written in the generic form (6.17)

$$\mathbf{r}^{\mathrm{T}} \hat{\mathcal{A}}\,\mathbf{r} + \hat{\mathbf{A}}^{\mathrm{T}}\mathbf{r} + \hat{A}_0 = 0 \tag{6.28}$$

with

$$\hat{\mathcal{A}} = \mathcal{R}\mathcal{A}\mathcal{R}^{\mathrm{T}}, \qquad \hat{\mathbf{A}} = \mathcal{R}\mathbf{A} - 2\hat{\mathcal{A}}\mathbf{t}, \qquad \hat{A}_0 = A_0 + \mathbf{t}^{\mathrm{T}}(\hat{\mathcal{A}}\mathbf{t} - \mathcal{R}\mathbf{A}). \tag{6.29}$$

From these relations, the parameters of the implicit equation (6.16) are easily obtained. Note that the transformation (scaling, rotation and translation) of a surface leaves its SPs unaltered, *i.e.*, points that are inside (outside) the surface transform into points inside (outside) the transformed surface.

6.2.2 Fuzzy quadric surfaces

Even with quadric surfaces, we can find numerical ambiguities when locating a particle that is close to a surface: because of the limited accuracy of floating-point numbers in a computer, we may be unable to assert with confidence whether the particle is inside or outside the surface. Indeed, similar ambiguities are found whenever a particle crosses an interface; due to round-off errors, the values of the surface function at the calculated intersection point may have either sign, although in this case we do know that the particle has just passed the interface. To get rid of these kinds of ambiguities, all surfaces are treated as fuzzy (Almansa *et al.*, 2016), that is, a surface swells or shrinks very slightly when the particle crosses it, so as to ensure that the particle is at the correct side of the surface. Of course, mathematical surfaces are not altered; instead, we define the SP of a given surface at the particle position $\mathbf{r}_0$ by considering the relative motion of the particle with respect to the surface. Notice that, to apply this strategy, we must know the direction of motion $\hat{\mathbf{d}}$ of the particle.

For quadric surfaces, side ambiguities can be readily resolved by considering only the values of the parameters of the master equation (6.20). On the one hand, the coefficient c is the value of the surface function at $\mathbf{r}_0$ and, in principle, its sign gives the side pointer of the surface at $\mathbf{r}_0$: SP = sign(c), where sign(c) = $+1$ if $c \geq 0$, and = -1 if $c < 0$. Ambiguities may occur only when $|c|$ is smaller than a certain small value ϵ, the "fuzziness" level, which will be specified below. On the other hand, the coefficient

b is the projection of the gradient of $F(\mathbf{r})$ at $\mathbf{r}_0$ along the direction of flight, *i.e.*, the directional derivative of $F(\mathbf{r})$. Hence, if $b > 0$ the particle "leaves" the surface (moving from inside to outside). Conversely, if $b < 0$ the particle "enters" the surface (moves from outside to inside). The surface is made fuzzy by simply defining the SP at an ambiguous point $\mathbf{r}_0$ (*i.e.*, such that $|c| = |F(\mathbf{r}_0)| < \epsilon$) to be the same as the SP at a position slightly advanced along the ray, where the sign of $F(\mathbf{r})$ is unambiguous. That is, we set SP $= -1$ (inside) if $b < 0$ and SP $= +1$ (outside) if $b > 0$. Summarising, the side pointer of a fuzzy quadric surface at the point $\mathbf{r}_0$, ambiguous or not, is given by the following simple algorithm,

```
if |c| > ε, then
   SP=sign(c)
else
   SP=sign(b)
end if
```
(6.30)

Once we have determined the current body, which contains the point $\mathbf{r}_0$ where the particle is, we can sample the length s_0 of the free flight to the next interaction by using the cross sections of the material in the current body. A call to the tracking subroutine will then simulate a "flight" of the particle to a new position, which will be either the end of the step $(\mathbf{r}_0 + s_0\hat{\mathbf{d}})$ or the intersection with one of the surfaces that limit the current body, whichever occurs first. In other words, the tracking subroutine must calculate the distances s to the intersections with the limiting surfaces by solving the corresponding master equation (6.20), and compare these distances with the free-flight length s_0. Note that there is no need to store the SPs of the limiting surfaces in memory because the SP of a surface at $\mathbf{r}_0$ is simply the sign of the c coefficient of its master equation (6.20), which has to be re-evaluated each time the tracking routine is invoked. Furthermore, we can use the calculated c values to verify that the particle effectively is in the declared body, and apply suitable corrections when this is not the case.

After the crossing of an interface, defined by the Eq. $F(\mathbf{r}) = 0$, the particle is placed at a point $\mathbf{r}'$ and the tracking subroutine has to identify the new body by computing the SPs of its limiting surfaces at $\mathbf{r}'$. Although the particle has just crossed one of these surfaces, because of round-off errors, its numerical position coordinates may be at either side of this mathematical surface, and even at a distance such that $|F(\mathbf{r}')| > \epsilon$. When $\mathbf{r}'$ is ambiguous with respect to the surface (*i.e.*, when $|F(\mathbf{r}')| < \epsilon$), algorithm (6.30) gives the correct SP. The difficulty arises when $\mathbf{r}'$ is not ambiguous, in which case the numerical position may be either before the mathematical surface (undershot) or beyond it (overshot). In the case of an overshot, the value of $F(\mathbf{r}')$ has the correct sign, and the program will generally give the correct SP. On the contrary, when there is an undershot, the wrong SP will be assigned. While a slight error in the position coordinates is tolerable, altering the SP implies changing the body and material where the particle moves. With mathematical surfaces, incorrect SPs have to be corrected by keeping track of the surfaces that are crossed by the particle, which complicates the tracking algorithm. An advantage of using fuzzy surfaces is that, in the case of an

undershot, the program usually corrects itself, because one additional jump of negligible length places the particle at an ambiguous position where the correct SP will be assigned.

As indicated above, the tracking subroutine must determine the distances s from the initial position $\mathbf{r}_0$ of the particle to the intersections of the ray $\mathbf{r}_0 + s\hat{\mathbf{d}}$ with the limiting surfaces of the current body. Non-trivial situations can arise when $\mathbf{r}_0$ is ambiguous with respect to the surface, as may occur just after a surface crossing, or when the ray intersects the surface twice and the intersections are very close to each other (*e.g.*, when the ray is almost tangent to the surface or when the ray crosses a cone near its tip). In such situations, the tracking subroutine has to evaluate the SP of the surface and also discriminate whether the surface is going to be crossed again or not. If the surface is a plane ($a = 0$), there is only one intersection; the SP is determined by the signs of c and b, and after updating it we can proceed as if the particle had effectively passed the surface. For non-planar quadric surfaces, we need to consider only the case in which the ray does cross the surface (*i.e.*, when the discriminant Δ is strictly positive). We can therefore assume that Eq. (6.20) has two different roots, s_1 and s_2, sorted in increasing order ($s_1 < s_2$). The general situation is sketched in Fig. 6.5, which shows a ray crossing a quadric surface and the corresponding "master" function, $f(s) = as^2 + bs + c$. Because the master functions for two $\mathbf{r}_0$ points *in the same ray* differ only by a global displacement in s, the graph of this function is a characteristic of the ray. However, the values of the coefficients a, b, and c do depend on the initial position $\mathbf{r}_0$ in the ray. The lower plot in Fig. 6.5 shows the signs of a and b corresponding to different initial positions along the ray. We note that when the point $\mathbf{r}_0$ is an ambiguous point, $|c| = |F(\mathbf{r}_0)| < \epsilon$ and either s_1 or s_2 is close to zero. When s_1 is negative and $s_2 \sim 0$, the ray has just crossed the surface and there are no intersections ahead. When $s_1 \sim 0$ and s_2 is positive, the ray crosses the surface again at s_2. Unfortunately, round-off errors may still lead to ambiguities when both $|s_1|$ and $|s_2|$ are small. Therefore, to reveal the existence of a second intersection ahead, it is preferable to consider some global property of the master function $f(s)$ that is less sensitive to round-off errors. Figure 6.5 shows that we may have four different situations, which are characterised by the signs of b and a:

A) $b > 0$ and $a < 0$: SP $= +1$, second crossing at s_2.
B) $b \geq 0$ and $a > 0$: SP $= +1$, no more crossings.
C) $b \leq 0$ and $a < 0$: SP $= -1$, no more crossings.
D) $b < 0$ and $a > 0$: SP $= -1$, second crossing at s_2.

When the product ab is negative (cases A and D), $s_1 \sim 0$ and the second root is positive, that is, the ray does intersect the surface again at a certain distance from the initial point $\mathbf{r}_0$, even if the numerical value of s_2 turns out to be negative due to round-off errors. In the latter case, we can simply consider that the intersection is at $s_2 = 0$ because the surface is fuzzy. On the other hand, when $ab > 0$ (cases B and C with $b \neq 0$), $s_2 \sim 0$ and the first root is negative, so that there are no intersections beyond $\mathbf{r}_0$. Summarising, in the case of ambiguous points $\mathbf{r}_0$, SP ambiguities are resolved by considering the sign of b [*i.e.*, of the derivative of $f(s)$ at $s = 0$], while the existence of a second crossing ahead is readily recognized from the sign of a.

To complete the description of the tracking algorithm, we only have to specify the

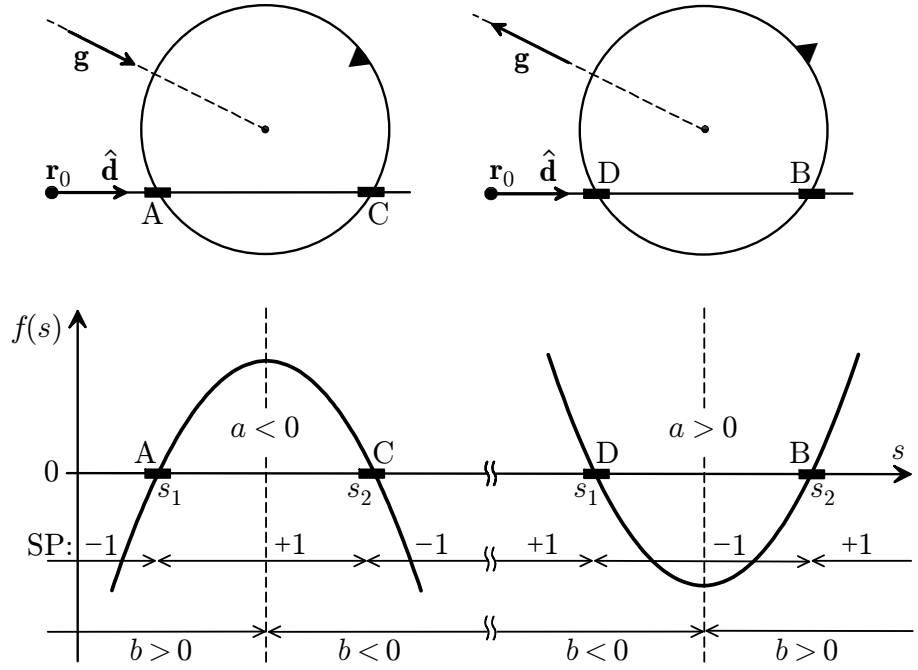

Figure 6.5: Top: Intersections of rays $\mathbf{r}_0 + s\hat{\mathbf{d}}$ with two quadric surfaces, which for simplicity are represented as spheres; the solid triangles indicate the outside of the surfaces (SP = +1). The two surfaces differ only in their orientation (*i.e.*, their defining functions differ by a global negative factor) and, consequently, their gradient vectors $\mathbf{g} = \nabla F(\mathbf{r})$ are in opposite directions. The solid blocks (labelled A, B, C and D) indicate the ray segments (not to scale) where the SP of the surface may be ambiguous, *i.e.*, such that $|F(\mathbf{r})| < \epsilon$. Bottom: Ray intersections described in terms of the master function, $f(s) = as^2 + bs + c$, Eq. (6.20). The path length s increases to the right. The roots s_1 and s_2 of the equation $f(s) = 0$ are the distances at which the ray intersects the surface, sorted in increasing order ($s_1 < s_2$). The signs of the coefficients a and b depend on the initial position $\mathbf{r}_0$ of the particle in the ray, as indicated. (From Almansa *et al.*, 2016)

value of the fuzziness level ϵ, so as to make sure that all points $\mathbf{r}$ such that $|F(\mathbf{r})| > \epsilon$ are assigned the correct SP. Let δ denote the distance that the fuzzy surface will swell or shrink along the direction of the ray. When the surface is not a plane, δ should be much smaller than the distance between the two crossings of the ray, $s_2 - s_1 = \Delta^{1/2}/|a|$. Recalling that the slope of the master function at the intersections is $f'(s_i) = 2as_i + b = \pm\Delta^{1/2}$, we set $\delta = 10^{-12}(s_2 - s_1)$ and

$$\epsilon = f'(s_i)\delta = 10^{-12}\frac{\Delta}{|a|}\,. \tag{6.31a}$$

In the case of planes ($a = 0$), the fuzziness level is assumed to be a constant, *i.e.*,

$$\epsilon \equiv 10^{-12}\,. \tag{6.31b}$$

so that the interval of ambiguous points has a half-length $\delta = \epsilon/|f'(s_i)| = 10^{-12}/|b|$, which increases when the angle between the ray and the normal to the plane increases.

Note that when $b = 0$ (and $a = 0$) the ray is parallel to the plane and does not intersect it. Numerical experiments, using double-precision arithmetic, confirm that the prescriptions (6.31) do work well. The precise value of the numerical constant 10^{-12} is not critical; when it is increased, say by a factor of 100, the ambiguity interval widens by approximately that factor (*i.e.*, surfaces become fuzzier), but we have not observed any harmful consequences.

The complete tracking algorithm is described in Table 6.2. This algorithm is robust, in the sense that it consistently assigns a SP to ambiguous points $\mathbf{r}_0$ and determines the distance to the next crossing, if there is one. Note, however, that two fuzzy surfaces may not be correctly resolved when they are too close to each other. As a measure of the accuracy of the algorithm we may use the resolution, defined as the minimal distance between two surfaces that are neatly resolved. The resolution is determined by the fuzziness level and by the accumulated numerical round-off errors from the surface coefficients and from the calculation of distances to interfaces. In general, the resolution worsens when the distance to the origin of coordinates increases. That is, a small geometrical structure, which is correctly resolved when placed near the origin, may become distorted or invisible when it is translated to a more distant position. With the adopted value of the fuzziness parameter, 10^{-12}, our algorithm is capable of resolving a sphere of unit radius located at a distance of 10^7 length units from the origin.

6.2.3 Modules and genealogical tree

As we have already mentioned, the obvious method to locate a point consists of computing the SPs for *all* surfaces and, then, explore the bodies in ascending order looking for the first that matches the given SPs. During simulation, when a particle attempts to fly a given distance s_0 (step length) in a direction $\hat{\mathbf{d}}$ from a given position $\mathbf{r}_0$ within a body B_0, we need to check whether the particle leaves the body in the course of its flight and, when this occurs, we should halt the particle just after leaving B_0 and resume simulation with the cross sections of the new material. This requires determining the intersections of the particle ray $\mathbf{r}_0 + s\hat{\mathbf{d}}$ $(0 < s \leq s_0)$ with *all* the surfaces that limit the body B_0 (including those that define any other bodies that limit B_0), and checking whether the final position $\mathbf{r}_0 + s\hat{\mathbf{d}}$ remains within B_0 or not.

It should be noted 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 particle ray with the limiting surfaces of B_0 and check if the particle has left the body after any of the intersections. In addition, when the particle leaves the initial body, say after travelling a distance s' $(< s_0)$, we need to locate the point $\mathbf{r}' = \mathbf{r}_0 + s'\hat{\mathbf{d}}$.

The straight procedure sketched above is robust, and easy to program, but becomes too slow for complex geometries. We can speed it up by simply disregarding those elements of the geometry that cannot be reached in a single step (*e.g.*, bodies that are

Table 6.2: Algorithm for computing the intersections of a ray, $\mathbf{r}_0 + s\hat{\mathbf{d}}$, with a fuzzy quadric surface (Almansa *et al.*, 2016). It solves the master equation $f(s) = as^2 + bs + c = 0$ and gives the number n of intersections ahead of the ray. If there are any, the algorithm also provides the corresponding distances s_k ($k = 1$ if $n = 1$; $k = 1, 2$ if $n = 2$) sorted in increasing order. Planes are treated separately to avoid unnecessary calculations.

$n = 0$
`if` $|a| < 10^{-36}$, `then` [the surface is a plane]
  `if` $|b| > 0$, `then`
    $\epsilon = 10^{-12}$
    `if` $|c| < \epsilon$, `then` [the point $\mathbf{r}_0$ is ambiguous]
      SP = sign(b)
    `else`
      SP = sign(c)
      $t = -c/b$
      `if` $t > 0$, set $n = 1$, $s_1 = t$
    `end if`
  `else` [ray parallel to the plane]
    SP = sign(c)
  `end if`
`else` [the surface is not a plane]
  $\Delta = b^2 - 4ac$, $\epsilon = 10^{-12}\Delta/|a|$
  `if` $|c| < \epsilon$, `then` [the point $\mathbf{r}_0$ is ambiguous]
    `ambig` $= 1$, SP = sign(b)
  `else`
    `ambig` $= 0$, SP = sign(c)
  `end if`
  `if` $\Delta < 10^{-36}$, `exit` [no "true" intersections]
  $t_1 = \dfrac{-b}{2a} - \dfrac{\sqrt{\Delta}}{2|a|}$, $t_2 = \dfrac{-b}{2a} + \dfrac{\sqrt{\Delta}}{2|a|}$
  `if` `ambig` $= 0$, `then`
    `if` $t_1 > 0$, set $n = 1$, $s_1 = t_1$
    `if` $t_2 > 0$, set $n = n + 1$, $s_n = t_2$
  `else`
    `if` $ab < 0$, set $n = 1$, $s_1 = \max\{t_2, 0\}$
  `end if`
`end if`

"screened" by other bodies or too far apart). 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 adopt a simple strategy that consists of grouping sets of bodies together to form *modules*.

A module is defined as a connected volume, limited only by quadric surfaces, which contains one or several bodies. A space volume is said to be connected when any two points in the volume can be joined by a trajectory that is completely contained within the volume. 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 material, which automatically fills the cavities of the module (*i.e.*, volumes that do not correspond to a body or to a submodule); these filled cavities are considered as a single new body. 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.*, portions of bodies or submodules that lie outside the module are not allowed) 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 (a radioactive source, an accelerator head, a detector, a phantom, *etc.*) of a more complex material system. To facilitate the definition of the geometry, PENGEOM allows free translations and rotations of individual modules. Thus, the definition of a module (see Table 6.3 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 all its bodies and submodules) with respect to the laboratory reference frame. As before, the rotation is applied first.

In practical simulations with finite geometries, the tracking of a particle should be discontinued when it leaves the material system. In PENGEOM this is done automatically by assuming that the complete system is contained within a single convex module, the *enclosure*, which comprises the whole system. A space volume is convex if the straight segment joining any two points in the volume is wholly contained within the volume. It is also convenient (but not necessary) to require that the enclosure has a finite volume, so that all rays starting from any point within the volume of the enclosure do intersect one of its limiting surfaces at a finite distance. When an enclosure is not defined by the user (*i.e.*, when the geometry consists of several separate modules and/or bodies that are not inside a single module), PENGEOM defines the enclosure as a large sphere of radius 10^7 length units, centred at the origin of coordinates. It is assumed that there is a perfect vacuum outside the enclosure, and in any inner volume that is not a body or a filled module. If the geometry definition contains bodies that extend beyond the enclosure, they are truncated and only the parts inside the enclosure are retained.

Hence, particles that leave the enclosure will never return to the material system.

For programming purposes, it is useful to consider 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. 6.6). 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. Consequently, individual bodies cannot be rotated or translated.

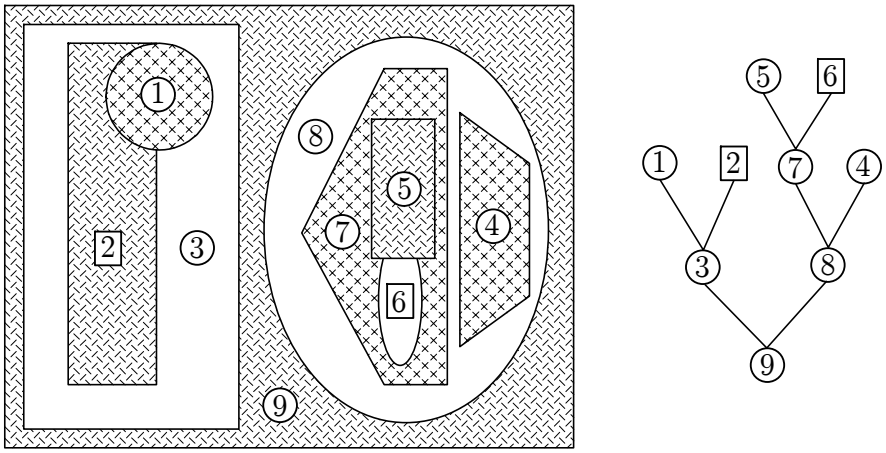

Figure 6.6: Planar cut of a geometry example, showing modules (number labels in circles) and bodies (number labels in squares), and the associated genealogical tree. Notice that a module can always be defined as a body limited by their submodules and daughter bodies, but this affects the structure of the genealogical tree and, therefore, the efficiency (speed) of the geometry operations.

The benefit of using modules is that, while a particle is moving within a module, the only accessible bodies are the daughters (bodies and submodules) of that module. This effectively limits the number of surfaces that need to be analysed at each move. Only when the particle crosses one of the limiting surfaces of the current module, do we need to consider the outer geometry. Furthermore, when a particle leaves a module, it remains within the volume of either the parent module or one of its ancestors. The numerical work needed to locate and track particles may thus be largely reduced by defining a well ramified genealogical tree of modules. Optimal organisation is obtained when each module has a relatively simple inner structure with a small number of daughters. Figure 6.6 shows an example of finely ramified tree. Practical experience indicates that tracking particles through complex quadric geometries may be faster than with elaborate ray-

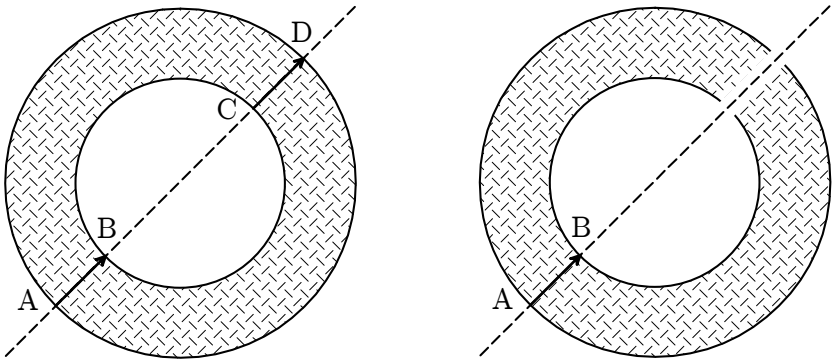

Figure 6.7: Tracking error caused by defining an enclosure that is concave. The example corresponds to a simple geometry consisting of a hollow cylinder in vacuum. The left diagram shows the correct tracking of a particle that crosses the material structure without interacting. When the hollow cylinder is defined as a body, the particle crosses the first wall (A–B), enters the hollow, moves freely to point C, and then crosses the second wall (C–D). If the hollow cylinder is defined as a module, PENGEOM identifies it with the enclosure and the program malfunctions: the particle crosses the first wall (A–B), enters the hollow at B and, because it is in the vacuum, flies freely to infinity, failing to "see" the second wall.

tracing methods, such as octree encoding (Meagher, 1982; Glassner, 1984), provided that the tree of modules is finely ramified.

Although modules may in general be concave, the enclosure module must be convex. Consider, for instance, the case of a simple geometry consisting of a single hollow cylinder in vacuum, Fig. 6.7, which can be defined either as a body or as a module. Since the hollow cylinder is concave, it *must* be defined as a body. If it were defined as a module, it would be considered as the enclosure and PENGEOM would malfunction, because particles that enter the hollow coming from the wall would be assumed to enter the "external" vacuum, and would not stop at the opposite wall. Note that the PENGEOM routines are unable to check whether the root module is convex or concave. This inconsistency becomes apparent when the geometry is analysed with the two-dimensional viewer of GVIEW2D (see Section 6.6); the three-dimensional viewer, however, will generate correct images even when the hollow cylinder is defined as a module (because the light rays used for rendering three-dimensional images do not penetrate the material bodies).

6.3 Geometry-definition file

The geometry is defined from an input text file, which consists of a sequence of blocks, which define the different elements (surfaces, bodies and modules). A definition block consists of a number of strictly formatted text lines; it starts and ends with a separation line filled with zeros (see Table 6.3). The first line in each block must start with one of the defining 8-character strings "`SURFACE␣`", "`SURFACE*`", "`BODY␣␣␣␣`", "`MODULE␣␣`",

Table 6.3: Formats of the geometry definition blocks and default parameter values (Almansa *et al.*, 2016).

```
0000000000000000000000000000000000000000000000000000000000000000000000
SURFACE (  A4)   reduced form
INDICES=(I2,I2,I2,I2,I2)
X-SCALE=(        E22.15        ,  I4)              (DEFAULT=1.0)
Y-SCALE=(        E22.15        ,  I4)              (DEFAULT=1.0)
Z-SCALE=(        E22.15        ,  I4)              (DEFAULT=1.0)
  OMEGA=(        E22.15        ,  I4) DEG          (DEFAULT=0.0)
  THETA=(        E22.15        ,  I4) DEG          (DEFAULT=0.0)
    PHI=(        E22.15        ,  I4) RAD          (DEFAULT=0.0)
X-SHIFT=(        E22.15        ,  I4)              (DEFAULT=0.0)
Y-SHIFT=(        E22.15        ,  I4)              (DEFAULT=0.0)
Z-SHIFT=(        E22.15        ,  I4)              (DEFAULT=0.0)
0000000000000000000000000000000000000000000000000000000000000000000000
SURFACE (  A4)   implicit form
INDICES=( 0, 0, 0, 0, 0)
    AXX=(        E22.15        ,  I4)              (DEFAULT=0.0)
    AXY=(        E22.15        ,  I4)              (DEFAULT=0.0)
    AXZ=(        E22.15        ,  I4)              (DEFAULT=0.0)
    AYY=(        E22.15        ,  I4)              (DEFAULT=0.0)
    AYZ=(        E22.15        ,  I4)              (DEFAULT=0.0)
    AZZ=(        E22.15        ,  I4)              (DEFAULT=0.0)
     AX=(        E22.15        ,  I4)              (DEFAULT=0.0)
     AY=(        E22.15        ,  I4)              (DEFAULT=0.0)
     AZ=(        E22.15        ,  I4)              (DEFAULT=0.0)
     A0=(        E22.15        ,  I4)              (DEFAULT=0.0)
1111111111111111111111111111111111111111111111111111111111111111111111
  OMEGA=(        E22.15        ,  I4) DEG          (DEFAULT=0.0)
  THETA=(        E22.15        ,  I4) DEG          (DEFAULT=0.0)
    PHI=(        E22.15        ,  I4) RAD          (DEFAULT=0.0)
X-SHIFT=(        E22.15        ,  I4)              (DEFAULT=0.0)
Y-SHIFT=(        E22.15        ,  I4)              (DEFAULT=0.0)
Z-SHIFT=(        E22.15        ,  I4)              (DEFAULT=0.0)
0000000000000000000000000000000000000000000000000000000000000000000000
BODY    (  A4) Text describing the body ...
MATERIAL(  A4)
SURFACE (  A4), SIDE POINTER=( 1)
BODY    (  A4)
MODULE  (  A4)
0000000000000000000000000000000000000000000000000000000000000000000000
MODULE  (  A4)  Text describing the module ...
MATERIAL(  A4)
SURFACE (  A4), SIDE POINTER=( 1)
BODY    (  A4)
MODULE  (  A4)
1111111111111111111111111111111111111111111111111111111111111111111111
  OMEGA=(        E22.15        ,  I4) DEG          (DEFAULT=0.0)
  THETA=(        E22.15        ,  I4) DEG          (DEFAULT=0.0)
    PHI=(        E22.15        ,  I4) RAD          (DEFAULT=0.0)
X-SHIFT=(        E22.15        ,  I4)              (DEFAULT=0.0)
Y-SHIFT=(        E22.15        ,  I4)              (DEFAULT=0.0)
Z-SHIFT=(        E22.15        ,  I4)              (DEFAULT=0.0)
0000000000000000000000000000000000000000000000000000000000000000000000
```

Table 6.3: Formats of the geometry definition blocks and default parameter values (Almansa *et al.*, 2016) (continued).

```
0000000000000000000000000000000000000000000000000000000000000000000000
CLONE   (  A4)   Copies a module and moves it
MODULE  (  A4)   Original module
1111111111111111111111111111111111111111111111111111111111111111111111
  OMEGA=(        E22.15        ,  I4) DEG          (DEFAULT=0.0)
  THETA=(        E22.15        ,  I4) DEG          (DEFAULT=0.0)
    PHI=(        E22.15        ,  I4) RAD          (DEFAULT=0.0)
X-SHIFT=(        E22.15        ,  I4)              (DEFAULT=0.0)
Y-SHIFT=(        E22.15        ,  I4)              (DEFAULT=0.0)
Z-SHIFT=(        E22.15        ,  I4)              (DEFAULT=0.0)
0000000000000000000000000000000000000000000000000000000000000000000000
INCLUDE
   FILE=(filename.ext)
0000000000000000000000000000000000000000000000000000000000000000000000
END      0000000000000000000000000000000000000000000000000000000000000000
```

`CLONE␣␣␣`", "`INCLUDE␣`", "`INCLUDE*`" or "`END␣␣␣␣␣`" (here, blank characters are denoted by "␣"; they are essential!). A line starting with the string "`END␣␣␣␣␣`" after a separation line discontinues the reading of geometry data. Each element is identified by its type (surface, body or module) and a four-character string, which will be referred to as the user label. The user label is only used to designate the elements in the geometry definition file. In the PENGEOM subroutines, geometry elements are identified by a numerical label: elements of a given type are numbered consecutively, according to their input order (see below). Notice that bodies and modules are considered as elements of the same type (*i.e.*, assigning the same user label to a body and to a module will cause an error of the reading routine).

In the input file, numerical quantities must be written within the parentheses in the specified format. Lengths are in arbitrary units, typically in cm; angles can be given in either degrees (`DEG`) or radians (`RAD`). When angles are in degrees, it is not necessary to specify the unit. Numerical parameters are assigned the default values indicated in Table 6.3; lines that define parameters with their default values can be removed from the definition file. Parameters in each definition block can be entered in any order. They can even be defined several times, in which case, only the last input value is accepted.

Each numerical parameter is followed by an `I4` value, which must be set equal to zero (or negative) to make the parameter value effective. When this field contains a positive integer `IP`, the parameter is set equal to the value stored in the `IP`-th component of the array `PARINP`, an input argument of subroutine `GEOMIN` (see Section 6.4). This permits the user to modify the geometry parameters from the main program (*e.g.*, to rotate and/or translate a module).

Informative text (as many lines as desired) can be written at the beginning of the file, before the first separation line. A short text description can be written at the end of the first line of each element, after the closing parenthesis of the numerical field. These

element descriptions are inherited by cloned elements, and are useful for understanding and modifying large geometry definition files. Furthermore, comment lines, starting with "C␣" or "c␣" (as in Fortran, but with a blank in the second column) or with "#", can be introduced anywhere in the geometry file. The information in comment lines, however, is not transferred to cloned elements. The program uses only the leftmost columns of each definition line, any text following the parameter field is ignored.

• Surfaces defined in reduced form

◦ Surface parameters are optional and can be entered in any order. Note that the scale parameters `X-SCALE`, `Y-SCALE`, and `X-SCALE` must be greater than zero.

◦ The keyword `SURFACE*` serves to define "fixed" surfaces, which will not be affected by possible translations or rotations in subsequent stages of the geometry definition. These starred surfaces are useful, *e.g.*, to define aligned modules (see below).

• Surfaces defined in implicit form

◦ When a quadric surface is defined in implicit form, the indices must be set to zero; this switches the reading subroutine to implicit mode.

◦ The rotation and the translation are optional. The line filled with `1`'s ends the definition of coefficients and starts that of transformation parameters (it can be skipped if no transformation parameters are entered).

◦ Starred surfaces, declared with the keyword `SURFACE*`, can also be defined in implicit form. These surfaces are fixed, *i.e.*, they are not modified by subsequent translations or rotations.

N.B.: The use of double planes (*e.g.*, $z^2 - 1 = 0$) should be avoided, because pairs of single planes ($z + 1 = 0$ and $z - 1 = 0$) are easier to handle and may allow possible surface redundancies to be unveiled (see Section 6.4).

• Bodies

◦ Bodies are delimited by previously defined surfaces, bodies and modules.

◦ The material number (2nd line, an integer) must conform with the convention adopted in the simulation. Void inner volumes can be described as material bodies with `MATERIAL` set equal to 0 (or a negative number).

◦ A line is required to define each limiting surface, with its side pointer, and each limiting body or module. Limiting surfaces, bodies and modules can be entered in any order.

N.B.: Bodies are assumed to be defined in ascending order so that, in principle, it would not be necessary to declare the limiting bodies and modules. However, to speed up the calculations, it is required to declare explicitly all the elements (surfaces, bodies and modules) that actually limit the body that is being defined. Omission of a limiting body will cause inconsistencies unless the materials in the limiting and the limited bodies are the same.

If we had chosen the alternative of not declaring all limiting elements, at each free flight of the transported particle we would have to calculate the intersections of the particle ray with *all* the surfaces of the sister bodies and modules, even those that do not limit the body where the particle is moving. For complex geometries this may be too costly.

In principle, any quadric geometry can be defined with only surfaces and bodies. As an example, Table 6.4 (Almansa *et al.*, 2016) shows the definition file of the structure displayed in Fig. 6.3. For complex geometries, however, tracking becomes very slow, and can only be accelerated by introducing modules and setting a properly ramified genealogical tree. Modules also offer the advantages of allowing global transformations (rotation and translations) and cloning.

Table 6.4: Definition file `example1` of the geometry shown in Fig. 6.3.

```
A sphere with a cylindrical inset.
0000000000000000000000000000000000000000000000000000000000000000000000
SURFACE ( AS1)   Plane Z=0.0
INDICES=( 0, 0, 0, 1, 0)
0000000000000000000000000000000000000000000000000000000000000000000000
SURFACE ( AS2)   Plane Z=-3.0
INDICES=( 0, 0, 0, 1, 0)
Z-SHIFT=(-3.000000000000000E+00,   0)
0000000000000000000000000000000000000000000000000000000000000000000000
SURFACE ( AS3)   Cylinder R=1.0, Z axis
INDICES=( 1, 1, 0, 0,-1)
0000000000000000000000000000000000000000000000000000000000000000000000
BODY    ( AB1)   Cylinder
MATERIAL(   1)
SURFACE ( AS1), SIDE POINTER=(-1)
SURFACE ( AS2), SIDE POINTER=(+1)
SURFACE ( AS3), SIDE POINTER=(-1)
0000000000000000000000000000000000000000000000000000000000000000000000
SURFACE ( AS4)   Sphere R=2.0
INDICES=( 1, 1, 1, 0,-1)
X-SCALE=(+2.000000000000000E+00,   0)
Y-SCALE=(+2.000000000000000E+00,   0)
Z-SCALE=(+2.000000000000000E+00,   0)
0000000000000000000000000000000000000000000000000000000000000000000000
BODY    ( AB2)   Hollow sphere
MATERIAL(   2)
SURFACE ( AS4), SIDE POINTER=(-1)
BODY    ( AB1)
0000000000000000000000000000000000000000000000000000000000000000000000
END      0000000000000000000000000000000000000000000000000000000000000000
```

- **Modules**

  ◦ The material (which must be explicitly declared) fills the cavities of the module. As in the case of bodies, `MATERIAL` = 0 corresponds to vacuum. The filled cavities are considered as a single new body, which is assigned the same label as the module.

  ◦ The limiting surfaces must define a connected volume. All inner bodies and modules

(submodules) must be declared. Notice that these cannot extend outside the module's volume and that a submodule cannot overlap with the other submodules and bodies.

◦ Limiting surfaces, inner bodies and submodules can be entered in any order.

◦ Rotation and translation are optional and apply to all elements of the module, except starred surfaces. The line filled with `1`'s ends the definition of elements and starts that of transformation parameters; it can be omitted if no transformation parameters are entered.

It should be noted that modules do not need to coincide with bodies of the material system. In certain cases, it may be convenient to consider modules that contain portions of bodies, even if this requires splitting homogeneous bodies by means of auxiliary surfaces.

A single surface can be used to define several bodies and/or modules. Conversely, the same surface can be defined several times (with different labels), *e.g.*, to keep the definition of a body complete and independent of those of other bodies. The unnecessary duplication of a surface does not affect the simulation speed because PENGEOM identifies "redundant" surfaces and effectively removes them.

Rotation or translation of a module modifies all the surfaces of its descendants (except starred surfaces) and, therefore, a transformed surface must be redefined if it is to be used again. Thus, if the system contains two identical modules in different positions (*e.g.*, two detectors in a coincidence experiment), each of them must be defined explicitly. In principle, this does not require too much editing work; after generating the first of the two modules, we can just duplicate its definition blocks and assign new values to their labels and parameters. A simpler and more versatile method is to use the `CLONE` operation, which automatically clones a module (with its submodules and inner bodies) and changes the position and orientation of the cloned module. This operation is helpful to define systems with repeated structures such as array detectors and multi-leaf collimators.

• Cloning a module

◦ Rotation and translation apply to all elements of the module, except starred surfaces. The line filled with `1`'s starts the definition of transformation parameters. Notice that it is necessary to apply a translation or a rotation (or both), to avoid overlaps between the original and cloned modules (which would make the geometry definition inconsistent).

◦ Bodies cannot be cloned. To clone a body that is limited only by surfaces, define it as a module.

◦ All the surfaces that define the original module (and its descendants) are cloned and transformed collectively with the module. In certain cases, however, it may be convenient to keep some surfaces unaltered (*e.g.*, those that limit both the original and cloned modules). We can do this by using the keyword `SURFACE*` in the definition of the surface; starred surfaces behave as normal surfaces, except for the fact that their coefficients are not affected by translation/rotation of the cloned module.

◦ When we run one of the geometry viewers (see below), PENGEOM generates an output

file, named `geometry.rep`, that strictly defines the same geometry as the input file. However, in `geometry.rep` the elements of all the cloned modules are defined explicitly (and labelled consecutively). We can edit, and rename, this file to modify the elements of cloned modules.

• Including a geometry file

The `INCLUDE` option allows a predefined structure to be inserted (a scintillation detector, an encapsulated radioactive source, ...) within the geometry file. The inserted structure is defined by a complete definition file (*i.e.*, ending with an "`END␣␣␣␣`" line).

◦ The name of the included file must be written between the parentheses. It may be up to twelve characters long; if it is shorter, blanks must be left on the right-hand side of the field. Only one-level `INCLUDE`s are allowed, *i.e.*, an included file cannot contain any `INCLUDE` blocks.

◦ The user labels of an included file do not need to be different from those used in the main file or in other included files.

◦ PENGEOM 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), PENGEOM 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.

◦ A file can also be included by using the keyword `INCLUDE*`. In this case, the included file is considered as if it were part of the main file. That is, the user labels of the elements in the included file must be different from those in the main file. In addition, the elements of the included file can be referenced in the main file to define new bodies or modules.

The use of modules and included files offers much flexibility to define new geometries. The example in Table 6.5 and Fig. 6.8 starts from the definition file of Table 6.4, a sphere with a cylindrical inset, which is included by preserving the original labels of the elements; the two bodies are grouped in a single module (limited by the plane $z = -3$), which is rotated and cloned to produce the structure shown in Fig. 6.8.

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. 6.9. 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 both of them into a two-cell module (3). To build this module, we only need to specify its limiting surfaces. The two-cell module is then cloned (6) and the resulting two modules are grouped together to produce a four-cell module (7). The

Table 6.5: Definition file `example2` illustrating the use of the `INCLUDE*` and `CLONE` operations.

```
Two spheres at the ends of a cylinder
0000000000000000000000000000000000000000000000000000000000000000000000
INCLUDE*                    (keeps original element labels)
   FILE=(    example1)
0000000000000000000000000000000000000000000000000000000000000000000000
SURFACE ( BS1)   Plane Z=2.0
INDICES=( 0, 0, 0, 0, 0)
     AZ=(+1.000000000000000E+00,   0)
     A0=(-2.000000000000000E+00,   0)
0000000000000000000000000000000000000000000000000000000000000000000000
SURFACE ( BS2)   Cylinder R=2.0, Z axis
INDICES=( 1, 1, 0, 0,-1)
X-SCALE=(+2.000000000000000E+00,   0)
Y-SCALE=(+2.000000000000000E+00,   0)
0000000000000000000000000000000000000000000000000000000000000000000000
MODULE  ( BM1)   Sphere and cylinder grouped in a single module
MATERIAL(   0)
SURFACE ( AS2), SIDE POINTER=( 1)
SURFACE ( BS1), SIDE POINTER=(-1)
SURFACE ( BS2), SIDE POINTER=(-1)
BODY    ( AB1)
BODY    ( AB2)
1111111111111111111111111111111111111111111111111111111111111111111111
# The module is rotated and shifted.
  THETA=(+90.00000000000000E+00,   0)
    PHI=(+90.00000000000000E+00,   0)
Y-SHIFT=(+3.000000000000000E+00,   0)
0000000000000000000000000000000000000000000000000000000000000000000000
CLONE   ( BM2)   The module is cloned and rotated 180 deg
MODULE  ( BM1)   about the Z axis.
1111111111111111111111111111111111111111111111111111111111111111111111
    PHI=(+180.0000000000000E+00,   0)
0000000000000000000000000000000000000000000000000000000000000000000000
SURFACE ( BS3)   Plane Y=-5.0
INDICES=( 0, 0, 0, 0, 0)
     AY=(+1.000000000000000E+00,   0)
     A0=(+5.000000000000000E+00,   0)
0000000000000000000000000000000000000000000000000000000000000000000000
MODULE  (FULL)   The whole system is enclosed in a single module
MATERIAL(   0)
SURFACE ( BS1), SIDE POINTER=(-1)
SURFACE ( BS2), SIDE POINTER=(-1)
SURFACE ( BS3), SIDE POINTER=( 1)
MODULE  ( BM1)
MODULE  ( BM2)
1111111111111111111111111111111111111111111111111111111111111111111111
  OMEGA=(+0.000000000000000E+00,   0) DEG          (DEFAULT=0.0)
  THETA=(+0.000000000000000E+00,   0) DEG          (DEFAULT=0.0)
    PHI=(+0.000000000000000E+00,   0) DEG          (DEFAULT=0.0)
0000000000000000000000000000000000000000000000000000000000000000000000
END      0000000000000000000000000000000000000000000000000000000000000000
```

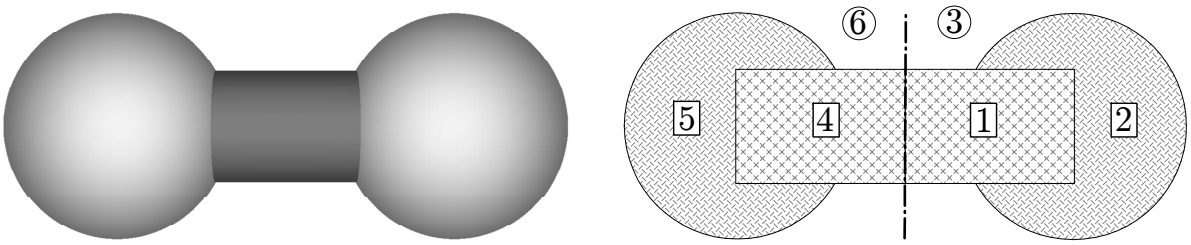

Figure 6.8: Three-dimensional (left) and two-dimensional (right) views of the geometry described by the definition file `example2` listed in Table 6.5. Module 3 groups the original bodies 1 and 2; module 6 is generated by cloning module 3, and contains bodies 4 and 5.

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. 6.9, 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.

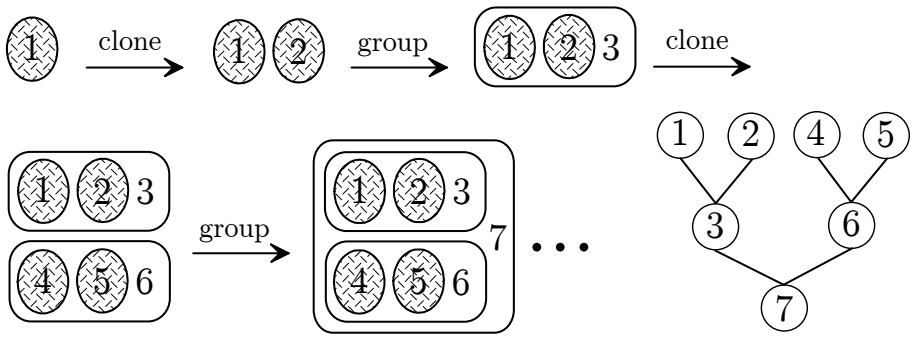

Figure 6.9: Schematic representation of the construction of an array of cells by progressively cloning and grouping modules obtained from a primary cell module. The labels of the modules are the internal labels assigned by PENGEOM. Notice the evolution of the modular tree at each step.

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.

6.4 The subroutine package PENGEOM

The Fortran subroutine package PENGEOM consists of several subroutines which perform geometrical operations guided by a steering main program. Most of the input/output of these subroutines is through the Fortran module `TRACK_mod`[3], Table 7.3, which contains the following relevant variables:

- `X, Y, Z` : position coordinates.
- `U, V, W` : direction cosines of the direction of movement.
- `IBODY` : body where the particle moves.
- `MAT` : material in body `IBODY`.

In PENGEOM the type of variables is determined by the Fortran implicit definition, *i.e.*, the first character of the name of a variable defines its type. Variables whose names start with `(A-H,O-Z)` are double precision (or `REAL*8`), while variables starting with `I-N` are `INTEGER*4`.

Array sizes and other geometrical parameters are defined in the Fortran module `PENGEOM_mod`, Table 6.6, which contains the following quantities:

- `NS` : maximum number of surfaces.
- `NB` : maximum number of bodies.
- `NXG` : maximum number of limiting elements in a body or module.
- `BALIAS(NB)` : user labels of bodies (*i.e.*, those in the input file). Defined only when declared explicitly in the geometry definition file. Bodies that result from the cloning of modules (as well as those defined in an `INCLUDE`d geometry file) do not have a user label.
- `MATER(NB)` : material in each body.
- `KDET(NB)` : array used to define impact detectors.
- `DSTOT` : path length travelled during the last step, including segments in void volumes.
- `KSLAST` : label of the last surface crossed by the particle when it enters a material medium coming from the external vacuum, or when it leaves a material body and enters either a different body or a void volume. Notice that `KSLAST` is defined only when `NCROSS` $\neq 0$.

[3]This is the Fortran module used in PENELOPE to transfer particle state variables. PENGEOM may only alter the values of the variables `X`, `Y`, `Z`, `IBODY`, and `MAT`; the rest of variables in module `TRACK_mod` remain unaltered.

Table 6.6: Source listing of the Fortran module `PENGEOM_mod`.

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

The variables `DSTOT` and `KSLAST` are part of the output of subroutine `STEP` (see below). They are useful for generating images of the geometry.

The subroutines to be invoked from the main program are the following:

`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` (nonnegative).

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 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 PENGEOM, is printed on the output file (UNIT IWR). This part of the output file describes the actual geometry used in the simulation.

> N.B.: During the simulation, geometry elements are identified by the internal labels assigned by PENGEOM, which are written in the output file. These labels may be different from the user labels in the geometry definition file.

For complex geometries, it may be practical to refer to a given body by means of its user label, *i.e.*, the label that appears in the geometry definition file. The user labels of bodies are kept in array KALIAB, which is made accessible through the Fortran module PENGEOM_mod. BALIAS(KB) is the user label of the body with internal label KB.

Subroutine GEOMIN also 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 in UNIT IWR, following the listing of the geometry definition.

SUBROUTINE LOCATE

Determines the body that contains the point with coordinates (X, Y, Z).

• Input values (through Fortran module TRACK_mod):
X, Y, Z : Particle position coordinates.
U, V, W : Direction cosines of the direction of movement.

• Output values (through Fortran module TRACK_mod):
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)

This subroutine performs the geometrical part of the track simulation under the guidance of the steering main program, which determines the direction of motion (U,V,W) and the length DS of each free flight. The particle starts from the point (X,Y,Z) and proceeds to travel a length DS in the direction (U,V,W) within the body

IBODY 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.

• 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).

• Input-output values (through Fortran module TRACK_mod):
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 IBODY (automatically changed when the particle 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 or leaves the enclosure).

• Other output values (through Fortran module PENGEOM_mod):
DSTOT : path length travelled during the last step, including segments in void volumes.
KSLAST : label of the last surface crossed by the particle when it enters a material medium coming from the external vacuum, or when it leaves a material body and enters either a different body or a void volume.

When the particle travels through a void volume between two material bodies, KSLAST corresponds to the exit surface of the material body where the particle started its flight; the entrance surface to the new body must be determined by moving the particle backwards a small distance (to place it in the void volume) and calling subroutine STEP with DS > DSTOT − DSEF. Notice that KSLAST is defined only when NCROSS ≠ 0.

For the handling and storage of geometric information we take advantage of the structure of the genealogical tree. The geometry enclosure (see Section 6.2.3) 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 bodies of the system (*i.e.*, in a void region). Notice that when we have found that a module KB does contain the point, we only need to consider the surfaces KS such that FLAG(KB, KS) = 1, 2 or 4 to perform the next step.

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 the tracking main program, 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 values). 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

(the output value of NCROSS is then larger than zero) 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). It is worth mentioning that the tracking routines may write warning messages in UNIT=26. These warnings refer to inconsistencies caused by round-off errors, which in most cases occur when the particle leaves the system (through the spherical enclosure of radius 10^7) and are innocuous.

In the case of tangent surfaces (*e.g.*, a sphere inside a cylinder of the same radius), the limited accuracy of double precision in Fortran may produce distortions of the geometry when a particle crosses the touching surfaces, because the program may be unable to identify which of the two surfaces should be crossed first. Examples of this kind of conflict are shown in Fig. 6.10. The left diagram represents a sphere, B_1, lying on the flat surface of a block, B_2. Because of round-off errors, the numerical surfaces are such that the sphere slightly penetrates the block. In this case a ray, $\mathbf{r}_0 + s\hat{\mathbf{d}}$, that starts from the block body will enter the sphere before leaving the block. As this eventuality is not contemplated, the behaviour of the tracking routines is essentially unpredictable. The user should make sure that such situations do not occur by altering slightly the conflicting surfaces. Of course, the conflict is avoided by declaring the sphere B_1 as a limiting body of the block B_2.

Similar problems may occur when a mathematical surface is defined more than once (*e.g.*, to limit two adjacent bodies, or a body and its mother module). A mathematical surface can be either defined directly or obtained through rotation and translation of a module that involves it. In the latter case, accumulated round-off errors from these transformations may yield a slightly different surface. The right diagram in Fig. 6.10 shows an example of adjacent bodies, a cube (B_1) on a larger block (B_2), where the common surface, a plane, is defined once for each body. Let us consider that the two bodies are defined separately. Because of assumed round-off errors, the two surfaces are slightly different and the bodies overlap a little. A particle coming from the block B_2 can then reach the cube B_1 before leaving the block, and then the tracking algorithm may become inconsistent. Subroutine GEOMIN removes redundant surfaces to prevent such situations, and to avoid unnecessary calculations. Two surfaces are identified as redundant when their respective coefficients differ by a constant factor, with a relative tolerance of 10^{-14}. Note, however, that accumulated round-off errors can impede the identification of redundant surfaces. *The use of double planes is discouraged, because they may hide possible surface redundancies.* For instance, if our geometry contains the plane $z - 1 = 0$ and the double plane $z^2 - 1 = 0$, GEOMIN will not take any action. If, instead, we replace the double plane by the equivalent pair of planes $z + 1 = 0$ and $z - 1 = 0$, the latter will be identified as redundant and will be removed, thus avoiding possible conflicts caused by round-off errors. Furthermore, the calculation of trajectory intersections with two planes is easier than with a double plane.

PENGEOM admits up to 10,000 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 (maxi-

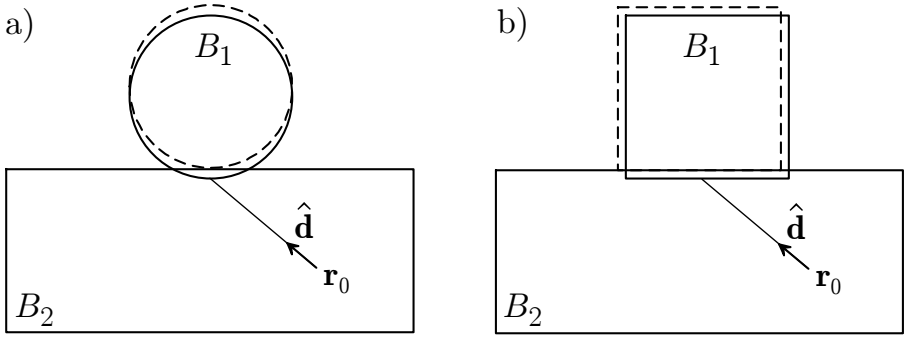

Figure 6.10: Possible inconsistencies caused by round-off errors, largely magnified. a) A sphere lying on a block. In reality the sphere (dashed) is tangent to the block plane surface, while the numerical sphere (solid) intersects the block surface. b) A cube on top of a block. The lower surface of the cube and the upper side of the block are the same mathematical surface. When they are defined as different surfaces, round-off errors may cause small variations and, occasionally, the interpenetration of the two bodies.

mum number of surfaces) and `NB` (maximum number of bodies) in the Fortran module `PENGEOM_mod`. 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 (*i.e.*, surfaces with `FLAG(KB,KS)` < 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. Additionally, simulation of a geometry with a module or body defined by such a large number of elements would be extremely slow.

Evidently, the speed of the geometry subroutines depends greatly on the structure of the modules' genealogical tree. The responsibility for optimising it rests with the user. As indicated above, subroutine `GEOMIN` issues a detailed report on the geometry structure as part of the output file. Inspecting this report and checking whether the modular tree is sufficiently ramified is recommended. For complex geometries it may be convenient to split large homogeneous bodies into several smaller parts pertaining to different modules, even when this requires defining new surfaces.

6.5 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 Fortran module `PENGEOM_mod`. 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.

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.

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. 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.

In practical simulations of radiation transport, impact detectors can be used to tally the energy spectrum and the angular distribution of "detected" particles (*i.e.*, particles that enter an active body) or, more specifically, to generate a *phase-space file*, where the state variables of particles at the detector entrance are recorded. 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.

Impact detectors may also be employed to calculate the average distribution of fluence with respect to energy in the volume of the detector, $\overline{\Phi}(E)$ [see Eq. (5.39)]. The distribution that can be evaluated from the simulation is the following

$$\mathcal{V}\,\overline{\Phi}(E) = \int_{\mathcal{V}} \mathrm{d}\mathbf{r}\,\Phi(\mathbf{r}, E), \tag{6.32}$$

where $\mathcal{V}$ is the volume of the impact detector (which is not calculated by the geometry routines) and $\Phi(\mathbf{r}, E)$ is the distribution of fluence with respect to energy at the point $\mathbf{r}$. Usually, $\mathcal{V}\,\overline{\Phi}(E)$ is calculated separately for each kind of particle. The quantity $\mathcal{V}\,\overline{\Phi}(E)\,\Delta E$ is the total path length travelled within the detector volume by particles of the considered kind that have energies in the interval $(E, E + \Delta E)$. It is worth noting that $\mathcal{V}\,\overline{\Phi}_E$, the distribution of fluence with respect to energy integrated over the volume of the detector, has dimensions of length/energy. When the flux density is nearly constant over the volume of an impact detector, $\overline{\Phi}_E$ approximates the distribution of fluence with respect to energy at points inside the detector.

The distribution of fluence with respect to energy, Eq. (6.32), can be used to obtain various global quantities of interest in dosimetry (see Chapter 5) and spectroscopy. For instance, in the case of electrons, the number of ionisations of the K-shell of atoms of the element Z_i within the volume $\mathcal{V}$ of the detector is given by

$$n_{\mathrm{K},i} = \int \left[\mathcal{V}\,\overline{\Phi}_E\right]\,\mathcal{N} x_i \sigma_{\mathrm{si,K}}(E)\,\mathrm{d}E, \tag{6.33}$$

where $\sigma_{\mathrm{si,K}}(E)$ is the atomic cross section for K-shell ionisation of the element Z_i (see Section 3.2.6), $\mathcal{N}$ is the number of molecules per unit volume, Eq. (1.102), and x_i is the number of atoms of the element Z_i in a molecule. Note that $\lambda_{\mathrm{si,K}}^{-1} = \mathcal{N} x_i \sigma_{\mathrm{si,\,K}}$ is the average number of ionisations per unit path length. If the detector is a very thin foil, the probability that an electron that enters the foil causes an ionisation is very small and, therefore, the value of $n_{\mathrm{K},i}$ evaluated by counting ionisations may be affected by considerable statistical uncertainties. On the contrary, all electrons that cross the foil contribute to the fluence, and the average number of ionisations derived from Eq. (6.33) will usually be more accurate.

Instead of evaluating the distribution of fluence with respect to energy, it may be advantageous to compute the number $n_{\mathrm{K},i}$ of ionisations by using the following track length estimator, which is similar to that described in Section 5.2.3 for the calculation of the absorbed dose. During simulation of the i-th shower, we consider the path lengths $s_{i,j}$ travelled by electrons of energy E_j within the volume $\mathcal{V}$ of the detector, and score the products $s_{i,j}\,\lambda_{\mathrm{si,K}}^{-1}(E_j)$ (mean number of ionisations along the segment). After generating N showers, we have the estimator

$$n_{\mathrm{K},i} = \frac{1}{N} \sum_i \left(\sum_j s_{i,j}\, \lambda_{\mathrm{si,K}}^{-1}(E_j) \right) , \tag{6.34}$$

where the first summation is over individual showers, and the second is over electron track segments in $\mathcal{V}$. An advantage of using track length estimators is that they allow the direct evaluation of statistical uncertainties using the general method of Section 1.5.

6.6 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. Almansa *et al.* (2016) have developed a Java application named PENGEOM.JAR which provides handier tools for editing and debugging geometry definition files, as well as two- and three-dimensional viewers. In this Section we shall limit our considerations to the less sophisticated programs contained in the PENELOPE distribution package.

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 leftmost pixel of each row 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 6.7, 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.

Table 6.7: 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,                    +
+  s --> change the window scale (zoom level),                 +
+  p --> change the coordinates of the screen centre,          +
+  c,m  --> change view mode (bodies <-> materials),           +
+  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.      +
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

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, change the coordinates of the window centre to place it 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 distribution of energy deposited within a particular body). The material and the body label shown in the lower right-hand corner of the window correspond to the central body (the one at the intersection of the axes).

The program 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 on 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.

It should be mentioned that the use of fuzzy quadric surfaces in the current version of PENGEOM gives a dramatic improvement in resolution over previous versions of the package, which used static surfaces and a rough protection against round-off errors. The viewers GVIEW2D and GVIEW3D now correctly render very small objects (*e.g.*, a spherical shell with inner radius of 10^{-9} units and a thickness of 5×10^{-11} units) as well as large structures with small details (*e.g.*, a spherical shell of 10^{6} radius and 10^{-5} thickness), which could not be displayed with the old versions of the codes.

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.

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 `geometry.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.

6.7 A short tutorial

The geometry-definition file can be written with any simple text editor, using fixed-width fonts to keep the prescribed line formats. 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 is operative. The starting file should look like this:

```
END      0000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000
SURFACE (    )   reduced form
INDICES=( 1, 1, 1, 1, 1)
X-SCALE=(+1.000000000000000E+00,   0)              (DEFAULT=1.0)
Y-SCALE=(+1.000000000000000E+00,   0)              (DEFAULT=1.0)
Z-SCALE=(+1.000000000000000E+00,   0)              (DEFAULT=1.0)
  OMEGA=(+0.000000000000000E+00,   0) DEG          (DEFAULT=0.0)
  THETA=(+0.000000000000000E+00,   0) DEG          (DEFAULT=0.0)
    PHI=(+0.000000000000000E+00,   0) RAD          (DEFAULT=0.0)
X-SHIFT=(+0.000000000000000E+00,   0)              (DEFAULT=0.0)
Y-SHIFT=(+0.000000000000000E+00,   0)              (DEFAULT=0.0)
Z-SHIFT=(+0.000000000000000E+00,   0)              (DEFAULT=0.0)
0000000000000000000000000000000000000000000000000000000000000000000
SURFACE (    )   implicit form
INDICES=( 0, 0, 0, 0, 0)
    AXX=(+0.000000000000000E+00,   0)              (DEFAULT=0.0)
    AXY=(+0.000000000000000E+00,   0)              (DEFAULT=0.0)
    AXZ=(+0.000000000000000E+00,   0)              (DEFAULT=0.0)
    AYY=(+0.000000000000000E+00,   0)              (DEFAULT=0.0)
    AYZ=(+0.000000000000000E+00,   0)              (DEFAULT=0.0)
    AZZ=(+0.000000000000000E+00,   0)              (DEFAULT=0.0)
     AX=(+0.000000000000000E+00,   0)              (DEFAULT=0.0)
     AY=(+0.000000000000000E+00,   0)              (DEFAULT=0.0)
     AZ=(+0.000000000000000E+00,   0)              (DEFAULT=0.0)
     A0=(+0.000000000000000E+00,   0)              (DEFAULT=0.0)
1111111111111111111111111111111111111111111111111111111111111111111
  OMEGA=(+0.000000000000000E+00,   0) DEG          (DEFAULT=0.0)
  THETA=(+0.000000000000000E+00,   0) DEG          (DEFAULT=0.0)
    PHI=(+0.000000000000000E+00,   0) RAD          (DEFAULT=0.0)
X-SHIFT=(+0.000000000000000E+00,   0)              (DEFAULT=0.0)
Y-SHIFT=(+0.000000000000000E+00,   0)              (DEFAULT=0.0)
Z-SHIFT=(+0.000000000000000E+00,   0)              (DEFAULT=0.0)
0000000000000000000000000000000000000000000000000000000000000000000
BODY    (    )   text
MATERIAL(    )
SURFACE (    ), SIDE POINTER=( 1)
BODY    (    )
MODULE  (    )
```

```
0000000000000000000000000000000000000000000000000000000000000000000000
MODULE  (    )   text
MATERIAL(    )
SURFACE (    ), SIDE POINTER=( 1)
BODY    (    )
MODULE  (    )
1111111111111111111111111111111111111111111111111111111111111111111111
  OMEGA=(+0.000000000000000E+00,   0) DEG          (DEFAULT=0.0)
  THETA=(+0.000000000000000E+00,   0) DEG          (DEFAULT=0.0)
    PHI=(+0.000000000000000E+00,   0) RAD          (DEFAULT=0.0)
X-SHIFT=(+0.000000000000000E+00,   0)              (DEFAULT=0.0)
Y-SHIFT=(+0.000000000000000E+00,   0)              (DEFAULT=0.0)
Z-SHIFT=(+0.000000000000000E+00,   0)              (DEFAULT=0.0)
0000000000000000000000000000000000000000000000000000000000000000000000
CLONE   (    )   copies one module and moves it
MODULE  (    )   original module
1111111111111111111111111111111111111111111111111111111111111111111111
  OMEGA=(+0.000000000000000E+00,   0) DEG          (DEFAULT=0.0)
  THETA=(+0.000000000000000E+00,   0) DEG          (DEFAULT=0.0)
    PHI=(+0.000000000000000E+00,   0) RAD          (DEFAULT=0.0)
X-SHIFT=(+0.000000000000000E+00,   0)              (DEFAULT=0.0)
Y-SHIFT=(+0.000000000000000E+00,   0)              (DEFAULT=0.0)
Z-SHIFT=(+0.000000000000000E+00,   0)              (DEFAULT=0.0)
0000000000000000000000000000000000000000000000000000000000000000000000
INCLUDE
   FILE=(filename.ext)
0000000000000000000000000000000000000000000000000000000000000000000000
```

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, together with an error message issued by the program.

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,

```
------------------------------------------------------------------
         Visualisation of reduced quadric surfaces.
Define the desired quadric (surface 1) by entering its indices.
The region with side pointer -1 (inside the quadric) corresponds
to MATERIAL=1.
0000000000000000000000000000000000000000000000000000000000000000000000
```

```
SURFACE (   1)   Reduced quadric
INDICES=( 1, 1,-1, 0, 1)
0000000000000000000000000000000000000000000000000000000000000000
BODY    (   1)   The interior of the quadric
MATERIAL(   1)
SURFACE (   1), SIDE POINTER=(-1)
0000000000000000000000000000000000000000000000000000000000000000
END      0000000000000000000000000000000000000000000000000000000
```

The definition files `example1` and `example2` shown in Tables 6.4 and 6.5 correspond to the geometries displayed in Figs. 6.3 and 6.8, respectively. The file `example2` illustrates the use of the `INCLUDE*` and `CLONE` operations. Of course, this simple geometry can be defined in a less intricate way; the definition adopted in `example2` is intended to show that 1) elements defined within included starred files may also be referred to by means of their original user labels and 2) the transformation of a module also transforms its limiting surfaces and submodules. We have also enclosed the entire system in a single module so that you may rotate and/or displace it arbitrarily by editing the definition file.

Writing a geometry file is nothing more than routine work. After a little practice, you can define quite complex systems by using only surfaces and bodies. You will soon realise that the visualisation programs (as well as the actual simulations!) slow down when the number of elements in the geometry increases. The only way of speeding up the programs is to group the bodies into modules. The best strategy for improving the calculation speed is to build relatively simple modules and combine them into larger parent modules to obtain a genealogical tree where the number of daughters of each module is small.

You may save a lot of time by defining each body separately (and checking it carefully) and then inserting it into the progressing module that, once finished, will be added to the file. Notice that the input element labels are arbitrary (as long as they are not repeated for elements of the same kind) and that we can insert new elements anywhere in the file, either manually or using the `INCLUDE` operation. Once the geometry definition file is complete, we can obtain an equivalent file, with elements labelled according to the internal sequential order, by running one of the viewer programs and re-naming the `geometry.rep` file.

The geometry files `example1` and `example2`, together with several other files of more complex geometries, are included in the distribution package. They can be directly visualised by running GVIEW2D and GVIEW3D. The file `well` defines a scintillation well detector in great detail; we have defined the system as a module, so that you can rotate the entire detector by editing 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; this is the reason why the two-dimensional viewer displays only sections of the geometry on planes that are perpendicular to the coordinate axes. The file `saturne` describes the head of a medical accelerator, quite a complicated geometry with 96 surfaces and 44 bodies. The structure `phantom`, which corresponds to a mathematical anthropomorphic

phantom, consists of 264 surfaces and 169 bodies and modules (Cristy and Eckerman, 1987; Almansa *et al.*, 2016). 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. The file `onion` defines a set of concentric spherical shells with radii ranging from 10^{-9} to 10; this file is useful to demonstrate the robustness and the actual spatial resolution of our tracking algorithm.

The files `glass` (a champagne glass) and `chair` (a chair) show that common objects can be described quite precisely with only quadric surfaces; in simple cases, such as `glass`, we do not need to use modules, which are useful only to accelerate the calculations, or to allow cloning. The files `chair2`, and `chair32` illustrate the use of the `CLONE` operation. The latter describes a small auditorium with 32 chairs and provides an example of optimally ramified tree.

We cannot finish without a word of caution about the use of PENGEOM, 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 features directly; this may enable a substantial reduction of the number of operations by taking full advantage of the symmetries and other peculiarities of the material system.

Chapter 7

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. In the following, particles emitted from the source will be referred to as primary particles. Typically, each shower is initiated by a single primary particle.

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, photoelectric absorption, Compton scattering and pair production) and as fluorescent radiation (characteristic x rays and Auger electrons). In Rayleigh scattering and Compton scattering, the incident photon is absorbed and the scattered photon is considered as a secondary particle. PENELOPE simulates fluorescent radiation that results from vacancies produced in inner shells[1] by photoelectric absorption and Compton scattering of photons, by triplet production and by electron/positron impact ionisation. Relaxation of these vacancies is followed until all inner shells are filled up, *i.e.*, until the vacancies have migrated to outer shells.

As it is a subroutine package, PENELOPE cannot operate by itself. The user must provide a steering main program for his or her particular problem. Nevertheless, this main program is normally fairly simple because it only has to control the evolution of the tracks generated 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

[1]Inner shells are K, L, M and N shells that have ionisation energies larger than the cutoff energy E_c given by Eq. (2.112). Shells with ionisation energies less than E_c are considered as outer shells.

or her 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 PENGEOM (see Chapter 6). The distribution package also includes examples of main programs for cylindrical geometries and for general quadric geometries. Although these main programs are mostly intended to illustrate the use of the simulation routines, they do allow the study of many cases of practical interest. The program system is written mostly in Fortran 77 format, but uses modules and a few other features of Fortran 90. Therefore, the programs should run on any platform with a Fortran 90 or later compiler.

7.1 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, $\lambda_B \simeq (150 \text{ eV}/E)^{1/2}$ Å, is similar to or greater than the interatomic spacing (~ 2 Å). 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 cutoff energy E_c, Eq. (2.112), 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 13,000 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). In order to organise these subroutines, 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, `PH` for photoelectric absorption, `RA` for Rayleigh, `CO` for Compton, 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).

7.1.1 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 an input material data file. 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, or weight fraction, of each element), 2) mass density, 3) mean excitation energy and 4) energy and oscillator strength of plasmon-like excitations, see Section 3.2.1. Alternatively, for a set of 280 prepared materials, the program `material` can read composition data directly from the `pdcompos.pen` 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 percentage number of atoms of the elements.

The PENELOPE database consists of the following 995 ASCII files,

`pdatconf.p14 ...` Atomic ground-state configurations, ionisation energies (Carlson, 1975), central values, $J_i(p_z = 0)$, of the one-electron shell Compton profiles (Biggs *et al.*, 1975, recalculated from the radial functions given by the MCDF program of Desclaux, 1975; 1977), and atomic level widths (taken from Campbell and Papp, 2001, and from the LLNL Evaluated Atomic Data Library of Perkins *et al.*, 1991) for the elements, from hydrogen to einsteinium ($Z = 1 - 99$).

Table 7.1: List of the 280 pre-defined materials included in the `pdcompos.pen` file, with their identifying numbers (adapted from Berger, 1992).

***	ELEMENTS (id. number = atomic number):				
1	Hydrogen	34	Selenium	67	Holmium
2	Helium	35	Bromine	68	Erbium
3	Lithium	36	Krypton	69	Thulium
4	Beryllium	37	Rubidium	70	Ytterbium
5	Boron	38	Strontium	71	Lutetium
6	Amorphous carbon	39	Yttrium	72	Hafnium
7	Nitrogen	40	Zirconium	73	Tantalum
8	Oxygen	41	Niobium	74	Tungsten
9	Fluorine	42	Molybdenum	75	Rhenium
10	Neon	43	Technetium	76	Osmium
11	Sodium	44	Ruthenium	77	Iridium
12	Magnesium	45	Rhodium	78	Platinum
13	Aluminium	46	Palladium	79	Gold
14	Silicon	47	Silver	80	Mercury
15	Phosphorus	48	Cadmium	81	Thallium
16	Sulfur	49	Indium	82	Lead
17	Chlorine	50	Tin	83	Bismuth
18	Argon	51	Antimony	84	Polonium
19	Potassium	52	Tellurium	85	Astatine
20	Calcium	53	Iodine	86	Radon
21	Scandium	54	Xenon	87	Francium
22	Titanium	55	Cesium	88	Radium
23	Vanadium	56	Barium	89	Actinium
24	Chromium	57	Lanthanum	90	Thorium
25	Manganese	58	Cerium	91	Protactinium
26	Iron	59	Praseodymium	92	Uranium
27	Cobalt	60	Neodymium	93	Neptunium
28	Nickel	61	Promethium	94	Plutonium
29	Copper	62	Samarium	95	Americium
30	Zinc	63	Europium	96	Curium
31	Gallium	64	Gadolinium	97	Berkelium
32	Germanium	65	Terbium	98	Californium
33	Arsenic	66	Dysprosium	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  Ferroboride
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  Mixed 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 methacrilate (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 vapour
280  Xylene
```

`pdcompos.pen` ... This file contains composition data, mass densities and mean excitation energies for 280 materials, adapted from the database of the ESTAR program of Berger (1992). The list of materials is given in Table 7.1. 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 $Z > 99$. Gases are characterised by assuming a pressure of 1 atm and a temperature of 20 °C.

`material-list.txt` ... List of materials predefined in file `pdcompos.pen`, with their identification numbers (same information as in Table 7.1 above).

`pdrelax.p11` ... Relaxation data for singly-ionised atoms with a vacancy in a K, L, M, or N 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 7.2), 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. Other transition energies are calculated from the energy eigenvalues of the Dirac–Hartree–Fock–Slater equations for neutral atoms (Perkins *et al.*, 1991).

Table 7.2: Numerical labels used to designate atomic electron shells. In the case of non-radiative transitions, the label 30 indicates outer shells with small ionisation energies.

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.p08` with `ZZ`=atomic number (01–99). These files contain integrated cross sections for elastic scattering of electrons and positrons by neu-

tral 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.p08` 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) which covers the energy interval from 50 eV to 100 MeV. All cross sections are in cm^2.

99 files named `peldxZZZ.p08` 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.
The ELSEPA database is distributed with the ICRU Report 77 (2007).

99 files named `pdebrZZ.p08` 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, from 1 keV to 10 GeV, and 32 reduced photon energies W/E, the same for ell elements and energies. This grid 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 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.p18` **...** 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. (3.175). The distribution

parameters were obtained by fitting a set of 910 benchmark partial-wave shape functions calculated with the program of Poškus (2018).

99 files named `pdaffZZ.p08` with `ZZ`=atomic number (01–99). They contain tables of atomic form factors and incoherent scattering functions of neutral free atoms, generated from the EPDL (Cullen *et al.*, 1997). The first line of each file contains the atomic number `ZZ` and the number of data lines; each subsequent line gives
1st column: momentum transfer q, in units of $m_e c$. The same q grid for all elements.
2nd and 3rd columns: atomic form factor $F(q, Z)$ and incoherent scattering function $S(q, Z)$, respectively.

99 files named `pdgraZZ.p08` with `ZZ`=atomic number (01–99). Tables of anomalous scattering factors and total cross sections for Rayleigh scattering of photons with energies up to 100 GeV by free neutral atoms, generated from the EPDL (Cullen *et al.*, 1997). The first line of each file contains the atomic number `ZZ` and the number of data lines; each subsequent line gives
1st column: energy E of the photon in eV.
2nd and 3rd columns: real and imaginary parts of the anomalous scattering factor, respectively.
4th column: atomic cross section for Rayleigh scattering, in cm^2.

99 files named `pdgppZZ.p11` with `ZZ`=atomic number (01–99). Total cross sections for electron-positron pair and triplet 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 *et al.* (2005). 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 and 3rd columns: total cross sections for pair production and for triplet production, respectively, in barn (10^{-24} cm^2).

99 files named `pdgphZZ.p18` with `ZZ`=atomic number (01–99), containing photoelectric total atomic cross sections and partial cross sections for photoionisation of inner shells (K shell and L, M, and N subshells with binding energies higher than 40 eV) for the elements and photon energies in the range from 50 eV to 1 GeV. The data were calculated by running the program `photacs` of Sabbatucci and Salvat (2016) with the normalisation screening correction included[2] The format is the following,
1) The first line contains the atomic number `ZZ`, the number `NS` of shells for which the partial cross sections are tabulated, and the number of grid energies `NGE`.
2) The second and third lines list the shell number codes and the ionisation energies (in eV), respectively.

[2]The compressed file `pdgph-photacs.zip` contains an alternative database of photoelectric cross sections that was calculated without the normalization screening correction. To use the uncorrected cross sections, the user must replace the files `pdgphZZ.p18` with those in `pdgph-photacs.zip`.

3) 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 to L3, M1 to M5, and N1 to N7 respectively (all cross sections in barn). Only subshells with binding energies higher than 40 eV 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 GeV) with the grid of absorption edges of the element, and adding additional points (where needed) in order to ensure that linear log-log interpolation will not introduce relative errors larger than about 0.001.

99 files named `pdesiZZ.p14` with `ZZ`=atomic number (01–99), containing total (integrated) cross sections for ionisation of inner shells (K shell, L, M and N subshells with binding energies larger than 50 eV) by impact of electrons with kinetic energies in the range from 50 eV to 1 GeV. These cross sections were calculated numerically using a combination of the distorted-wave and plane-wave Born approximations (Bote and Salvat, 2008; Llovet *et al.*, 2014b). The format of the files is the following:
1) The first line gives the atomic number `ZZ`, the number of grid energies `NGE`, and the number `NS` of shells for which the partial cross sections are tabulated.
2) The second and third lines list the shell number codes and the ionisation energies (in eV), respectively.
3) Each of the following lines contains a value of the kinetic energy of the projectile (in eV) and the corresponding ionisation cross sections (in cm^2) of the shells K, L1 to L3, M1 to M5, and N1 to N7, respectively (all cross sections in barn). For low-Z elements, L, M and N subshells are empty and, therefore, they do not appear in the table.

99 files named `pdpsiZZ.p14` with `ZZ`=atomic number (01–99), containing total (integrated) cross sections for ionisation of inner shells (K shell, L, M and N subshells) by impact of positrons with kinetic energies in the range from 50 eV to 1 GeV. These cross sections were calculated numerically by Bote and Salvat (2008) using a combination of the distorted-wave and plane-wave Born approximations. The format of the files is the same as that of the `pdesiZZ.p14` files.

Atomic cross sections for 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/cm^2) and eV/mtu, respectively. Photon mass attenuation coefficients are expressed in cm^2/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 modified Wentzel DCS for electron and positron elastic scattering is completely defined by the mean free paths and transport mean free paths (see Section 3.1.2). 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 and Rayleigh attenuation coefficients, which are obtained by simple linear log-log interpolation.

To simulate geometrical structures with several materials, the names of the corresponding material data files generated by the program MATERIAL are declared as elements of the array `PMFILE`. PENELOPE labels the `M`-th material in this array 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 `penelope.f` source file.

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. When the material data file does not exist, TABLES builds it before computing the output tables. These tables can be plotted on the screen by running the provided GNUPLOT scripts. 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.

7.1.2 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 Fortran modules `TRACK_mod` and `PENELOPE_mod`, which are listed in Tables 7.3 and 7.4.

The module `TRACK_mod` contains the following particle state variables and control parameters:

- `KPAR` : type 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 is stored here.

• `IBODY` : this label serves to identify different bodies in complex material structures.

• `MAT` : material where the particle moves (*i.e.*, the material in the body labelled `IBODY`).

• `ILB(5)` : an auxiliary array of 5 labels that describe the origin of secondary particles (see Table 7.6 below). It is useful, *e.g.*, to study partial contributions from particles

Table 7.3: Source listing of the Fortran module `TRACK_mod`. This module contains particle state variables and geometry location parameters.

```
      MODULE TRACK_mod
C
C  ****  Particle TRACK variables (to be initialised before calling
C        subroutine START).
      SAVE  ! Saves all items in the module.
C
C  ----  Energy, position, direction, and weight.
      DOUBLE PRECISION :: E,X,Y,Z,U,V,W,WGHT
C  ----  Particle type, current body, and material.
      INTEGER*4 :: KPAR,IBODY,MAT
C  ----  Particle history flags.
      INTEGER*4, DIMENSION (5) :: ILB
C
C  ****  Photon polarisation.
C  ----  Polarised photons if IPOL=1, otherwise unpolarised photons.
      INTEGER*4 :: IPOL=0
C  ----  Stokes parameters.
      DOUBLE PRECISION :: SP1,SP2,SP3
C
C  ****  The particle age (time elapsed since the start of the shower)
C        is recorded when LAGE=.TRUE.
      LOGICAL :: LAGE =.FALSE.
      DOUBLE PRECISION :: PAGE=0.0D0
C
C  ****  Random-hinge slowing-down parameters (output from subroutines
C        JUMP and KNOCK).
C
C            <----------------- step ----------------->
C           hard event -------- hinge ------- hard event
C            <---- segment 0 ----><---- segment 1 ---->
C
C     MHINGE ... labels the two segments (jumps) of a step between
C                 two hard events;
C                 = 0 (1) for the segment before (after) the hinge,
C     EOSEGM ... energy at the beginning of the segment,
C     DESOFT ... energy loss along the whole step,
C     SSOFT .... effective stopping power, = DESOFT/step_length.
      INTEGER*4 :: MHINGE
      DOUBLE PRECISION :: EOSEGM,DESOFT,SSOFT
C
      END MODULE TRACK_mod
```

originated by a given process.

• `IPOL, SP1, SP2, SP3` : Photon polarisation flag, and Stokes parameters.

• `LPAGE, PAGE` : Age flag and current value of the particle age.

• `MHINGE`, `EOSEGM`, `SSOFT`, `DESOFT` : Characteristics of the slowing-down of charged particles in class II simulation (see Section 4.3.3), delivered as part of the output of subroutine `JUMP` (see below). `MHINGE` identifies the two segments (jumps) of a step between two hard events [= 0 for the segment before the hinge, and = 1 for the segment after the hinge]; `EOSEGM` is the energy of the particle at the beginning of the track segment; `SSOFT` is the effective stopping power of soft energy-loss interactions; and `DESOFT` is the energy lost along the step, assuming that the particle flies the whole length of the step.

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.

N.B.: During the simulation, all energies and lengths are expressed in eV and cm, respectively.

The scattering of polarised photons is simulated by using the conventions defined in Appendix C and the theory presented in Section 2.7. Arbitrary partial polarisations are described in terms of the Stokes parameters relative to the "natural" basis of linear-polarisation states [see Eqs. (C.6)]. The method allows primary photons with arbitrary partial polarisation to be tracked. In principle, it can also be used to track secondary photons (*i.e.*, bremsstrahlung quanta and x rays). However, this requires specifying the initial polarisation of secondary photons because PENELOPE does not define it. In the example programs `pencyl` and `penmain` (see Section 7.2), we assume that secondary photons are initially unpolarised. The simulation of polarised photons is controlled through the variables `IPOL` and (`SP1`,`SP2`,`SP3`) in module `TRACK_mod`. The simulation of photon polarisation effects is activated by setting `IPOL=1`. The quantities `SP1`, `SP2` and `SP3` are the Stokes parameters of the photon. These parameters must be defined before starting the simulation of each photon; they are modified internally by the PENELOPE subroutines.

The age of particles is defined as the time elapsed since the emission of the primary particle that started the shower. On-the-fly calculation of particle ages is activated by setting `LAGE=.TRUE.` in the main program. The variable `PAGE` must be set equal to zero when a primary is started, and its value must be updated after each segment of a particle's trajectory (by adding the time spent in travelling that segment). Notice that, because segment lengths may be altered by the geometry routines, the age increments must be calculated in the main program. Secondary particles resulting directly from interactions are assigned initial ages equal to that of the parent particle. The ages of x rays and Auger electrons are defined by subroutine `RELAX`, which adds a random delay (corresponding to the finite lifetimes of atomic levels) to the age of the parent particle that produced the ionisation.

Table 7.4: Source listing of the Fortran module PENELOPE_mod.

```
      MODULE PENELOPE_mod
      SAVE  ! Saves all items in the module.
C
C  ****  Maximum number of materials in the geometry.
C
      INTEGER*4, PARAMETER :: MAXMAT=10
C
C  ****  Simulation parameters (must be defined before calling the
C        initialisation subroutine PEINIT).
C
C  ----  Absorption energies, EABS(KPAR,MAT).
      DOUBLE PRECISION, DIMENSION (3,MAXMAT) :: EABS=50.0D0
C  ----  Electron/positron transport parameters.
      DOUBLE PRECISION, DIMENSION (MAXMAT) :: C1=0.01D0,C2=0.01D0
      DOUBLE PRECISION, DIMENSION (MAXMAT) :: WCC=1.0D2,WCR=1.0D2
C
C  ****  Size of the secondary stack (maximum number of particles that
C        can be stored).
      INTEGER*4, PARAMETER :: NMS=1000
C
C  ****  Energy interpolation, number of grid points.
      INTEGER*4, PARAMETER :: NEGP=200
C
C  ****  Global information on the material system (defined by
C        subroutine PEINIT).
C  ----  Number of materials present.
      INTEGER*4 :: NMAT
C  ----  Material densities and its reciprocals.
      DOUBLE PRECISION, DIMENSION (MAXMAT) :: DEN=1.0D0, RDEN=1.0D0
C
      END MODULE PENELOPE_mod
```

The Fortran module PENELOPE_mod, Table 7.4, contains array dimensions, simulation parameters, material densities, and variables defining the slowing-down of charged particles. The following array dimensions determine the amount of memory used by the program:

- MAXMAT : Maximum number of materials in the geometry.
- NMS : Size of the secondary stack, maximum number of particles that can be stored.
- NEGP : Number of data points in the tables of energy-dependent quantities.

These dimensions can be changed by editing the penelope.f source file, if necessary. The number NEGP of energy grid points is large enough to make sure that interpolation errors are not larger than the global uncertainties of the cross sections. Increasing the stack size NMS is not recommended. Stack overflows that may occur with the variance reduction technique of interaction forcing (Section 1.6.1) are usually due to excessively high (and ineffective!) forcing factors.

We recall that the label KPAR in module TRACK_mod identifies the type 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 absorbed 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 about 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 about 0.05 is adequate. The maximum allowed value of C2(M) is 0.2.

• WCC(M) : Cutoff energy loss, $W_{\rm cc}$ (in eV), for hard inelastic collisions.

• WCR(M) : Cutoff energy loss, $W_{\rm cr}$ (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 (~ 0.01). 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.

Other quantities in module PENELOPE_mod are:

• NMAT : Number of different materials that have been loaded (output of subroutine PEINIT, see below).

• DEN(M), RDEN(M) : Mass density of material M, and its reciprocal (output of subroutine PEINIT). Used to evaluate absorbed doses in homogeneous materials.

The simulation package is initialised from the main program with the statement

→ CALL PEINIT(EMAX,NMATER,IWR,INFO,PMFILE)

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:

`EMAX ...` 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) `EMAX` $= 1.21(\texttt{EP} + \mathrm{m_e}c^2)$; in this special case, the maximum energy is larger than the initial kinetic energy.

`NMATER ...` Number of different materials (less than or equal to `MAXMAT`). A material can be loaded several times, *e.g.*, to use different sets of simulation parameters in different volumes.

`IWR ...` Output unit.

`INFO ...` Determines the amount of information that is written on the output unit. Minimal for `INFO`$\leq$`1`, and increasingly detailed for `INFO=1`, `2`, and `3` (fully detailed printout).

`PMFILE ...` Array of 20-character strings and dimension `MAXMAT`. The first `NMAT` elements are the names of the material-data files. The file with the name `PMFILE(M)` defines the material number `M` (the order of filenames in `PMFILE` is important!).

For the preliminary computations, `PEINIT` needs to know the particle absorption energies `EABS(KPAR,M)` and the simulation parameters `C1(M)`, `C2(M)`, `WCC(M)` and `WCR(M)` for the various materials. These variables have 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 initiates a new step, *i.e.*, it forces the following interaction event to be a hinge. 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 type of particle because it checks whether the energy is within the expected range, and can thus help to detect inconsistencies 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; notice that each step between hard interactions consists of two track segments.

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.

In the case of electrons/positrons, subroutine `JUMP` delivers information on the energy-loss due to soft interactions along the step through the variables `EOSEGM`, `DESOFT`, and `SSOFT` in module `PENELOPE_mod`. Their respective output values are the energy E of the particle at the beginning of the track segment, the sampled soft energy loss ω along the step, and the effective soft stopping power, $S_{\rm soft}$. This information allows estimation of the energy loss in track segments that are truncated at an interface, and the time spent by the particle in travelling each segment (see Section 4.3.3).

`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 7.5.

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 and Rayleigh inverse mean free paths, 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`).

For electrons and positrons, the energy loss ω due to soft events along a step is delivered as the energy `DE` deposited at the hinge (`ICOL=1`), where the energy of the particle is reduced ($E \leftarrow E - $ `DE`). For some purposes, it is convenient to describe the stopping effect of soft events as a continuous slowing-down process with stopping power $S_{\rm soft}$. In this case, the energy loss along a track segment of length s is equal to $s\,S_{\rm soft}$. Notice that, to avoid double counting, when `ICOL = 1`

the output value DE from subroutine KNOCK must then be set equal to zero. In the main programs pencyl and penmain (Section 7.2), the soft energy loss along each segment is deposited in a random position uniformly distributed in the segment. When the particle crosses an interface, the programs adequately account for the reduced energy loss in the truncated track segment.

Table 7.5: Code number (ICOL) for the various interaction events.

ICOL	electrons (KPAR=1)	photons (KPAR=2)	positrons (KPAR=3)
1	hinge	coherent (Rayleigh) scattering	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 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.

SUBROUTINE STORES(E,X,Y,Z,U,V,W,WGHT,KPAR,ILB,IPOL) ... Stores a particle in the secondary stack. Arguments have the same meaning as in module TRACK_mod, but refer to the particle that is being stored. The variables IBODY and MAT are set equal to the current values in module TRACK_mod.
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), or to describe the simultaneous emission of several particles from a radioactive source.

As mentioned above, ILB(5) is an array of labels that describe the origin of secondary particles. The possible values and meanings of these labels are described in Table 7.6. 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 appropriate labels to each particle of the induced shower. The ILB label values are delivered by subroutine SECPAR, through the module TRACK_mod, and remain unaltered during the simulation of the particle track. Because photons are quanta, in the cases of Rayleigh and Compton

scattering the incident photon is absorbed and a new, "scattered" photon is emitted. However, to avoid unnecessary manipulations, after a scattering event the scattered photon is assigned the same generation ILB(1) as the incident photon and it is tracked automatically. Hence a photon history [a sequence of flights and scatterings of photons with the same ILB(1) value] may represent the evolution of several photons belonging to different generations.

Table 7.6: Values and meanings of the particle labels ILB(.).

ILB(1)	generation of the particle; 1 for primary particles, 2 for their direct descendants, *etc.* Primary (source) particles are assumed to be labelled with ILB(1)=1 in the main program.
ILB(2)	type KPAR of parent particle, only if ILB(1)>1 (secondary particles and "scattered" photons).
ILB(3)	interaction mechanism ICOL (see Table 7.5) 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 $= Z \cdot 10^6 + \texttt{IS1} \cdot 10^4 + \texttt{IS2} \cdot 100 + \texttt{IS3}$, where Z is the atomic number of the emitting atom and IS1, IS2 and IS3 are the labels of the active atomic electron shells (see Table 7.2). For instance, ILB(4) = 29010300 designates a K-L2 x ray from copper ($Z = 29$), and ILB(4) = 29010304 indicates a K-L2-L3 Auger electron from the same element. When ILB(4) $\neq 0$, the value of ILB(3) indicates the interaction mechanism that caused the initial vacancy in the decaying atom.
ILB(5)	this label can be defined by the user; it is transferred to all descendants of the particle.

For simplicity, in the present Section we assume that the energy loss of electrons and positrons resulting from soft events is deposited at the hinge. The sequence of calls to generate a random track is then independent of the type of particle that is being simulated. The generation of random showers proceeds as follows (see Fig. 7.1):

(i) Set the initial state of the primary particle, *i.e.*, assign values to the state variables: type of particle KPAR, energy E, position coordinates $\mathbf{r}$ =(X,Y,Z) and direction of movement $\hat{\mathbf{d}}$ =(U,V,W). Specify the body and material where the particle moves by defining the values of IBODY and MAT, respectively. Set WGHT=1 and ILB(1)=1. When the primary particle is a photon and the simulation of polarised photons is activated (IPOL=1), specify the Stokes parameters SP1, SP2 and SP3.

(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, $\mathbf{r} \leftarrow \mathbf{r} + \mathtt{DS}\,\hat{\mathbf{d}}$:

- If the track has crossed an interface, stop the particle at the position where the track intersects the interface. Change to the new material (the one behind the interface) by redefining the variables `IBODY` and `MAT`.
  When the particle escapes from the system, simulation of the track has 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`$=0$, 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 relevant events. The required information is delivered by the simulation routines, either as output arguments or through the variables in modules `TRACK_mod` and `PENELOPE_mod`.

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 6) can be used to handle arbitrary quadric geometries. The flow diagram of a main program for PENELOPE and PENGEOM is shown in Fig. 7.2. As PENGEOM performs all geometrical operations automatically, the structure of the main program is relatively simple (cf. Figs. 7.1 and 7.2).

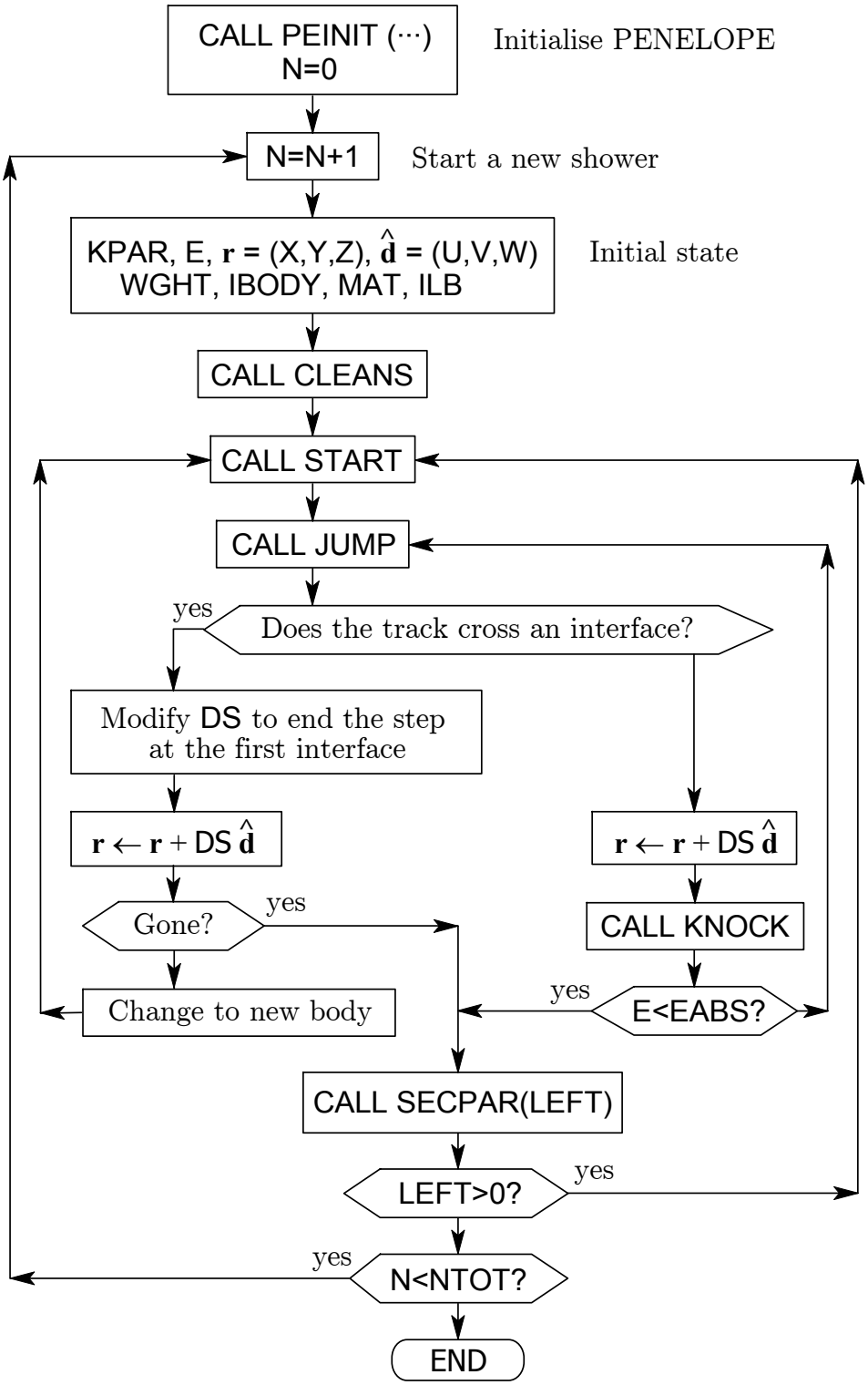

Figure 7.1: Flow diagram of the main program for simulating electron-photon showers with PENELOPE.

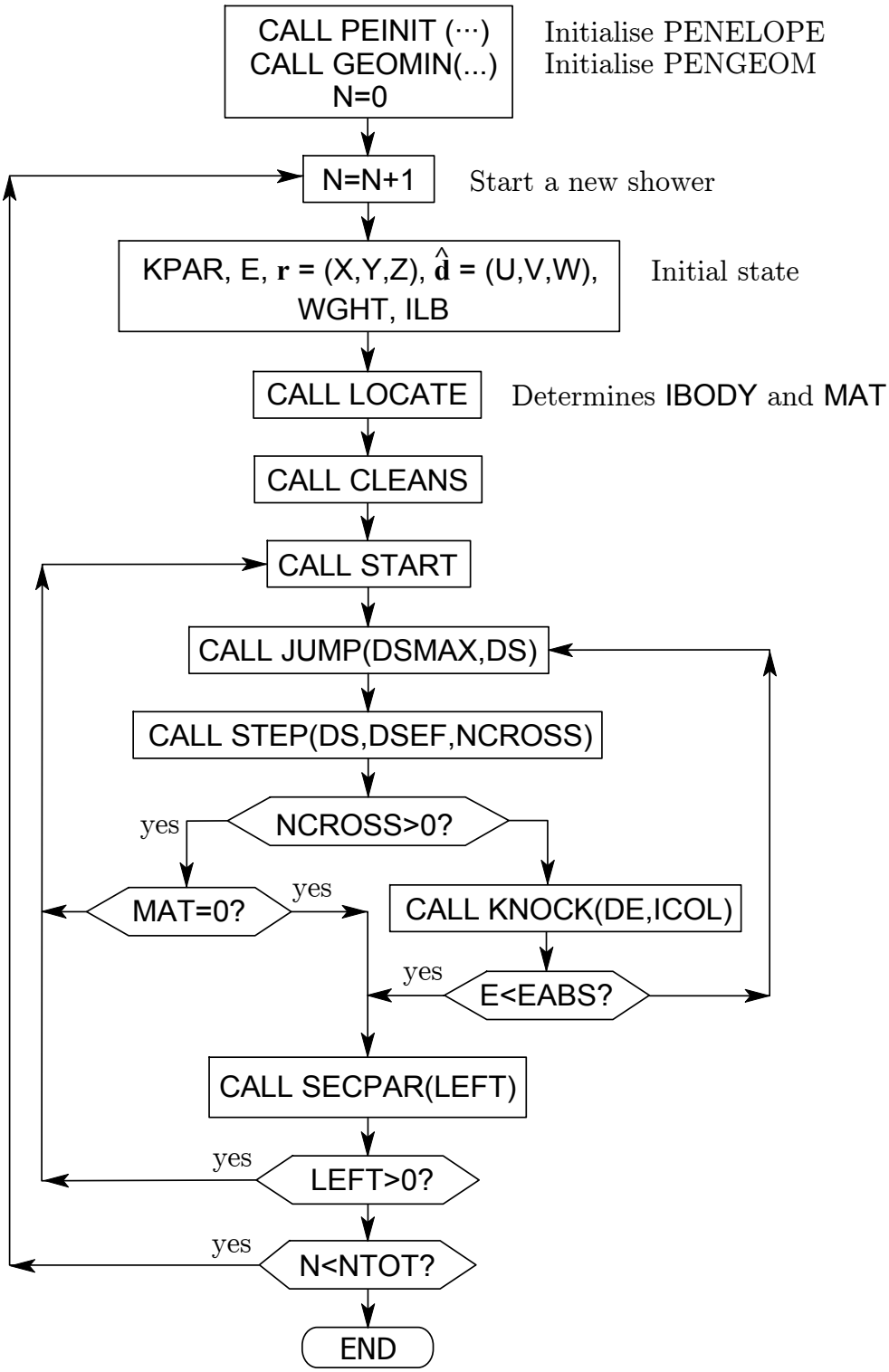

Figure 7.2: Flow diagram of the main program for simulating electron-photon showers in quadric geometries with PENELOPE and PENGEOM.

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 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),EMAX). 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<EMAX 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 set in the main program and transferred through the common block RSEED; alternatively, predefined seeds can be loaded by calling suboutine RAND0 (see the rita.f source file). The random-number generator can be changed by merely replacing the FUNCTION RAND; 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

• In the simulation routines, real and integer variables are declared as DOUBLE PRECISION and INTEGER*4, respectively. To prevent type mismatches, the statement

```
→  IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER*4 (I-N)
```

is used in all PENELOPE routines and in the example main programs.

• The main program *must* include the following declarations:

```
USE TRACK_mod
USE PENELOPE_mod;  USE PENERROR_mod
COMMON/RSEED/ISEED1,ISEED2 !  Random-number generator seeds.
```

The source listing of the modules TRACK_mod, PENELOPE_mod, and PENERROR_mod is at the head of the penelope.f file. This file must be compiled before any other program unit that uses those modules.

• The subroutines in PENELOPE and PENGEOM may print warning and error messages in unit 26, which is also the default output unit in the example main programs pencyl and penmain described in Section 7.2.

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 with any Fortran 95 compiler) is included in the PENELOPE distribution package.

7.1.3 Layout of a generic main program

The following Fortran listing illustrates the structure of a main program for simulation with quadric geometries (similar to the example program `penmain.f` described in Section 7.2.2). The program layout follows the flux diagram of Fig. 7.2. The simulation of interaction events is controlled by sequential calls to subroutines `START`, `JUMP` and `KNOCK`. The simplicity of the geometry operations is noteworthy: all operations are performed internally, and control from the main program reduces to successive calls to subroutines `LOCATE` and `STEP`.

Notice that in the case of electrons and positrons (`KPAR` = 1 and 3, respectively) the energy loss `DE` due to soft interactions along each segment is evaluated immediately after the call to subroutine `STEP`. This energy is assumed to be deposited uniformly along the segment and, to avoid double counting of soft energy losses, the energy deposited in the hinges (`ICOL` = 1) is ignored. This strategy has the advantage of ensuring that the average energy deposited in thin material bodies is evaluated correctly, even when the parameter `DSMAX` is larger than the characteristic thickness of the body. The program would also work if the soft energy losses are deposited at the hinges (*i.e.*, with the 6 lines ending with the string `CSDA` removed), but then `DSMAX` is required to be much less than the body thickness to prevent the charged particles from crossing a thin body without "seeing" it.

The comment lines beginning with "`cu`" indicate parts of the program that are specific to each experiment and have to be coded by the user. These include initialisation of the random number generator, definition of the source characteristics (*i.e.*, the specification of initial states of primary particles) and scoring of relevant quantities and distributions.

```
C...+....1....+....2....+....3....+....4....+....5....+....6....+....7..
C     PROGRAM MAIN
C
      USE PENELOPE_mod ! I/O of PENELOPE
      USE TRACK_mod ! Particle track variables
      USE PENGEOM_mod ! I/O of the geometry routines
      USE PENERROR_mod ! Initialization and run-time errors
C
      IMPLICIT DOUBLE PRECISION (A-H,O-Z), INTEGER*4 (I-N)
      CHARACTER PMFILE(MAXMAT)*20 ! Material data files
      DIMENSION PARINP(20),DSMAX(5000) ! Geometry parameters
      COMMON/RSEED/ISEED1,ISEED2 ! Random-number generator seeds
```

```
C
cu << Define counter arrays and initialise them to zero
cu    Set NTOT (total number of showers to be simulated) >>
C  ************  Initialisation of PENELOPE and PENGEOM.
cu << Set the values of the simulation parameters in module PENELOPE_mod
cu    and the seeds of the random-number generator in common RSEED >>
cu << Define EMAX (largest energy in the simulation) and NMAT (number
cu    of materials in the geometrical structure) >>
      PMFILE(1)=material_1.mat ! Material data files (input)
      PMFILE(2)=material_2.mat ! etc.
      INFO=3 ! Print detailed information on the transport models
      OPEN (UNIT=16,FILE='pen_init.dat') ! Output file
      CALL PEINIT(EMAX,NMAT,16,INFO,PMFILE) ! Initialises PENELOPE
      IF(IRETRN.NE.0) STOP ! Input or run-time code error
      CLOSE(UNIT=16)
C
      NPINP=0 ! All geometry parameters are defined from the input file
      OPEN(UNIT=17,FILE='my-geometry.geo')  ! Geometry definition file
      OPEN(UNIT=18,FILE=geometry.rep) ! Geometry report
      CALL GEOMIN(PARINP,NPINP,NMATG,NBOD,17,18) ! Initialises PENGEOM
      IF(IRETRN.NE.0) STOP ! Input or run-time code error
      CLOSE(UNIT=17)
      CLOSE(UNIT=18)
      IF(NMATG.GT.NMAT) STOP ! The geometry contains too many materials
cu << Define DSMAX(IBODY) for all bodies >>
C
C  ************  Simulation.
      ISEED1=1; ISEED2=1 ! Initialize the random number generator.
cu << Initialise global counters >>
      N=0
C
C  ++++++++++++++++++++++++  Generate a new shower
 10   N=N+1 ! New shower
cu << Set the initial state variables of the primary particle, possibly
cu    by random sampling from the source distribution. Define _ALL_ the
cu    parameters in module TRACK_mod >>
C  ****  Check if the trajectory intersects the material system
      CALL LOCATE ! Determines the body where the particle moves
      IF(MAT.EQ.0) THEN ! The particle is outside all material bodies
         CALL STEP(1.0D30,DSEF,NCROSS) ! Move the particle ahead
         IF(MAT.EQ.0) THEN ! The particle does not enter the system
           GOTO 10 ! Exit
         ENDIF
      ENDIF
```

```
      CALL CLEANS ! Cleans the secondary stack
C  ------------------------  Simulation of a new track
 20   CALL START ! Starts simulation in current medium
      IF(IRETRN.NE.0) STOP ! Run-time code error
 30   CALL JUMP(DSMAX(IBODY),DS) ! Determines the segment length
      CALL STEP(DS,DSEF,NCROSS) ! Moves particle to end of segment
      IF(KPAR.NE.2) THEN ! Soft energy loss                        ! CSDA
        IF(DESOFT.GT.0.0D0) DE=SSOFT*DSEF                          ! CSDA
      ENDIF                                                        ! CSDA
cu << Score relevant quantities >>
      IF(NCROSS.GT.0) THEN
        IF(KPAR.NE.2) E=E0SEGM-SSOFT*DSEF ! Soft energy loss correction
        IF(MAT.EQ.0) THEN ! The particle is outside the enclosure
          GOTO 40 ! Exit
        ENDIF
        GOTO 20
      ENDIF
      CALL KNOCK(DE,ICOL) ! Simulates the interaction event
      IF(IRETRN.NE.0) STOP ! Run-time code error
      IF(KPAR.NE.2) THEN ! Ignore energy loss in the hinge         ! CSDA
        IF(ICOL.EQ.1) DE=0.0D0                                     ! CSDA
      ENDIF                                                        ! CSDA
cu << Score relevant quantities >>
      IF(E.LT.EABS(KPAR,MAT)) THEN ! The particle is absorbed
        GOTO 40 ! Exit
      ENDIF
      GOTO 30
C  ------------------------  The simulation of the track ends here
 40   CONTINUE
cu << Score relevant quantities >>
C  ****  Any secondary left?
      CALL SECPAR(LEFT)
      IF(LEFT.GT.0) THEN
cu << The secondary particle extracts energy from the site; modify
cu    deposited energy counters accordingly >>
        GOTO 20
      ENDIF
C  ++++++++++++++++++++++++  The simulation of the shower ends here
C
      IF(N.LT.NTOT) GOTO 10
cu << Calculate final averages and write results in output files >>
      END
C...+....1....+....2....+....3....+....4....+....5....+....6....+....7..
```

7.1.4 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, interaction forcing, and Woodcock's delta scattering.

These variance-reduction methods are designed to speed up the calculation of local quantities (*e.g.*, average deposited energy or average particle fluence) that are obtained by scoring contributions of individual interaction events. Other quantities, such as the distribution of energy deposited in a body (which, in the case of a scintillation detector, may be transformed into a pulse-height spectrum) are determined by contributions of complete showers, and require strict control of shower evolution through the whole geometry.

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.

A simple variance-reduction technique, which can be applied in the main program, is the *splitting of x rays* (and of Auger electrons) released through the relaxation of atoms with vacancies in inner shells. As this radiation is emitted isotropically, it is advantageous to split each x ray into a number `NSPLIT` of photons with the same energy and with independent random directions sampled from the isotropic distribution. The main program `penmain` (Section 7.2.2) implements this technique, which is found to be effective in calculations of the angular distribution of photons from x-ray sources.

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 in the secondary stack. The current particle continues with a reduced statistical weight.

`SUBROUTINE VRR(PSURV)` and `SUBROUTINE VKILL(PKILL)`
These subroutines apply the Russian roulette technique. The particle is killed with probability `PKILL = 1-PSURV` ($0 <$ `PKILL` < 1); if it survives, its weight is increased by a factor `1/(1-PKILL)`.

`SUBROUTINE JUMPF(DSMAX,DS)` and `SUBROUTINE KNOCKF(DE,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 cross section $\sigma_{\texttt{ICOL}}$ of the desired interaction by a larger value FORCE(.)$\times\sigma_{\texttt{ICOL}}$. The forcing factors FORCE(IBODY,KPAR,ICOL) [for the bodies in the material structure where interaction forcing is applied, and for the types of particles and the kinds of interactions (see Table 7.5) that are forced] have to be specified by the user in the main program. These factors are transferred through the Fortran module PENVARED_mod (Table 7.7). Processes that do not involve a net transfer of energy to the material (namely, elastic scattering of electrons and positrons, and Rayleigh scattering of photons) cannot be forced; the subroutines ignore any forcing factor that may have been defined for these processes.

Forcing factors must be larger than, or equal to unity; obviously, the value FORCE(.) = 1 means "no forcing". To keep the simulation unbiased, interactions are allowed to affect the state of the projectile only with probability WFORCE = 1/FORCE(.), and, at the same time, the deposited energy and the secondary particles generated in the interactions are assigned a weight smaller than that of the projectile by a factor equal to WFORCE. The quantity DEA in module PENVARED_mod is the energy loss of the projectile in the last simulated event (*i.e.*, the value generated by the interaction sampling routines for a projectile with unit weight).

Table 7.7: Source listing of the Fortran module PENVARED_mod.

```
      MODULE PENVARED_mod
C
C  ****  Variance-reduction parameters.
C
      SAVE  ! Saves all items in the module.
C  ----  Parameter values defined for each body. NBV is the maximum
C        number of bodies. When using PENGEOM, NBV must have the same
C        value as the parameter NB in module PENGEOM_mod.
      INTEGER*4, PARAMETER :: NBV=5000
C  ----  Forcing factors, FORCE(IBODY,KPAR,ICOL).
      DOUBLE PRECISION, DIMENSION (NBV,3,8) :: FORCE=1.0D0
      DOUBLE PRECISION :: WFORCE
C  ----  Bremsstrahlung splitting numbers, IBRSPL(IBODY).
      INTEGER*4, DIMENSION (NBV) :: IBRSPL=1
C  ----  Energy deposited in the last event (analogue simulation).
      DOUBLE PRECISION :: DEA
C
      END MODULE PENVARED_mod
```

To apply interaction forcing, the main program must call subroutines JUMPF and KNOCKF instead of the usual subroutines JUMP and KNOCK. In addition, subroutine START *must be called* before starting a track and after each interface crossing, even for photons. The output argument DE of subroutine KNOCKF is the net energy deposited in the event, assuming that the particle has unit weight, *i.e.*, DE = DEA $*$ WFORCE.

Combined with interaction forcing, subroutine KNOCKF can also apply *bremsstrahlung splitting* in selected bodies. This technique consists of sampling a "normal" photon in each radiative event, and then splitting it into a number IBRSPL (> 1) of photons, all of them with the energy and polar emission angle of the "normal" photon, weight equal to 1/IBRSPL times the "normal" weight, and azimuthal emission angles sampled uniformly in the interval from zero to 2π. This method is computationally simple (a single DO loop) and very effective in increasing the number of photons emitted by electrons and positrons, because each "split" photon is obtained by only applying a rotation to the direction of the normal photon. The bremsstrahlung splitting number IBRSPL(IBODY) in each body must be specified by the user in the main program. This number is transferred through the Fortran module PENVARED_mod.

SUBROUTINE JUMPW(DSMAX,DS) and SUBROUTINE KNOCKW(DE,ICOL)

These two subroutines implement Woodcock's delta-scattering method for photons (see Section 1.6.3). We recall that this method will improve the efficiency only for those geometries where locating a particle (*i.e.*, finding the material at its current position) is faster than normal tracking. The output argument DE of subroutine KNOCKW is the net energy deposited in the event, assuming the weight of the particle equals unity; it includes a weight correction such that no special action needs to be taken in the main program.

To apply the delta-scattering method, the MAIN program must call subroutines JUMPW and KNOCKW instead of the usual subroutines JUMP and KNOCK. Furthermore, subroutine START *must* be called before starting a photon track and after each interface crossing.

Woodcock's method can be applied together with interaction forcing. In this case, the interaction forcing parameters must be loaded before the first call to subroutine JUMPW. Notice that, when the geometry is described with PENGEOM, the delta-scattering method is incompatible with the use of impact detectors, which require strict control of interface crossings.

Usually, the variance-reduction parameters [*i.e.*, the splitting number NSPLIT, the killing probability PKILL, and the forcing factors FORCE(.)] 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, PKILL or PSURV, and IBRSPL(.) can be modified at any time during the simulation of a particle. To ensure consistency of the mixed simulation scheme, the values of the forcing factors FORCE(.) for electrons and positrons should only be changed immediately after a call to subroutine START.

In spite of the fact that 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 `NMS` = 1,000 particles, which 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 is filled up, 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; this can be automatically enforced by using a weight window with its lower limit higher than `WFORCE`. If needed, the size of the stack can be increased by editing the file `penelope.f` and changing the value of the parameter `NMS` in the Fortran module `PENELOPE_mod`.

It should be mentioned that, when interaction forcing is applied, energy is conserved only on average because the energy deposited in the medium is, in general, different from the energy lost by the projectile. This implies that the spectrum of deposited energy in a given volume (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.

Range rejection (*i.e.*, killing particles that are far away from the region of interest) should never be used with photons. In the case of charged particles, this technique may be applied automatically by considering the absorption energies `EABS(KPAR)` as local variables. Because simulation routines work properly when they are passed an increased value of `EABS(KPAR)` (*i.e.*, they discontinue the simulation of particles with energy below that value), we may use this parameter to terminate tracking charged particles that move in bodies that are too far from the detector to produce an appreciable contribution. The example main programs `pencyl` and `penmain` implement this strategy to reduce the simulation work in regions of lesser interest.

7.2 Examples of main programs

The distribution package of PENELOPE includes two examples of main programs: `pencyl` (which simulates electron-photon transport in cylindrical geometries) and `penmain` (for generic quadric geometries). By default, they describe sources that emit a single type of particles. `penmain` offers the option of considering radioactive sources with a single emitting nuclide. With moderate editing work, the programs can be adapted to simulate multi-particle sources with discrete or continuous energy spectra.

`Pencyl` and `penmain` provide fairly detailed information, but their scoring capabilities are limited. It should be borne in mind that these 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 accurately describe 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 (see the flow diagrams in Figs. 7.1 and 7.2, and the program layout in Section 7.1.3).

Although the sources and geometries considered in `pencyl` can also be simulated by `penmain`, the relative simplicity of cylindrical geometries gives some practical advantages to `pencyl`. For instance, with cylindrical geometries it is easy to apply the variance reduction method of delta scattering of photons (Section 1.6.3), which may not be effective for complicated quadric geometries. `pencyl` provides slightly more detailed results than `penmain`, such as two-dimensional maps of absorbed dose and deposited charge covering selected bodies of the material structure. It also gives the average numbers of the various types of interactions undergone by primary particles; this feature is useful to analyze the influence of the various user parameters on the efficiency of the simulations.

The two example main programs operate in a similar way. They both read data from a corresponding input file and output the results in a number of files with fixed names[3]. The input files of `pencyl` and `penmain` have similar structures and formats; they are described in detail in the following Sections[4]. 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 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).

[3]Warning: The programs overwrite older output files that are left in the working directory. You should save all result files on a separate directory before re-running a program.

[4]Although the input files of the two 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.

The main programs `pencyl` and `penmain` generate 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. Notice that all simulated distributions are normalized "per shower" (*i.e.*, per primary particle or nuclear decay) and, consequently, the integral of a distribution equals the probability that a shower contributes to the relevant counters. 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, `pencyl` and `penmain` 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 disk; otherwise, the results will be lost. When run interactively, the programs write the number of simulated showers on the screen just after closing all output files.

Both `pencyl` and `penmain` are designed to be run on a single processor. The distribution package includes two auxiliary programs `pencyl-sum` and `penmain-sum`, which add the values of the counters in a number of dump files generated from independent runs of `pencyl` and `penmain`, respectively. These programs allow the user to run the same problem on several processors, using different seeds of the random number generator (see Section 1.2.1), and then combine the results to produce a single set of output files, with accumulated statistics. They work as effective "poor man's" parallelisation devices.

Previous versions of `pencyl` and `penmain` 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, 2003).

7.2.1 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. 7.3).

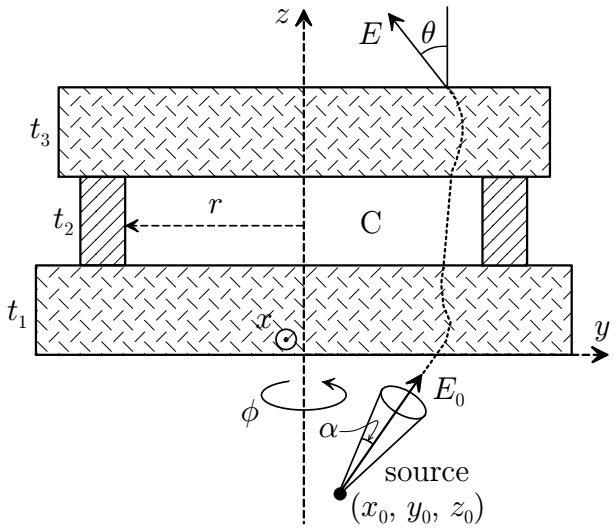

Figure 7.3: An example of cylindrical geometry, a thick-walled cavity (C) with a point off-axis source (conical beam). In this case, the material structure is symmetrical about the z-axis, but the radiation flux and other three-dimensional quantities (*e.g.*, absorbed dose and deposited charge distributions) depend on the azimuthal angle ϕ.

Primary particles of a given type, `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 particles is sampled uniformly within a circle of the unit sphere (conical beam), or within a "rectangular" window on the unit sphere (rectangular beam).

The program can simulate two different types of sources:
a) An external source with uniform activity over a cylindrical volume, which is defined separately from the geometry of the material system.
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. If there are several active bodies, they 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.

The source file `pencyl.f` includes a stand-alone 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 6.6) 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 or bodies intersected by the window plane. It is useful for debugging the geometry definition.

The program `pencyl` delivers detailed information on 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). To generate more specific information, the user can define energy-deposition detectors. Each detector consists of a set of active (non-void) bodies, which must have been previously defined as parts of the geometry. The output spectrum of an energy-deposition detector is the distribution of absorbed energy (per shower) in the active bodies.

Relevant input data are written in a file named `pencyl.dat`. The output file `pencyl-res.dat` contains a report on the global simulation and some partial results. The calculated continuous distributions (histograms) are written in separate files, whose names have the extension "`.dat`". These files are in a format suited for direct visualisation with GNUPLOT (version 5.0). `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.

7.2.1.1 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 are not part of the input file).

```
....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
TITLE  Title of the job, up to 65 characters.
       . (the dot prevents editors from removing trailing blanks)
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.
       The labels KL,KC denote the KC-th cylinder in the KL-th layer.
       .
       >>>>>>>> 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]
```

```
SGPOL  SP1,SP2,SP3          [Stokes parameters for polarised photons]
SEXTND KL,KC,RELAC    [Extended source in KL,KC, rel. activity dens.]
STHICK STHICK                                          [Source height]
SRADII SRIN,SROUT                      [Source inner and outer radii]
SPOSIT SX0,SY0,SZ0                  [Coordinates of the source centre]
SCONE  THETA,PHI,ALPHA                   [Conical beam; angles in deg]
SRECTA THETAL,THETAU,PHIL,PHIU      [Rectangular beam; angles in deg]
       .
       >>>>>>>> Material data and simulation parameters.
                Up to MAXMAT materials; 2 lines for each material.
MFNAME mat-filename.ext               [Material file, up to 20 chars]
MSIMPA EABS(1:3),C1,C2,WCC,WCR              [EABS(1:3),C1,C2,WCC,WCR]
       .
       >>>>>>>> Local maximum step lengths and absorption energies.
DSMAX  KL,KC,DSMAX(KL,KC)         [Maximum step length in body KL,KC]
EABSB  KL,KC,EABSB(1:3,KL,KC)        [Local EABSB(1:3) in body KL,KC]
       .
       >>>>>>>> Interaction forcing.   [KL,KC,KPAR,ICOL,FORCER,WL,WH]
IFORCE KL,KC,KPAR,ICOL,FORCER,WLOW,WHIG         [Interaction forcing]
       .
       >>>>>>>> Bremsstrahlung splitting.
IBRSPL KL,KC,IBRSPL                              [KL,KC,splitting factor]
       .
       >>>>>>>> X-ray splitting.
IXRSPL KL,KC,IXRSPL                              [KL,KC,splitting factor]
       .
       >>>>>>>> Woodcock's delta-scattering method for photons.
IWOODC                                  [Delta scattering is turned on]
       .
       >>>>>>>> Counter array dimensions and PDF ranges.
NBE    EL,EU,NBE                        [Energy window and no. of bins]
NBANGL NBTH,NBPH           [No. of bins for the angles THETA and 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]
       .
       >>>>>>>> Particle positions at the lower and upper planes.
EMERGP RADM,NBRE          [Radius of the scoring region, no. of bins]
       .
       >>>>>>>> Energy-deposition detectors (up to 25).
ENDETC EL,EU,NBE                        [Energy window and no. of bins]
EDSPC  spc-enddet-##.dat        [Output spectrum file name, 20 chars]
EDBODY KL,KC                                          [Active cylinder]
       .
```

```
       >>>>>>>> Dose and charge distributions.
DOSE2D KL,KC,NZ,NR     [Tally distributions in KL,KC with NZ,NR bins]
       .
       >>>>>>>> Job properties.
RESUME dump1.dmp               [Resume from this dump file, 20 chars]
DUMPTO dump2.dmp                  [Generate this dump file, 20 chars]
DUMPP  DUMPP                                   [Dumping period, in sec]
       .
RSEED  ISEED1,ISEED2           [Seeds of the random-number generator]
NSIMSH DSHN                     [Desired number of simulated showers]
TIME   TIMEA                       [Allotted simulation time, in sec]
END                                  [Ends the reading of input data]
....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
```

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 65 characters).
`-- Default:` none (the input file must start with this line)

The `TITLE` string is used to mark dump files. To prevent improper use of incorrect resuming files (generated, *e.g.*, from runs with different conditions), 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 material and the inner and outer radii of the various concentric rings in the layer; empty layers are disregarded. No blank lines are allowed in the geometry definition list.

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) which is analogous to the two-dimensional viewer GVIEW2D of quadric geometries (see Section 6.6). 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␣ ... Type 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 a 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 = 1,000 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

SGPOL␣ ... This line activates the simulation of polarisation effects in the scattering of photons, *i.e.*, it sets IPOL=1 (electrons and positrons are assumed to be unpolarised). SP1, SP2, SP3 are the Stokes parameters of primary photons, which define the degrees of linear polarisation at 45 deg azimuth, of circular polarisation, and of linear polarisation at zero azimuth, respectively (see Appendix C). It is assumed that secondary photons are emitted with null polarisation (SP1=SP2=SP3=0.0).
-- 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=SZ0=0.0

SCONE␣ ... Conical source beam (the initial direction of primary particles is sampled uniformly within the solid angle of a cone). Polar and azimuthal angles of the beam axis direction, THETA and PHI, and angular semiaperture, ALPHA, in deg.
-- Defaults: THETA=0.0, PHI=0.0, ALPHA=0.0

The case ALPHA=0.0 defines a monodirectional source, and ALPHA=180.0 corresponds to an isotropic source.

`SRECTA` ... Rectangular source beam (the initial direction of primary particles is sampled uniformly within the solid angle of a rectangular window on the unit sphere). Limiting polar and azimuthal angles of the source beam window, `(THETAL, THETAU)`×`(PHIL,PHIU)`, in deg.
`-- Defaults: THETAL=0.0, THETAU=0.0, PHIL=0.0, PHIU=0.0`

The case `THETAL=THETAU`, `PHIL=PHIU` defines a monodirectional source. To define an isotropic source, set `THETAL=0.0`, `THETAU=180.0`, `PHIL=0.0` and `PHIU=360.0`.

Note that the default source is a pencil beam that moves upwards along the z-axis.

• Material data and simulation parameters

`MFNAME` ... Name of a PENELOPE input material data file (up to 20 characters). This file must be generated in advance by running the program `material`.
`-- Default:` none

`MSIMPA` ... Values of the simulation parameters for this material: absorption energies, `EABS(1:3,M)`, elastic scattering parameters, `C1(M)` and `C2(M)`, and cutoff energy losses for inelastic collisions and bremsstrahlung emission, `WCC(M)` and `WCR(M)`.
`-- Defaults: EABS(1,M)=EABS(3,M)=0.01*EMAX, EABS(2,M)=0.001*EMAX`
`C1(M)=C2(M)=0.1, WCC(M)=EABS(1,M), WCR(M)=EABS(2,M)`
`EMAX` is the highest energy of all particles found in the simulation. It is determined by the source energies.

For each material we must provide the name of the material-data file and a set of simulation parameters. The label (material number) assigned by PENELOPE to each material is determined by the ordering of the material list in the input file. That is, the first, second, ... materials are assigned the labels 1, 2, ... These labels are also used in the geometry definition. The material file may contain multiple declarations of the same material, *e.g.*, to specify different sets of simulation parameters in different bodies with the same composition.

The original programs in the distribution package allow up to 10 materials. This number can be increased by changing the value of the parameter `MAXMAT` in the Fortran module `PENELOPE_mod` (see Table 7.4), which is at the heading of the `penelope.f` source file.

• Local maximum step lengths and absorption energies

`DSMAX␣` ... Maximum step length `DSMAX(KL,KC)` of electrons and positrons in cylinder `KL`,`KC`. This parameter is important only for thin bodies; it should be given a value of the order of one tenth of the cylinder thickness or less.
`-- Default: DSMAX=0.1` times the cylinder thickness

`EABSB␣` ... Local absorption energies `EABSB(KPAR,KL,KC)` of particles of type `KPAR` in body `KL`,`KC`. These values must be larger than `EABS(KPAR,M)`, where `M` is the

material of body KL,KC. When the particle is moving within body KL,KC, the absorption energy EABS(KPAR,M) is temporarily set equal to EABSB(KPAR,KL,KC). Thus, the simulation of the particle history is discontinued when the energy becomes less than EABSB(KPAR,KL,KC). This feature can be used, *e.g.*, to reduce the simulation work in regions of lesser interest.
-- Defaults: EABSB(KPAR,KL,KC)=EABS(KPAR,M) (no action)

- **Variance reduction techniques**

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 (Section 1.6.1). The values WLOW, WHIG are the limits of the weight window where interaction forcing is applied (see Section 7.1.3). When several interaction mechanisms are forced in the same body, the effective weight window is set equal to the intersection of the windows for these mechanisms.
-- Default: no interaction forcing

If the mean free path for real interactions of type ICOL is MFP, the program will simulate interactions of this type (real or forced) with an effective mean free path equal to MFP/FORCER.

The real effect of interaction forcing on the efficiency is not easy to predict. Please, do tentative runs with different FORCER values and check the efficiency gain (or loss!).

TRICK: a negative input value of FORCER, $-\mathcal{F}$, is interpreted as follows,
KPAR = 1 or 3: electrons or positrons with energy E = EMAX should interact, on average and approximately, $\mathcal{F}$ times in the course of their slowing down to rest, *i.e.*, in a path length equal to their CSDA range.
KPAR = 2: the mean free path of photons with energy E = EMAX between interactions of type ICOL is set equal to $\mathcal{F}$.
This is very useful, *e.g.*, to generate x-ray spectra.

IBRSPL ... Activates bremsstrahlung splitting in cylinder KC of layer KL for electrons and positrons with weights in the window (WLOW,WHIG) where interaction forcing is applied. The integer IBRSPL is the splitting factor.
-- Default: no bremsstrahlung splitting

Note that bremsstrahlung splitting is applied in combination with interaction forcing and, consequently, it is activated only in those bodies where interaction forcing is active.

IXRSPL ... Splitting of characteristic x rays emitted in cylinder KC of layer KL, from any element. Each unsplit x ray with ILB(2)=2 (*i.e.*, of the second generation) when extracted from the secondary stack is split into IXRSPL quanta. The new, lighter, quanta are assigned random directions.
-- Default: no x-ray splitting

`IWOODC` ... Activates Woodcock's delta-scattering method for photons (see Section 7.1.4). The method is very effective for the calculation of dose, preferably combined with interaction forcing.
`-- Default:` off

• Counter array dimensions and tallied ranges

`NBE␣␣␣` ... Limits, `EL` and `EU`, of the interval where energy distributions of emerging particles are tallied and number of energy bins, `NBE` ($\leq 1,000$).
`-- Defaults: EL=0.0, EMAX=EMAX, NBE=100`

Although the number of bins must be positive, the program also admits negative values of `NBE`. When `NBE` is positive, energy bins have uniform width equal to `DE=(EU-EL)/NBE`. When the input value of `NBE` is negative, the output distribution consists of |`NBE`| bins whose widths increase geometrically with the energy, *i.e.*, the energy bins have uniform width on a logarithmic scale.

`NBANGL` ... Number of bins for the polar angle θ and the azimuthal angle ϕ, respectively, `NBTH` and `NBPH` (≤ 180).
`-- Defaults: NBTH=90, NBPH=1` (azimuthal average)

If `NBTH` is positive, polar-angle bins have uniform width, `DTH=180./NBTHE`. When `NBTH` is negative, the bin width increases geometrically with `THETA`, *i.e.*, the bins have uniform width on a logarithmic scale.

NOTE: In the output files, the terms "upbound" and "downbound" are used to denote particles that leave the material system moving upwards (`W`>0) and downwards (`W`<0), respectively[5].

`NBZ␣␣␣` ... Number of bins for the z-coordinate. Less than or equal to 200.
`-- Default: NBZ=100`

`NBR␣␣␣` ... Number of bins for the radial variable, $r = (x^2 + y^2)^{1/2}$. Less than or equal to 200.
`-- Default: NBR=100`

`NBTL␣␣` ... Limits of the interval where track-length distributions of primary particles are tallied. Number of track-length bins, `NBTL` ≤ 200.
`-- Defaults: TLMIN=0.0, TLMAX=5*RANGE(EMAX,KPARP,1), NBTL=100`

• Particle positions at the lower and upper planes

`EMERGP` ... The program generates the PDFs of the coordinates (`X`,`Y`) of points where trajectories of emerging particles intersect the upper and lower planes of the geometry definition (upbound and downbound particles, respectively), and the PDF of the radial distance of the intersections, `R=SQRT(X*X+Y*Y)`.

[5]In previous versions of the programs we used the respective terms "transmitted" and "backscattered", which did cause some confusion when primary particles were incident from above.

RADM is the radius of the scoring region, and NBRE is the number of bins of the radial distributions. The (X,Y) distributions are tallied on a square of side 2*RADM, with (2*NBRE)^2 bins.
-- Default: None

• **Energy-deposition detectors**

ENDETC ... Starts the definition of a new energy-deposition detector. Up to 25 different detectors can be considered. EL and EU are the lower and upper limits of the energy window covered by the detector.

NBE is the number of bins in the output energy spectrum ($\leq$1,000). If NBE is positive, energy bins have uniform width, DE=(EU-EL)/NBE. When NBE is negative, the bin width increases geometrically with the energy, *i.e.*, the energy bins have uniform width in a logarithmic scale.

WARNING: The energy-deposition spectrum may be strongly biased when interaction forcing or x-ray splitting are applied, even outside the detector bodies.

EDSPC␣ ... Name of the output spectrum file (up to 20 characters).
-- Default: spc-enddet-##.dat

EDBODY ... Active body KL,KC of the detector. One line for each active body.
-- Default: none

NOTE: A body cannot be part of more than one energy-deposition detector.

• **Absorbed dose and charge 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 200. Up to ten different bodies can be selected, a DOSE2D line is required for each body.
-- Default: off

• **Job properties**

RESUME ... The program will read the dump file dump1.dmp (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 dump2.dmp (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 dump2.dmp already exists, it is overwritten.

`DUMPP␣ ...` Generate a dump file and write the simulation results in the output files every `DUMPP` seconds. The name of the dump file is the one declared in the preceding `DUMPTO` line; when a `DUMPTO` line is not included, the dump file has the default name `dump.dmp`.
`-- Default: DUMPP=1.0E15`

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. Because the simulation work is halted while results are being written the dumping period `DUMPP` should not be too small.

`RSEED␣ ...` Seeds of the-random number generator.
`-- Default: ISEED1=1; ISEED2=1`

When `ISEED1` is equal to a negative integer, `-N`, the seeds are set by calling subroutine `RANDO(N)` with the input argument equal to `N`. This ensures that sequences of random numbers used in different runs of the program (with different values of `N`) are truly independent.

`NSIMSH ...` Desired number of simulated showers.
`-- Default: DSHN=2.0E9`

`TIME␣␣ ...` Allotted simulation time, in sec.
`-- Default: TIMEA=2.0E9`

`END␣␣␣ ...` Ends the reading of the input file. This line is needed only when the lines `NSIMH` and `TIME␣␣` are missing.

7.2.1.2 Example

The example of an 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"×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. Note that 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 $\theta = 0$ (*i.e.*, upwards) and impinge normally on the lower surface of the detector. The geometry is shown schematically in the insets of Fig. 7.4, which displays two of the distributions generated by `pencyl`.

• Example input file of the `pencyl` program (`cyld.in`).

```
....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
TITLE  NaI detector with Al cover and Al2O3 reflecting foil
GSTART >>>>>>>> Beginning of the geometry definition list.
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   <<<<<<<< End of the geometry definition list.
       >>>>>>>> Source definition.
SKPAR  2        [Primary particles: 1=electron, 2=photon, 3=positron]
SENERG 1.25e6           [Initial energy (monoenergetic sources only)]
SPOSIT 0 0 -10.0                   [Coordinates of the source centre]
SCONE  0 0 0                            [Conical beam; angles in deg]
       >>>>>>>> Material data and simulation parameters.
MFNAME NaI.mat                         [Material file, up to 20 chars]
MSIMPA 5e4 5e3 5e4 0.1 0.1 2e3 2e3          [EABS(1:3),C1,C2,WCC,WCR]
MFNAME Al2O3.mat                       [Material file, up to 20 chars]
MSIMPA 5e4 5e3 5e4 0.1 0.1 2e3 2e3          [EABS(1:3),C1,C2,WCC,WCR]
MFNAME Al.mat                          [Material file, up to 20 chars]
MSIMPA 5e4 5e3 5e4 0.1 0.1 2e3 2e3          [EABS(1:3),C1,C2,WCC,WCR]
       >>>>>>>> Energy-deposition detectors (up to 25).
ENDETC 0 1.3e6 130                     [Energy window and no. of bins]
EDBODY 3 1                                          [Active cylinder]
       >>>>>>>> Dose and charge distributions.
DOSE2D 3 1 50 50       [Tally distributions in KL,KC with NZ,NR bins]
       >>>>>>>> Job properties
RESUME dump.dmp                [Resume from this dump file, 20 chars]
DUMPTO dump.dmp                   [Generate this dump file, 20 chars]
DUMPP  60                                          [Dumping period, in sec]
NSIMSH 1.0e7                    [Desired number of simulated showers]
TIME   1.0e9                       [Allotted simulation time, in sec]
....+....1....+....2....+....3....+....4....+....5....+....6....+....7..
```

Figure 7.4 displays results obtained from the simulation of 10 million random showers, which took 1 hour on an Intel i7-3520M processor at 2.90 GHz. The uncertainty bars represent statistical uncertainties (3σ), which are pretty small in this case. The upper plot 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 downbound photons, *i.e.*, photons that emerge from the system pointing downwards, with $\mathtt{W} = \cos\theta < 0$. 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 $\sim$ 511 keV x rays resulting from positron annihilation (A) and escape of $\sim$ 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).

7.2.2 Program `penmain`

This is a generic main program that performs simulations of coupled 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 PENGEOM (see Chapter 6), which is able to handle complicated quadric 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 "canned" program, `penmain` is flexible enough to solve a broad class of practical problems.

In the default mode, `penmain` assumes that primary particles of a given type are emitted from a point or an extended source, either with fixed energy or with a specified energy spectrum, defined as a histogram or as a set of discrete lines. The initial direction of the primary particles is sampled uniformly within a circle of the unit sphere (conical beam), or within a "rectangular" window on the unit sphere (rectangular beam). Alternatively, the program can read the initial state variables of "primary" particles from pre-calculated phase-space files. This option is useful for splitting the simulation of complex problems into several consecutive stages. The program can also simulate radioactive sources with a single radioisotope by using the approach described by García-Toraño *et al.* (2017), which utilizes nuclear and atomic relaxation data from the NUCLEIDE evaluated data files (Bé *et al.*, 2011). For these sources the user must provide an additional input file with information on the nuclear decay process, which can be downloaded from the NUCLEIDE web site.

`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 6.5) and energy-deposition detectors. Each detector consists of a set of active (non-void) bodies, which must have been previously defined as parts of the geometry. The output spectrum from an impact detector is the energy distribution of particles that have 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 one of the defined impact detectors, the program can generate a *phase-space file* where the state variables of particles at the detector entrance are recorded. It is also possible to tally the average distribution of fluence with respect to the energy of particles in the volume of each impact detector. The output spectrum of an energy-deposition detector is the distribution of absorbed energy (per shower) in the active bodies. The user can also define "outer"

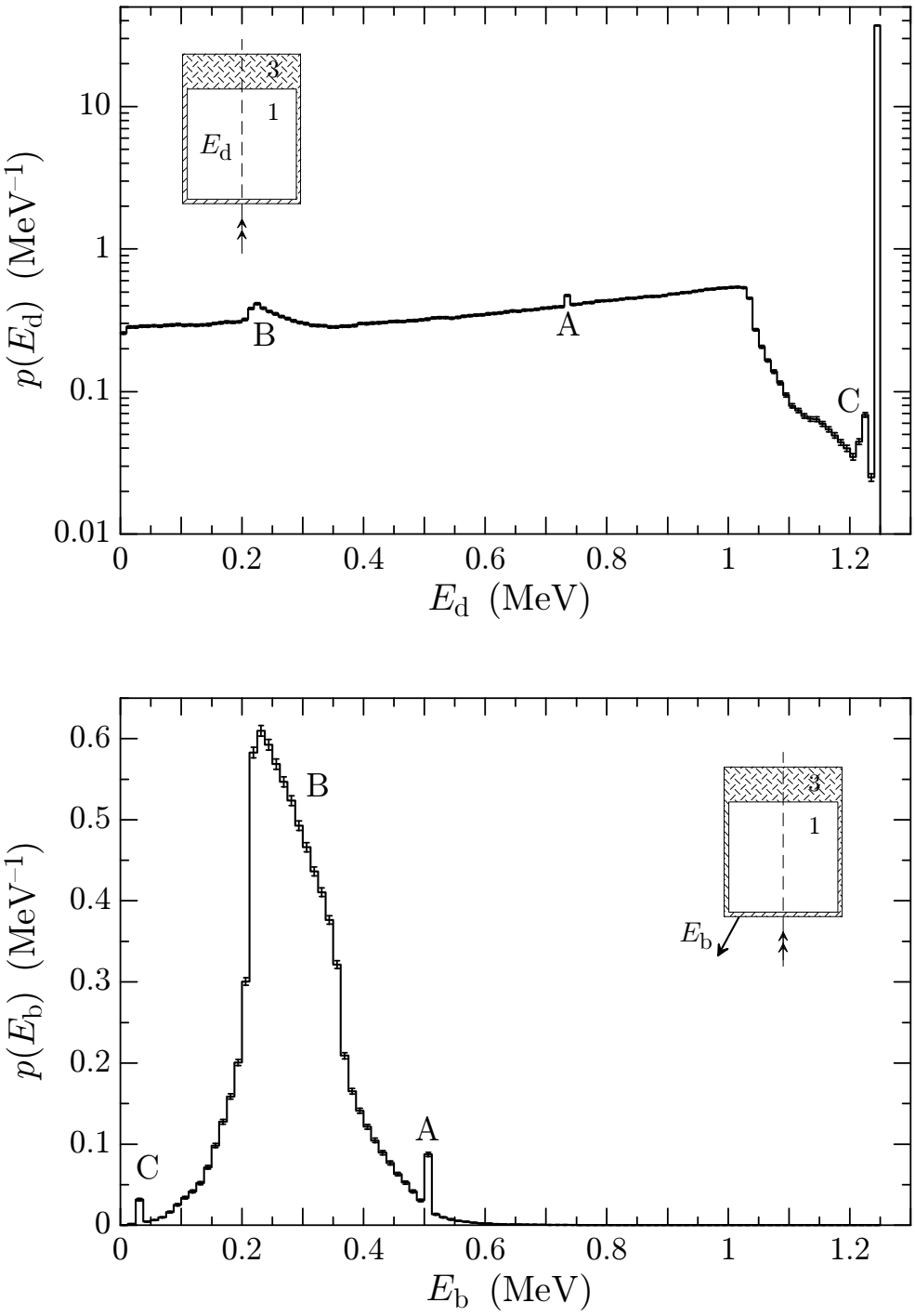

Figure 7.4: Partial results from `pencyl` for the NaI photon detector described in the text (input file `cyld.in`). Top: distribution of energy deposited in the NaI crystal (`MAT=1`). Bottom: energy distribution of backscattered (downbound) photons.

angular detectors, which collect particles that leave the enclosure with directions within a "rectangular window" on the unit sphere, $(\theta_1, \theta_2) \times (\phi_1, \phi_2)$. The output spectrum of an angular detector is the energy distribution of particles that emerge within the acceptance solid angle of the detector. As mentioned above, all simulated distributions are normalized "per shower", *i.e.*, per primary particle or nuclear decay. Hence, the integral of a detector's spectrum equals the "efficiency" of the detector, that is, the probability that an individual shower contributes to the spectrum. The program also offers the option of tallying a dose map on an orthogonal mesh defined by the user or, for problems with the corresponding symmetry, on a cylindrical or spherical mesh.

Relevant input data are written in a file named `penmain.dat`. The output file `penmain-res.dat` contains a report on the global simulation and some partial results. The calculated continuous distributions (histograms) are written in separate files, whose names have the extension "`.dat`". These files are in a format suited for direct visualisation with GNUPLOT (version 5.0).

7.2.2.1 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 are not part of the input file).

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

```
       >>>>>>>> Material data and simulation parameters.
                Up to MAXMAT materials; 2 lines for each material.
MFNAME mat-filename.ext                [Material file, up to 20 chars]
MSIMPA EABS(1:3),C1,C2,WCC,WCR               [EABS(1:3),C1,C2,WCC,WCR]
       .
       >>>>>>>> Geometry and local simulation parameters.
GEOMFN geo-filename.ext                [Geometry file, up to 20 chars]
PARINP IP,PARINP(IP)                xxxxxxxxxx [Replacement parameter]
DSMAX  KB,DSMAX(KB)               [KB, maximum step length in body KB]
EABSB  KB,EABSB(1:3,KB)   [KB, local absorption energies, EABSB(1:3)]
       .
       >>>>>>>> Interaction forcing.
IFORCE KB,KPAR,ICOL,FORCER,WLOW,WHIG  [KB,KPAR,ICOL,FORCER,WLOW,WHIG]
       .
       >>>>>>>> Bremsstrahlung splitting.
IBRSPL KB,IBRSPL                                  [KB,splitting factor]
       .
       >>>>>>>> X-ray splitting.
IXRSPL KB,IXRSPL                                  [KB,splitting factor]
       .
       >>>>>>>> Emerging particles. Energy and angular distributions.
NBE    EL,EU,NBE                       [Energy window and no. of bins]
NBANGL NBTH,NBPH           [No. of bins for the angles THETA and PHI]
       .
       >>>>>>>> Impact detectors (up to 25 different detectors).
       IPSF=0; no psf is created.
       IPSF=1; a psf is created (for only one detector).
       IDCUT=0; tracking is discontinued at the detector entrance.
       IDCUT=1; the detector does not affect the tracking.
       IDCUT=2; the detector does not affect tracking, the energy
                distribution of particle fluence (integrated over the
                volume of the detector) is calculated.
IMPDET EL,EU,NBE,IPSF,IDCUT      [E-window, no. of bins, IPSF, IDCUT]
IDSPC  spc-impdet-##.dat               [Spectrum file name, 20 chars]
IDPSF  psf-impdet.dat                [Phase-space file name, 20 chars]
IDFLNC fln-impdet-##.dat       [Fluence spectrum file name, 20 chars]
IDAGEL AGEL,AGEU,NAGE                  [Age interval and no. of bins]
IDAGEF age-impdet-##.dat       [Age-distribution file name, 20 chars]
IDBODY KB                                             [Active body]
IDKPAR KPAR                             [Type of detected particles]
       .
       >>>>>>>> Outer angular detectors (up to 25).
ANGDET THETA1,THETA2,PHI1,PHI2         [Angular window, angles in deg]
ADENER EL,EU,NBE                       [Energy window and no. of bins]
```

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

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 65 characters).
`-- Default:` none (the input file must start with this line)

The `TITLE` string is used to mark dump files. To prevent improper use of incorrect resuming files (generated, *e.g.*, from runs with different conditions), 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␣ ...` Type of primary particle (1 = electrons, 2 = photons, or 3 = positrons).
`-- Default: KPARP=1`

The input value `KPARP=0` activates the simulation of a radioactive source (see García-Toraño *et al.*, 2017), and requires to load the NUCLEIDE decay data file of the active radionuclide. That is, the input file *must* contain the following line.

SRNUCL ... this option allows simulating radioactive sources (one nuclide at a time) using nuclear decay characteristics from the NUCLEIDE database, which contains evaluated data for more than 200 radionuclides. penmain reads the decay data of the considered nuclide from the text file nuclide.nuc in the specific format "PenNuc", which can be downloaded from the NUCLEIDE web site (www.nucleide.org/DDEP_WG/DDEPdata.htm)

The NUCLEIDE PenNuc file nuclide.nuc, as well as the files pdrelax.p11 and pdatconf.p14 from the PENELOPE database must be placed in the PenNuc data directory. The relative path of this directory is defined in the module PENNUC_mod, at the head of the Fortran source file pennucm.f. In the distributed program the PenNuc data directory is defined to be the working directory.

Notice that the line SRNUCL implies that the initial directions of primary particles released by the decaying atom are sampled isotropically, irrespective of other source specifications.

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 = 1,000 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

SLINES ... The energy spectrum of the source consists of a number of discrete lines. Each SLINES line in the input file defines the energy (Ei) and the relative probability (Pi) of a line. Up to NSEM = 1,000 lines, in arbitrary order.
-- Default: none

SGPOL␣ ... This line activates the simulation of polarisation effects in the scattering of photons, *i.e.*, it sets IPOL=1. SP1, SP2, SP3 are the Stokes parameters of primary photons, which define the degrees of linear polarisation at 45 deg azimuth, of circular polarisation, and of linear polarisation at zero azimuth, respectively (see Appendix C). It is assumed that secondary photons are emitted with null polarisation (SP1=SP2=SP3=0).
-- Defaults: SP1=SP2=SP3=0.0

SPOSIT ... Coordinates (x_0, y_0, z_0) of the source centre.
-- Defaults: SX0=SY0=SZ0=0.0

SBOX␣␣ ... Extended source box. The source has uniform activity within the volume of a right prism centred at the point (SX0,SY0, SZ0) and whose sides have lengths SSX, SSY and SSZ.
-- Defaults: SSX=SSY=SSZ=0.0

In the case of an extended source, the active volume can be restricted to that of a body or a set of bodies, which must be defined as parts of the geometry. The activity of the source is assumed to be uniform within the volume of the intersection of the active bodies and the source box. Note that the initial coordinates of primary particles are sampled by the rejection method; the sampling efficiency is equal to the fraction of the source box volume that is occupied by active bodies. The program issues multiple warning messages when the efficiency is less than 0.005.

To define each active source body, add the following line:

`SBODY␣ ...` Active source body. One line for each body.
`-- Default:` none

The bodies in the material structure are normally identified by the sequential labels assigned by PENGEOM (see Section 6.4). The program stops if the source box has not been defined previously.

`SCONE␣ ...` Conical source beam (the initial direction of primary particles is sampled uniformly within the solid angle of a cone). Polar and azimuthal angles of the beam axis direction, `THETA` and `PHI`, and angular semiaperture, `ALPHA`, in deg.
`-- Defaults: THETA=0.0, PHI=0.0, ALPHA=0.0`
The case `ALPHA=0.0` defines a monodirectional source, and `ALPHA=180.0` corresponds to an isotropic source.

`SRECTA ...` Rectangular source beam (the initial direction of primary particles is sampled uniformly within the solid angle of a rectangular window on the unit sphere). Limiting polar and azimuthal angles of the source beam window, `(THETAL, THETAU)`×`(PHIL,PHIU)`, in deg.
`-- Defaults: THETAL=0.0, THETAU=0.0, PHIL=0.0, PHIU=0.0`

The case `THETAL=THETAU`, `PHIL=PHIU` defines a monodirectional source. To define an isotropic source, set `THETAL=0.0`, `THETAU=180.0`, `PHIL=0.0` and `PHIU=360.0`.

Note that the default source is a pencil beam that moves upwards along the z-axis.

- **Input phase-space file**

`IPSFN␣ ...` Name of an input phase-space file (up to 20 characters).

The initial state variables of particles (primary and secondary) 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. In addition, photons from phase-space files are assumed to be unpolarised.
`-- Default:` none

Up to 100 phase-space files may be declared. They are read sequentially.

Phase-space files are in text 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 7.1.2 and Table 7.6).
-- NSHI, incremental shower number, defined as the difference between the shower numbers of the present particle and of the one preceding it in the phase-space file (employed instead of the shower number to reduce the file size).

Phase-space files can be generated by running penmain using an impact detector with the flag IPSF = 1 (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 for by splitting the particles from the phase-space files.

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. Excessive splitting manifests in the form of apparently smooth results accompanied by large error bars.

WGTWIN ... Weight window, (WGMIN,WGMAX). Particles in the phase-space file that have initial weights WGHT less than WGMIN will be subjected to Russian roulette, and those with WGHT larger than WGMAX will be split. Note that the weight window has preference over the splitting option, *i.e.*, a particle will be split into NSPLIT or less particles only if the latter have weights larger than WGMIN.
-- Default: WGMIN=1.0E-35, WGMAX=1.0E35 (no action)

Normally, the efficiency of a simulation increases when all simulated particles have similar weights.

EPMAX␣ ... Highest energy (in eV) of particles in the phase-space files. EPMAX is the upper limit of the energy interval covered by the simulation look-up tables. To minimise interpolation errors, EPMAX should not be much larger than the highest 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)

When the value of `EPMAX` exceeds 1 GeV, which is the highest energy covered by the database, the macroscopic cross sections for energies higher 1 GeV are approximated by linear log-log extrapolation of the database values.

- **Material data and simulation parameters**

`MFNAME ...` Name of a PENELOPE input material data file (up to 20 characters). This file must be generated in advance by running the program `material`.
`-- Default:` none

Alternatively, the material data file can be generated by using the program `tables`, which produces tables of macroscopic cross sections ready for visualisation with GNUPLOT. `tables` can also be used to verify the consistency of existing material files.

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

```
-- Defaults:  EABS(1,M)=EABS(3,M)=0.01*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 highest energy of all particles found in the simulation. For radiation sources other than phase-space files, `EPMAX` is determined by the source energies.

For each material we must provide the name of the material-data file and a set of simulation parameters. The label (material number) assigned by PENELOPE to each material is determined by the ordering of the material list in the input file. That is, the first, second, ... materials are assigned the labels 1, 2, ... These labels are also used in the geometry definition. The input file may contain multiple declarations of the same material, *e.g.*, to specify different sets of simulation parameters in different bodies with the same composition.

The original programs in the distribution package allow up to 10 materials. This number can be increased by changing the value of the parameter `MAXMAT` in the Fortran module `PENELOPE_mod` (see Table 7.4), which is at the heading of the `penelope.f` source file.

- **Geometry definition and local simulation parameters**

`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 6.6).

The bodies in the material structure are normally identified by the sequential labels assigned by PENGEOM. These labels can be found by running the viewer GVIEW2D or by inspecting the geometry report file `pengeom.rep` written by PENGEOM. For

complex geometries, however, it may be more practical to employ user labels, *i.e.*, the four-character strings that identify the body in the geometry definition file.

In `penmain`, a body can be specified by giving either its PENGEOM numerical label or its user label enclosed in a pair of apostrophes (*e.g.*, `'BOD1'`). However, bodies that result from the cloning of modules (as well as those defined in an `INCLUDE`d geometry file) do not have a user label and only the `pengeom` numerical label is acceptable.

`PARINP ...` The values of certain parameters of the geometry definition may be defined from the main program by means of the array `PARINP` (an input argument of the `GEOMIN` subroutine, Section 6.4). The entered `PARINP(IP)` value replaces the parameter values that are marked with the index `IP` in the geometry definition file.
`-- Default:` none

`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)

`EABSB␣ ...` Local absorption energies `EABSB(KPAR,KB)` of particles of type `KPAR` in body `KB`. These values must be larger than `EABS(KPAR,M)`, where `M` is the material of body `KB`. When the particle is moving within body `KB`, the absorption energy `EABS(KPAR,M)` is temporarily set equal to `EABSB(KPAR,KB)`. Thus, the simulation of the particle history is discontinued when the energy becomes less than `EABSB(KPAR,KB)`. This feature can be used, *e.g.*, to reduce the simulation work in regions of lesser interest.
`-- Default: EABSB(KPAR,KB)=EABS(KPAR,M)` (no action)

- **Variance reduction techniques**

`IFORCE ...` Activates forcing of interactions of type `ICOL` of particles `KPAR` in body `KB`; a negative value of `KB` indicates the set of bodies made of material |`KB`|. `FORCER` is the forcing factor, $\mathcal{F}$, which must be larger than unity (Section 1.6.1). The values `WLOW`, `WHIG` are the limits of the weight window where interaction forcing is applied (see Section 7.1.4). When several interaction mechanisms are forced in the same body, the effective weight window is set equal to the intersection of the windows for these mechanisms.
`-- Default:` no interaction forcing

If the mean free path for real interactions of type `ICOL` is `MFP`, the program will simulate interactions of this type (real or forced) with an effective mean free path equal to `MFP/FORCER`.

TRICK: a negative input value of `FORCER`, $-\mathcal{F}$, is interpreted as follows,
`KPAR` = 1 or 3: electrons or positrons with energy $E =$ `EMAX` should interact, on

average and approximately, $\mathcal{F}$ times in the course of their slowing down to rest, *i.e.*, in a path length equal to their CSDA range.
KPAR = 2: the mean free path of photons with energy $E =$ EMAX between interactions of type ICOL is set equal to $\mathcal{F}$.
This is very useful, *e.g.*, to generate x-ray spectra.

The real effect of interaction forcing on the efficiency is not easy to predict. Please, do tentative runs with different FORCER values and check the efficiency gain (or loss!).

IBRSPL ... Activates bremsstrahlung splitting in body KB for electrons and positrons with weights in the window (WLOW,WHIG) where interaction forcing is applied. A negative value of KB indicates the set of bodies made of material |KB|. The integer IBRSPL is the splitting factor.
-- Default: no bremsstrahlung splitting

Note that bremsstrahlung splitting is applied in combination with interaction forcing and, consequently, it is activated only in those bodies where interaction forcing is active.

IXRSPL ... Splitting of characteristic x rays emitted in body KB, from any element. A negative value of KB indicates the set of bodies made of material |KB|. Each unsplit x ray with ILB(2)=2 (*i.e.*, of the second generation) when extracted from the secondary stack is split into IXRSPL quanta. The new, lighter, quanta are assigned random directions distributed isotropically.
-- Default: no x-ray splitting

• Distributions of emerging particles

The program generates energy and angular distributions of the particles that emerge from the material system, that is, that leave the enclosure (the root module in the PENGEOM definition of the geometry).

NBE␣␣␣... Limits EL and EU of the interval where energy distributions of emerging particles are tallied, and number NBE of bins in the output distribution (NME $\leq$ 1,500).
-- Defaults: EMIN=0.0, EMAX=EPMAX, NBE=100

Although the number of bins must be positive, the program also admits negative values of NBE. When NBE is positive, energy bins have uniform width equal to DE=(EU-EL)/NBE. When the input value of NBE is negative, the output distribution consists of |NBE| bins whose widths increase geometrically with the energy, *i.e.*, the energy bins have uniform width on a logarithmic scale.

NBANGL ... Number of bins for the polar angle θ and the azimuthal angle ϕ, respectively, NBTH and NBPH ($\leq$ 1800 and 180, respectively).
-- Default: NBTH=90, NBPH=1 (azimuthal average)

If NBTH is positive, polar angular bins have uniform width, DTH=180/NBTH. When NBTH is negative, the bin width increases geometrically with θ, *i.e.*, the bins have uniform width on a logarithmic scale (limited to angles larger than 10^{-2} deg).

NOTE: In the output files, the terms "upbound" and "downbound" are used to denote particles that leave the material system moving upwards (W>0) and downwards (W<0), respectively[6].

• Impact detectors

IMPDET ... Starts the definition of a new impact detector (see Section 6.5). Up to 25 different detectors can be considered.

EL and EU are the lower and upper limits of the energy window covered by the detector.

NBE is the number of bins in the output energy spectrum of the detector ($\leq$ 1,000). If NBE is positive, energy bins have uniform width, DE=(EU-EL)/NBE. When NBE is negative, the bin width increases geometrically with the energy, *i.e.*, the energy bins have uniform width on a logarithmic scale.

Notice that the spectrum is given in absolute units (per primary particle and per eV).

The integer flag IPSF serves to activate the creation of a phase-space file (psf), which contains the state variables of all particles that enter the detector. The psf may be generated for only one detector in each run. Use this option with care because psf's may grow very quickly.
IPSF=0; no psf is created.
IPSF=1; the psf is created.

The integer flag IDCUT allows the tracking of particles that enter the detector to be discontinued.
IDCUT=0; the simulation of a particle is discontinued when it enters the detector (useful to stop the simulation of particles recorded in a psf).
IDCUT=1; the presence of the detector does not affect the tracking of particles.
IDCUT=2; the presence of the detector does not affect the tracking of particles. The distribution of particle fluence with respect to energy (integrated over the volume of the detector) is tallied. The calculated distribution has dimensions of length/energy.
-- Default: none

IDSPC␣ ... Name of the output energy-spectrum file (up to 20 characters).
-- Default: spc-impdet-##.dat

IDPSF␣ ... Name of the output phase-space file (up to 20 characters).
-- Default: psf-impdet.dat

[6]In previous versions of the programs we used the respective terms "transmitted" and "backscattered", which did cause some confusion when primary particles were incident from above.

`IDFLNC ...` Name of the output file with the energy distribution of particle fluence (20 characters). This file is generated only when `IDCUT=2`.
`-- Default: fln-impdet-##.dat`

`IDAGEL␣ ...` Activates the evaluation of the age of particles, defined as the time elapsed since the start of the primary particle that originated the shower. The program generates the age distribution of detected particles, *i.e.*, particles of the types declared in lines `IDKPAR` (see below) that enter the detector with energy in the window (`EL`,`EU`). The age distribution is tallied in the interval between `AGEL` and `AGEU` (both in seconds), which is partitioned into `NAGE` bins. If `NAGE` is positive, the age bins have uniform width. When `NAGE` is negative, the width of the age bins is uniform on a logarithmic scale.
`-- Defaults: NAGE=100, AGEL=0.0, AGEU` must always be specified

`IDAGEF␣ ...` Name of the output age distribution file (up to 20 characters).
`-- Default: age-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. Active bodies cannot be void because the geometry routines would not stop particles at their limiting surfaces. If you need to define detectors outside the material system, fill them with an arbitrary material of very small density to avoid perturbing the transport process.

`IDKPAR ...` Type of particle that is detected (1=electrons, 2=photons or 3=positrons). One line for each type.
`-- Default:` All particles are detected

The detector has no effect for particles that are not detected. This feature can be used, *e.g.*, to make a body or a set of bodies opaque to particles of a certain type.

- **Outer angular detectors**

Each angular detector collects particles that leave the enclosure with directions within a rectangular window on the unit sphere limited by the "parallels" `THETA1` and `THETA2` and the "meridians" `PHI1` and `PHI2`. The output spectrum is the energy distribution of particles that emerge within the acceptance solid angle of the detector with energies in the interval from `EL` to `EU`, recorded using `NBE` bins. Notice that the spectrum is given in absolute units (per primary particle, per eV, and per unit solid angle).

`ANGDET ...` Starts the definition of a new angular detector. Up to 25 different detectors can be considered. `THETA1`, `THETA2`, and `PHI1`, `PHI2` are the limits of the angular intervals covered by the detector, in degrees.
`-- Defaults: none`

NOTE: `PHI1` and `PHI2` must be both either in the interval [0,360] or in the interval [-180,180].

ADENER ... EL and EU are the lower and upper limits of the energy window covered by the detector. NBE is the number of bins in the output energy spectrum ($\leq$ 1,000). If NBE is positive, energy bins have uniform width, DE=(EU-EL)/NBE. When NBE is negative, the bin width increases geometrically with the energy, *i.e.*, the energy bins have uniform width on a logarithmic scale.
-- Defaults: EL=0.0, EU=EPMAX, NBE=100

ADSPC␣ ... Name of the output spectrum file (up to 20 characters).
-- Default: spc-angdet-##.dat

• **Energy-deposition detectors**

Each energy-deposition detector consists of a set of active bodies, which must have been defined as parts of the geometry. The output spectrum is the distribution of absorbed energy (per primary shower) in the active bodies.

ENDETC ... Starts the definition of a new energy-deposition detector. Up to 25 different detectors can be considered. EL and EU are the lower and upper limits of the energy window covered by the detector.

NBE is the number of bins in the output energy spectrum ($\leq$ 1,000). If NBE is positive, energy bins have uniform width, DE=(EU-EL)/NBE. When NBE is negative, the bin width increases geometrically with the energy, *i.e.*, the energy bins have uniform width on a logarithmic scale.

WARNING: The energy-deposition spectrum may be strongly biased when interaction forcing or x-ray splitting are applied, even outside the detector bodies. The program does not admit energy-deposition detectors when interaction forcing is active.

EDSPC␣ ... Name of the output spectrum file (up to 20 characters).
-- Default: 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.

• **Absorbed dose distributions**

Generally, the program can calculate the absorbed dose distribution inside a parallelepiped (dose box) whose edges are parallel to the axes of the laboratory frame. The dose box is defined by giving the coordinates of its vertexes. The dose is tallied using a uniform orthogonal grid with NDBX, NDBY and NDBZ ($\leq$ 201) bins (= voxels) along the directions of the respective coordinate axes. These numbers should be odd, to make sure that the "central" axes (*i.e.*, lines that join the centres of opposite faces of the box) go through the centres of a row of voxels.

`GRIDX␣ ...` x-coordinates of the vertices of the dose box and number of bins in the x direction (≤ 201).
`-- Default:` none

`GRIDY␣ ...` y-coordinates of the vertices of the dose box and number of bins in the y direction (≤ 201).
`-- Default:` none

`GRIDZ␣ ...` z-coordinates of the vertices of the dose box and number of bins in the z direction (≤ 201).
`-- Default:` none

The efficiency of the dose map calculation can be increased by taking advantage of possible symmetries of the system (source and geometry). In problems with axial symmetry about the z axis, it is advantageous to tally the dose distribution in the volume of a cylinder of radius `RU` centred at the z axis, limited by the planes $z =$`ZL` and $z =$`ZU`. For systems with spherical symmetry about the origin of coordinates, it is most convenient to consider the radial dose distribution in a sphere of radius `RU`. The generation of these symmetric dose maps is activated by entering the line

`GRIDR␣ ...` Radius RU of the dose zone and number of radial bins (≤ 201).
`-- Default:` none

Specifically, when the input file contains only the lines `GRIDZ␣` and `GRIDR␣`, the program assumes that the dose distribution is axially symmetric and generates a cylindrical map. When the input file has only the line `GRIDR␣`, spherical symmetry is assumed and the radial distribution of absorbed dose is tallied.

The different types of dose maps are mutually exclusive. Notice that when the assumed symmetry does not hold, the program may not be able to evaluate the masses of voxels correctly.

- **Job properties**

`RESUME ...` The program will read the dump file `dump1.dmp` (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 `dump2.dmp` (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 `dump2.dmp` already exists, it is overwritten.

`DUMPP␣ ...` Generate a dump file and write the simulation results in the output files every `DUMPP` seconds. The name of the dump file is the one declared in the

preceding DUMPTO line; when a DUMPTO line is not included, the dump file has the default name dump.dmp.
-- Default: DUMPP=1.0E15

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. Because the simulation work is halted while results are being written the dumping period DUMPP should not be too small.

LTRACK ... When this line is included in the input file with a positive value of NSHOW, the program generates three files, named electron.trk, photon.trk, and positron.trk with the particle tracks of up to NSHOW showers. The contents of these files may be visualized by using the GNUPLOT script viewtracks.gnu. The highest allowed value of NSHOW is 50.
-- Default: none

RSEED␣ ... Seeds of the random-number generator.
-- Default: ISEED1=1; ISEED2=1

When ISEED1 is equal to a negative integer, -N, the seeds are set by calling subroutine RAND0(N) with the input argument equal to N. This ensures that sequences of random numbers used in different runs of the program (with different values of N) are truly independent.

NSIMSH ... Desired number of simulated showers.
-- Default: DSHN=2.0E9

TIME␣␣ ... Allotted simulation time, in sec.
-- Default: TIMEA=2.0e9

END␣␣␣ ... Ends the reading of the input file. This line is needed only when the lines NSIMH and TIME␣␣ are missing.

The program is aborted when an incorrect input datum is found. The conflicting quantity usually appears in the last line of the output file penmain.dat. If the trouble is with arrays having dimensions smaller than required, the program indicates how the problem can be solved (this usually requires editing the source file, be careful).

7.3 Selecting the simulation parameters

The speed and accuracy of the simulation of electrons and positrons is determined by the values of the simulation parameters $E_{\rm abs}$, C_1, C_2, $W_{\rm cc}$, and $W_{\rm cr}$, which are selected

by the user for each material in the simulated structure[7]. Here we summarise the rules for assigning "safe" values to these parameters.

The absorption energies $E_{\rm abs}$ are determined either by the characteristics of the experiment or by the required space resolution. If we want to tally dose or deposited-charge distributions, $E_{\rm abs}$ should be such that the residual range $R(E_{\rm abs})$ of electrons/positrons is less than the typical dimensions of the volume bins used to tally these distributions[8]. In other cases, it is advisable to run short simulations (for the considered body alone) with increasing values of $E_{\rm abs}$ (starting from 50 eV) to study the effect of this parameter on the results. It is worth mentioning that the input values `EABS(KPAR,M)` are used to set the lower limit of the energy interval covered by the look-up tables. During the simulation, `EABS(KPAR,M)` can be given arbitrary higher values, for instance, to absorb particles in bodies that are far from the region of interest. This feature is used in the example main programs, which allow the user to define local absorption energies.

It is worth noting that the use of different absorption energies in neighbouring bodies may originate visible artefacts in the space distribution of absorbed dose. For instance, if the values of $E_{\rm abs}$ for electrons in bodies 1 and 2 are, respectively, 10 keV and 100 keV, electrons entering body 2 from body 1 with E less than 100 keV will be absorbed at the first interaction, giving an excess of dose in the border of body 2. When the spatial distribution of absorbed dose is important, absorption energies should be given essentially the same values over the region of interest. If the absorption energies of *the three types of transported particles* are given the same value in all the materials present, the simulated dose distribution is continuous when there is effective equilibrium of radiation with energy less than $E_{\rm abs}$.

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 the results remain essentially unaltered with the augmented parameter value (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 $W_{\rm cc}$ and $W_{\rm cr}$ have a weak influence on the accuracy of the results provided that they are both smaller than the width of the bins used to tally energy distributions. It is worth recalling that the DCS for inelastic collisions and bremsstrahlung emission decrease rapidly with the energy loss W (roughly as W^{-2} and W^{-1}, respectively). As a consequence, for particles with energies larger

[7] To specify simulation parameters for a single body we can simply assign a specific material to this body (the input file may contain multiple declarations of the same material). Note also that individual bodies can be assigned absorption energies higher than those specified for the material (see Section 7.1.2)

[8] PENELOPE 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`.

than about 100 keV, when $W_{\rm cc}$ and $W_{\rm cr}$ are increased, the simulation speed tends to a saturation value. For these high energies, the gain in speed is small when the cutoffs are made larger than about 5 keV. On the other hand, these cutoff energies have an effect on the energy-straggling distributions, which are faithfully described only when the number of hard interactions is "statistically sufficient". Therefore, the cutoff energies should not be too large. Our recommendation is to set the cutoff energies equal to one hundredth of the typical energy of primary particles, or 5 keV, whichever is the smallest. Note that, for the sake of consistency, $W_{\rm cc}$ must be smaller than the absorption energy of electrons in the material, `EABS(1,M)` (otherwise, we would lose secondary electrons that have energies larger than `EABS(1,M)`). Similarly, $W_{\rm cr}$ must be less than the photon absorption energy `EABS(2,M)`.

The maximum allowed step length $s_{\rm max}$ (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 about 10 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 $\sim 4\lambda_{\rm T}^{(\rm h)}$ (see Section 4.4). Therefore, for thick bodies (thicker than $\sim 10\lambda_{\rm T}^{(\rm h)}$), the average number of hinges along each track is larger than about 10, and it is not necessary to limit the length of the steps. In this case, we can set $s_{\rm max} = 10^{35}$ cm, or some other very large value, to switch off the external step-length control. It is worth noticing that, when the slowing-down of the particle due to soft events is described within the CSDA (see Section 4.3.3), the average energy lost in very thin bodies is described correctly. When using the CSDA for soft energy-loss events, external step control is normally unnecessary.

The example main program `pencyl` (see Section 7.2.1) 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. 7.5 and 7.6 display partial results from `pencyl` 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 types of particles) and $W_{\rm cr}$ was given a negative value, which compels PENELOPE to set $W_{\rm cr} = 10$ eV, and to disregard emission of soft bremsstrahlung (with $W < 10$ eV). That is, radiative events that yield bremsstrahlung photons with energy higher than 10 eV were simulated individually. First, we ran `pencyl.exe` using $W_{\rm cc} = 0$ 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 7.5 and 7.6 show results from this detailed simulation, obtained from 100

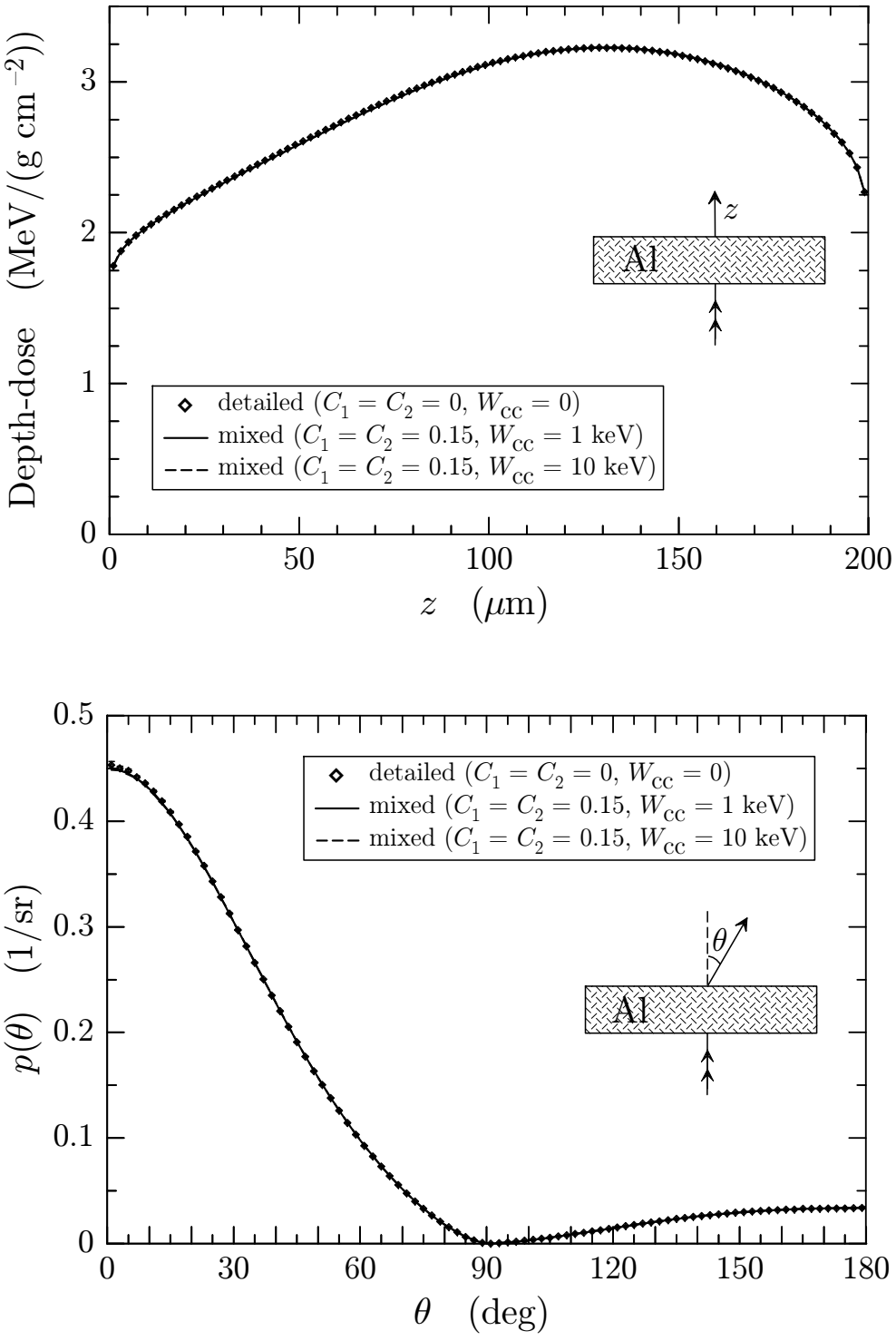

Figure 7.5: Results from `pencyl` for a 500-keV electron beam impinging normally on the surface of a 200-μm-thick aluminium slab (further details are given in the text). Top: depth-dose distribution within the slab. Bottom: angular distribution of emerging (transmitted and backscattered) electrons (primaries and secondaries).

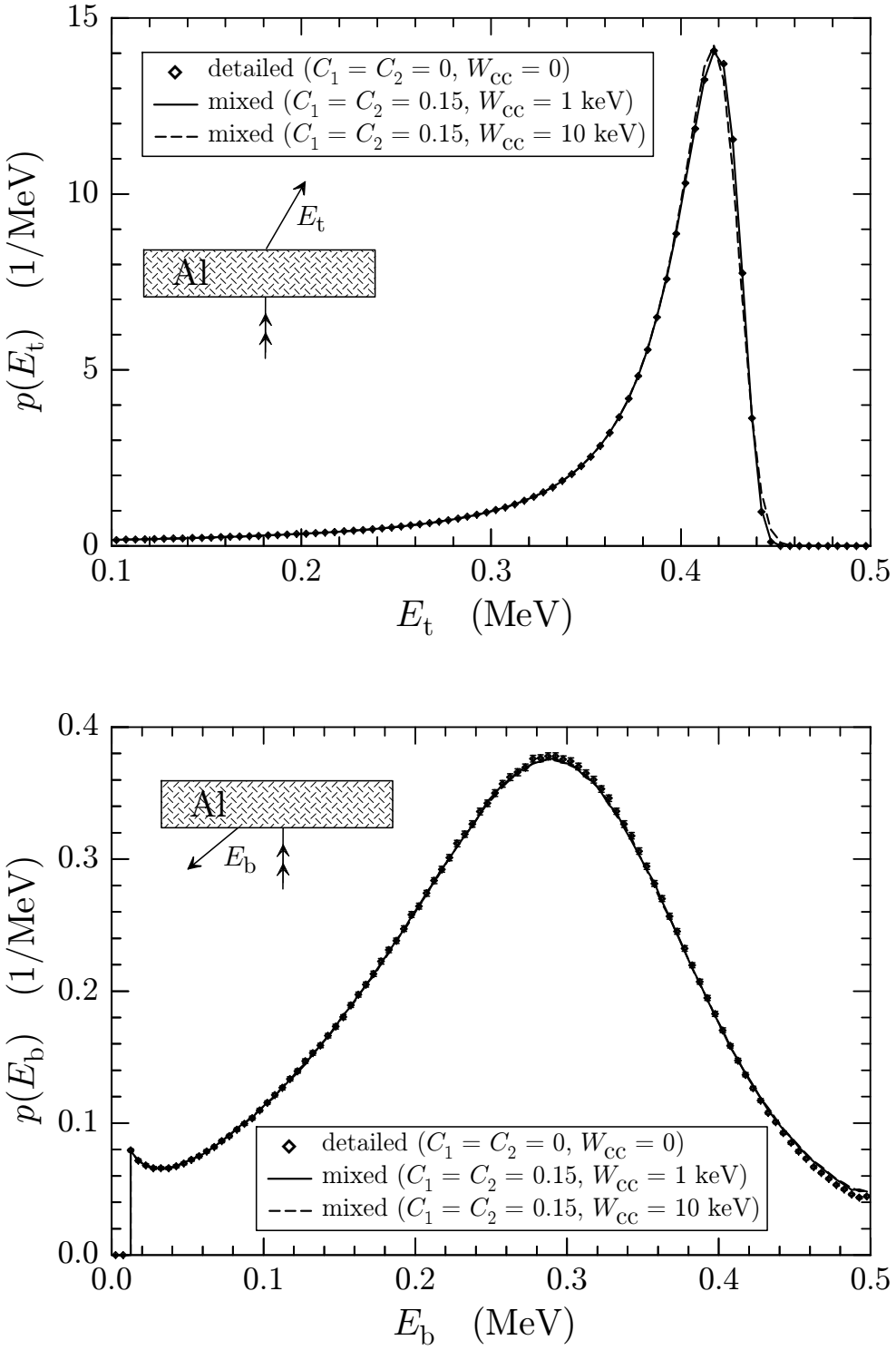

Figure 7.6: Results from `pencyl` for a 500-keV electron beam impinging normally on the surface of a 200-μm-thick aluminium slab (further details are given in the text). Top: energy distribution of transmitted (upbound) electrons. Bottom: energy distribution of backscattered (downbound) electrons. Secondary electrons are included in both cases.

million random showers[9]; the error bars indicate statistical uncertainties (3σ). The continuous lines in Figs. 7.5 and 7.6 represent results from a mixed simulation with $C_1 = C_2 = 0.15$, $s_{\max} = 0.002$ cm, $W_{\rm cc} = 1$ keV (and $W_{\rm cr} = -10$ 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 35 times higher than that of the detailed simulation[10]. The continuous lines in Figs. 7.5 and 7.6 represent results from this mixed simulation obtained from 100 million showers; the associated statistical uncertainties are similar to 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. 7.6), where mixed simulation tends to slightly overestimate 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 both near and across an interface.

To illustrate the effect of the cutoff energy loss $W_{\rm cc}$, 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 $W_{\rm cc} = 1$ keV, except for the energy distribution of transmitted electrons, which were slightly distorted (see the top plot in Fig. 7.6); a similar distortion occurs for the distribution of energy deposited in the slab. On the other hand, the simulation was only ∼22% faster than with $W_{\rm cc} = 1$ keV. As indicated above, when $W_{\rm cc}$ and $W_{\rm cr}$ are larger than ~ 5 keV, the simulation speed is fairly insensitive to variations of these cutoff values because most inelastic collisions and bremsstrahlung emission events involve energy losses much smaller than ~ 5 keV.

7.4 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 features 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.exe` that generates electron-photon showers within a slab and displays them (projected) on the computer screen plane. The current version of SHOWER operates only under Microsoft

[9]The program was compiled with the GFORTRAN compiler and ran on an Intel Core i7-3770 at 3.40 GHz. The generation of the 100 million showers took 63.5 hours.

[10]The simulation speed here is bound by the fact that each particle must undergo a minimum number of hinges. For thick bodies, much higher simulation speeds can be reached.

Windows. By default, the program considers one of the 280 materials defined in file `pdcompos.pen` and listed in Table 7.1 (and in file `material-list.txt`). It can also be run for any material whose definition file (generated by one of the programs `material` or `tables`) is placed in the directory of the executable binary `shower.exe`. 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.

Primary particles of a selected type and energy start moving either in the direction of the inward normal to the lateral surface (external or internal beams) or along random directions (isotropic inner source). Electron, photon and positron tracks are displayed in different colours and intensities that vary with the energy of the particle. 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 graphic window, the view plane can be rotated about the horizontal screen axis by typing "`r`" and the rotation angle in degrees; the screen plane can also be rotated progressively, by 15 deg steps, by pressing the "enter" key repeatedly. Entering the single-character command "`n`" erases the screen and displays a new bunch of showers. Observation of single showers projected on a revolving plane gives a truly three-dimensional perspective of the transport process.

7.5 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. 7.7):

• **Directory fsource**. This contains the following 6 Fortran source files:

`penelope.f ...` basic subroutine package for the simulation of coupled electron-photon transport in unbounded media.

`rita.f ...` subroutines for random sampling from single-variate discrete and continuous probability distributions.

`pengeom.f ...` subroutine package for tracking particles through modular quadric geometries (handles systems with up to 10,000 surfaces and 5,000 bodies).

`penvared.f ...` variance-reduction subroutines (splitting, Russian roulette, interaction forcing with bremsstrahlung splitting, and Woodcock's photon delta scattering).

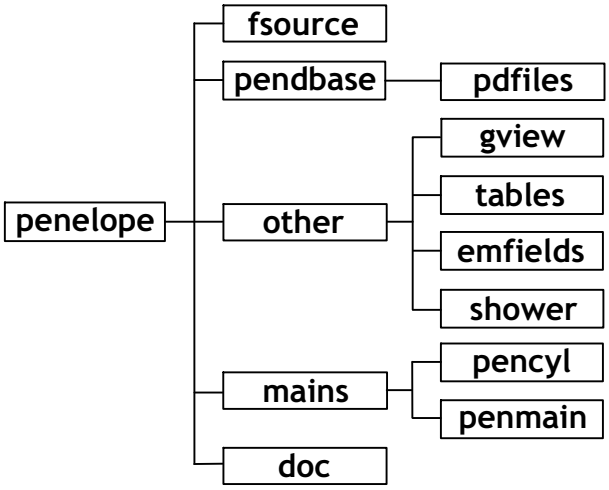

Figure 7.7: Directory tree of the PENELOPE code system.

`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`, `penelope.f` and `rita.f`. Notice that `penelope.f` and `rita.f` are declared through an include statement inside `material.f` and do 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`). Material data files can also be generated by running the program `tables` (see below).

• **Directory pendbase**. PENELOPE database. The directory `pdfiles` comprises the 995 files listed in Section 7.1.1. The compressed file `pdgph-photacs.zip` contains an alternative database of photoelectric cross sections that were calculated without the screening normalisation correction (see Section 2.1). To run PENELOPE without that correction, replace the files `pdgphZZ.p18` in `/pendbase/pdfiles` with those in `pdgph-photacs.zip`.

• **Directory other**. Consists of the following 4 subdirectories,

`gview ...` . Contains the geometry viewers GVIEW2D and GVIEW3D (see Section 6.6), which are operable under Microsoft Windows, and several examples of geometry definition files.

`shower ...` Contains the executable binary files of the programs `shower` (see Section 7.4) and `tables`, which operate under Microsoft Windows. The code `shower` generates electron-photon showers within a slab and displays the showers projected

on the screen plane. The program `tables` reads material data files and generates tables of interaction data (cross sections, mean free paths, stopping powers, ranges, ...) as functions of energy. `tables` can also generate material data files. To use these codes, just copy the files `shower.exe` and `tables.exe` into the directory `pendbase` and run them from there. These tools are useful for teaching purposes, they make 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 source file `tables.f` of the program `tables`, which reads a material data file and generates tables of interaction properties (cross sections, mean free paths, ranges, ...). A number of GNUPLOT scripts are provided for plotting 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, as well as restricted stopping powers. At the start, `tables` asks for the name of the material data file, which is expected to be located in the working directory. If the material data file does not exist, and if the executable binary of `tables` is in directory `pendbase`, the program builds that file by calling the PENELOPE routines.

The subdirectory `class-II` contains the program `tables-class-II.f`, which gives mean-free paths and other transport parameters for hard events of electrons and positrons. This information is useful to analyse the effect of the simulation parameters on the speed of mixed (class-II) simulations.

The subdirectory `LEA` includes the programs `LEA.f` and `mutren.f` that calculate the linear energy absorption of electrons, positrons and photons from simulations with PENELOPE (see Chapter 5).

• **Directory mains**. Examples of main programs, 2 subdirectories.

`pencyl ...` Contains the main program `pencyl.f` for particle transport in cylindrical structures, the geometry viewer `gviewc.exe` for these structures (which operates under Microsoft Windows), and a layout of the input file. It also includes the following subdirectories:

- `examples`: complete sets of input and material-data files for several simple simulation exercises, each in a separate subdirectory.
- `gscripts`: a set of GNUPLOT scripts (with the extension `.gnu`) that display continuous distributions generated by `pencyl`. The script `filename.gnu` plots the contents of the output file of the same name and the extension `.dat`.
- `tools`: contains the source file `pencyl-sum.f` of a program that combines data in dump files resulting from multiple independent runs of the same problem.

`penmain ...` Contains the generic main program `penmain.f` for particle transport in quadric geometries, the file `pmcomms.f` with common blocks and parameter definitions (inserted in the main program and its subroutines through include statements), an example of a subroutine with the definition of a special source,

and a layout of the input file. It also includes the following subdirectories:
- `examples`: input and material-data files for several simulation exercises.
- `gscripts`: a set of GNUPLOT scripts (with the extension `.gnu`) that display continuous distributions generated by `penmain`. The script `filename.gnu` plots the contents of the output file of the same name and the extension `.dat`.
- `tools`: contains complementary tools for visualising the contents of phase-space files, for plotting 3D dose maps in planes perpendicular to the coordinate axes, and the source file `penmain-sum.f` of a program that combines data in dump files resulting from multiple independent runs of the same problem.

The executable files of `pencyl` and `penmain` are obtained by compiling and linking the following groups of source files:

```
pencyl :  pencyl.f, penelope.f, rita.f, penvared.f, timer.f
penmain:  penmain.f, penelope.f, rita.f, pengeom.f, penvared.f,
          source.f, pennucm.f, timer.f
```

NOTE: To simplify the typing of compilation commands, all the subroutine packages used by a main program have been declared through include statements within the program (see, for example, the source file `pencyl.f`) and do not have to be listed in the compilation command; still, all the included files must be in the same directory as the main program.

The main programs `pencyl` 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, `www.gnuplot.info`. The GNUPLOT scripts provided in the distribution package display the different distributions generated by the main programs. For instance, in Windows, after running `pencyl.exe` you can visualise the contents of the output file `depth-dose.dat` by simply 1) copying the file `depth-dose.gnu` from the directory `mains/pencyl/gscripts` to the directory that contains the simulation results; and 2) entering the command "`wgnuplot depth-dose.gnu`" (or clicking the icon of the script). Some of the scripts assume that the output files have their default names; if the user assigns different filenames to those output files, the scripts need to be edited and modified appropriately.

• **Directory doc**. Documentation files.

`tutorial.pdf ...` A guided tour through the PENELOPE code system. Includes brief descriptions of the programs and subroutine packages, instructions to install the code system, to build binary executable files, and to run various exercises.

`penelope-2018.pdf ...` The present manual.

The simulation programs are written in Fortran language[11]. To generate the executable binary files of the simulation programs you need to have a Fortran compiler installed on your computer. For Microsoft Windows (9x/NT/2000/XP/7/10), a number of free compilers are available. We use the Fortran compiler GFORTRAN[12] from the Free Software Foundation. Silverfrost[13] offers its Fortran 95 compiler Salford FTN95 Personal Edition free of charge for personal, non-commercial usage; on execution, programs compiled with FTN95PE display a banner.

[11]Updated information on Fortran can be found at "Clive Page's list of Fortran Resources", `www.star.le.ac.uk/~cgp/fortran.html`, including Page's (1988) book, *Professional Programmer's Guide to Fortran 77*, in various formats.

[12]`http://mingw-w64.sourceforge.net`

[13]`www.silverfrost.com/default.aspx`

Appendix A

Collision kinematics

To cover the complete energy range of interest in radiation transport studies we use relativistic kinematics. Let P denote the energy-momentum 4-vector of a particle, *i.e.*,

$$\mathsf{P} = (\mathcal{W}c^{-1}, \mathbf{p}), \tag{A.1}$$

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

$$\mathsf{P} \cdot \mathsf{P}' = \mathcal{W}\mathcal{W}'c^{-2} - \mathbf{p}\cdot\mathbf{p}', \tag{A.2}$$

is invariant under Lorentz transformations. The rest mass m of a particle determines the invariant length of its energy-momentum,

$$\mathsf{P} \cdot \mathsf{P} = \mathcal{W}^2c^{-2} - \mathbf{p}^2 = (mc)^2. \tag{A.3}$$

The kinetic energy E of a massive particle ($m \neq 0$) is defined as

$$E = \mathcal{W} - mc^2, \tag{A.4}$$

where mc^2 is the rest energy. The magnitude of the momentum is given by

$$(cp)^2 = E(E + 2mc^2). \tag{A.5}$$

In terms of the velocity $\mathbf{v}$ of the particle, we have

$$E = (\gamma - 1)mc^2 \quad \text{and} \quad \mathbf{p} = \beta\gamma mc\hat{\mathbf{v}}, \tag{A.6}$$

where

$$\beta \equiv \frac{v}{c} = \sqrt{\frac{\gamma^2 - 1}{\gamma^2}} = \sqrt{\frac{E(E + 2mc^2)}{(E + mc^2)^2}} \tag{A.7}$$

is the velocity of the particle in units of c and

$$\gamma \equiv \sqrt{\frac{1}{1 - \beta^2}} = \frac{E + mc^2}{mc^2} \tag{A.8}$$

is the total energy in units of the rest energy. From the relation (A.5), it follows that

$$E = \sqrt{(cp)^2 + m^2c^4} - mc^2 \tag{A.9}$$

and

$$\frac{\mathrm{d}p}{\mathrm{d}E} = \frac{1}{v} = \frac{1}{c\beta}. \tag{A.10}$$

For a photon (and any other particle with $m = 0$), the energy and momentum are related by

$$E = cp. \tag{A.11}$$

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)

$$\mathsf{P}_1 = (\mathcal{W}_1 c^{-1}, 0, 0, p_1), \tag{A.12a}$$

$$\mathsf{P}_2 = (m_2 c, 0, 0, 0), \tag{A.12b}$$

$$\mathsf{P}_3 = (\mathcal{W}_3 c^{-1}, p_3 \sin\theta_3, 0, p_3 \cos\theta_3), \tag{A.12c}$$

$$\mathsf{P}_4 = (\mathcal{W}_4 c^{-1}, -p_4 \sin\theta_4, 0, p_4 \cos\theta_4). \tag{A.12d}$$

Energy and momentum conservation is expressed by the 4-vector equation

$$\mathsf{P}_1 + \mathsf{P}_2 = \mathsf{P}_3 + \mathsf{P}_4. \tag{A.13}$$

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,

$$\begin{aligned} m_4^2c^2 &= \mathsf{P}_4{\cdot}\mathsf{P}_4 = (\mathsf{P}_1 + \mathsf{P}_2 - \mathsf{P}_3){\cdot}(\mathsf{P}_1 + \mathsf{P}_2 - \mathsf{P}_3) \\ &= \mathsf{P}_1{\cdot}\mathsf{P}_1 + \mathsf{P}_2{\cdot}\mathsf{P}_2 + \mathsf{P}_3{\cdot}\mathsf{P}_3 + 2\mathsf{P}_1{\cdot}\mathsf{P}_2 - 2\mathsf{P}_1{\cdot}\mathsf{P}_3 - 2\mathsf{P}_2{\cdot}\mathsf{P}_3 \\ &= m_1^2c^2 + m_2^2c^2 + m_3^2c^2 + 2\mathcal{W}_1\mathcal{W}_2c^{-2} \\ &\qquad - 2\left(\mathcal{W}_1\mathcal{W}_3c^{-2} - p_1p_3\cos\theta_3\right) - 2\mathcal{W}_2\mathcal{W}_3c^{-2}, \end{aligned} \tag{A.14}$$

and it follows that

$$\cos\theta_3 = \frac{m_4^2c^4 - m_1^2c^4 - m_2^2c^4 - m_3^2c^4 + 2\mathcal{W}_1(\mathcal{W}_3 - \mathcal{W}_2) + 2\mathcal{W}_2\mathcal{W}_3}{2\left(\mathcal{W}_1^2 - m_1^2c^4\right)^{1/2}\left(\mathcal{W}_3^2 - m_3^2c^4\right)^{1/2}}. \tag{A.15}$$

Clearly, by symmetry, we can obtain a corresponding expression for $\cos\theta_4$ by interchanging the indices 3 and 4

$$\cos\theta_4 = \frac{m_3^2c^4 - m_1^2c^4 - m_2^2c^4 - m_4^2c^4 + 2\mathcal{W}_1(\mathcal{W}_4 - \mathcal{W}_2) + 2\mathcal{W}_2\mathcal{W}_4}{2\left(\mathcal{W}_1^2 - m_1^2c^4\right)^{1/2}\left(\mathcal{W}_4^2 - m_4^2c^4\right)^{1/2}}. \tag{A.16}$$

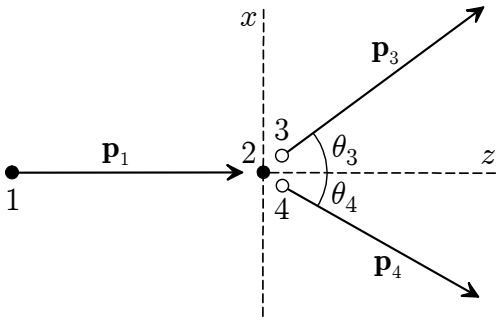

Figure A.1: Kinematics of two-body reactions.

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	$m_1 = \mathrm{m_e}$,	$\mathcal{W}_1 = E + \mathrm{m_e}c^2$.
Target:	Electron	$m_2 = \mathrm{m_e}$,	$\mathcal{W}_2 = \mathrm{m_e}c^2$.
Scattered particle:		$m_3 = \mathrm{m_e}$,	$\mathcal{W}_3 = E - W + \mathrm{m_e}c^2$.
Recoil electron:		$m_4 = \mathrm{m_e}$,	$\mathcal{W}_4 = W + \mathrm{m_e}c^2$.

$$\cos\theta_3 = \left(\frac{E-W}{E}\,\frac{E+2\mathrm{m_e}c^2}{E-W+2\mathrm{m_e}c^2}\right)^{1/2}, \tag{A.17}$$

$$\cos\theta_4 = \left(\frac{W}{E}\,\frac{E+2\mathrm{m_e}c^2}{W+2\mathrm{m_e}c^2}\right)^{1/2}. \tag{A.18}$$

- Compton scattering of photons by free electrons at rest.

Projectile:	Photon	$m_1 = 0$,	$\mathcal{W}_1 = E \equiv \kappa\,\mathrm{m_e}c^2$.
Target:	Electron	$m_2 = \mathrm{m_e}$,	$\mathcal{W}_2 = \mathrm{m_e}c^2$.
Scattered photon:		$m_3 = 0$,	$\mathcal{W}_3 \equiv \tau E$.
Recoil electron:		$m_4 = \mathrm{m_e}$,	$\mathcal{W}_4 = \mathrm{m_e}c^2 + (1-\tau)E$.

$$\cos\theta_3 = \frac{1}{\kappa}\left(\kappa + 1 - \frac{1}{\tau}\right), \tag{A.19}$$

$$\cos\theta_4 = (\kappa+1)\left(\frac{1-\tau}{\kappa\,[2+\kappa(1-\tau)]}\right)^{1/2}. \tag{A.20}$$

- Annihilation of positrons with free electrons at rest.

Projectile:	Positron	$m_1 = \mathrm{m_e}$,	$\mathcal{W}_1 = E + \mathrm{m_e}c^2 \equiv \gamma\,\mathrm{m_e}c^2$.
Target:	Electron	$m_2 = \mathrm{m_e}$,	$\mathcal{W}_2 = \mathrm{m_e}c^2$.
Annihilation photons:		$m_3 = 0$,	$\mathcal{W}_3 \equiv \zeta(E + 2\mathrm{m_e}c^2)$.
		$m_4 = 0$,	$\mathcal{W}_4 = (1-\zeta)(E + 2\mathrm{m_e}c^2)$.

$$\cos\theta_3 = \left(\gamma^2-1\right)^{-1/2}(\gamma+1-1/\zeta)\,, \tag{A.21}$$

$$\cos\theta_4 = \left(\gamma^2-1\right)^{-1/2}\left(\gamma+1-\frac{1}{1-\zeta}\right). \tag{A.22}$$

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

$$\cos\theta = \frac{E(E+2mc^2) - W(E+mc^2+Mc^2)}{\sqrt{E(E+2mc^2)\;(E-W)(E-W+2mc^2)}}. \tag{A.23}$$

The target recoil direction is given by Eq. (A.16),

$$\cos\theta_\mathrm{r} = \frac{(E+mc^2+Mc^2)W}{\sqrt{E(E+2mc^2)\,W(W+2mc^2)}}. \tag{A.24}$$

Solving Eq. (A.23), we obtain the following expression for the energy transfer W corresponding to a given scattering angle θ,

$$\begin{aligned} W \;=\; &\left[(E+mc^2)\sin^2\theta + Mc^2 - \cos\theta\sqrt{M^2c^4 - m^2c^4\sin^2\theta}\,\right] \\ &\times \frac{E(E+2mc^2)}{(E+mc^2+Mc^2)^2 - E(E+2mc^2)\cos^2\theta}. \end{aligned} \tag{A.25}$$

In the case of collisions of particles with equal mass, $m = M$, this expression simplifies to

$$W = \frac{E(E+2mc^2)\sin^2\theta}{E\sin^2\theta + 2mc^2} \qquad \text{if } M = m. \tag{A.26}$$

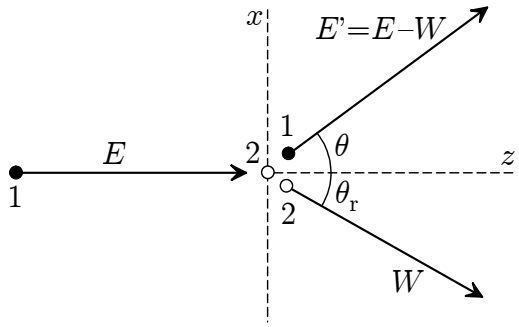

Figure A.2: Kinematics of elastic collisions.

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 ($m = \mathrm{m_e}$), 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

$$W = \frac{\left[(E + mc^2)\sin^2\theta + Mc^2(1 - \cos\theta)\right] E(E + 2mc^2)}{(E + Mc^2)^2 - E(E + 2mc^2)\cos^2\theta} \qquad \text{if } M \gg m. \tag{A.27}$$

The non-relativistic limit ($c \to \infty$) of this expression is

$$W = \frac{2m}{M}(1 - \cos\theta)E \qquad \text{if } M \gg m \text{ and } E \ll mc^2. \tag{A.28}$$

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 3.2.2). 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

$$Q(Q + 2\mathrm{m_e}c^2) = (cq)^2 = c^2\left(p^2 + p'^2 - 2pp'\cos\theta\right), \tag{A.29}$$

where $\mathrm{m_e}$ is the electron rest mass and $\theta = \arccos(\hat{\mathbf{p}}\cdot\hat{\mathbf{p}}')$ is the scattering angle. Equivalently, we can write

$$Q = \sqrt{(cq)^2 + \mathrm{m_e}^2c^4} - \mathrm{m_e}c^2. \tag{A.30}$$

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

$$\begin{aligned} Q_\pm &= \sqrt{(cp \pm cp')^2 + \mathrm{m_e}^2 c^4} - \mathrm{m_e} c^2 \\ &= \sqrt{\left[\sqrt{E(E+2mc^2)} \pm \sqrt{(E-W)(E-W+2mc^2)}\right]^2 + \mathrm{m_e}^2 c^4} - \mathrm{m_e} c^2. \end{aligned} \quad \text{(A.31)}$$

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

$$cp - cp' \simeq c\left(\frac{\mathrm{d}p}{\mathrm{d}E}W - \frac{1}{2}\frac{\mathrm{d}^2 p}{\mathrm{d}E^2}W^2\right) = \frac{W}{\beta}\left(1 + \frac{1}{2\gamma(\gamma+1)}\frac{W}{E}\right) \quad \text{(A.32)}$$

and calculate Q_- as

$$Q_- \simeq \sqrt{(cp - cp')^2 + \mathrm{m_e}^2 c^4} - \mathrm{m_e} c^2 \quad \text{(A.33)}$$

or, if $cp - cp' \ll \mathrm{m_e} c^2$,

$$Q_- \simeq \frac{1}{2}\frac{(cp - cp')^2}{\mathrm{m_e} c^2} - \frac{1}{8}\frac{(cp - cp')^4}{(\mathrm{m_e} c^2)^3}. \quad \text{(A.34)}$$

Thus, for $E \gg W$,

$$Q_-(Q_- + 2\mathrm{m_e} c^2) \simeq W^2/\beta^2. \quad \text{(A.35)}$$

In the non-relativistic limit,

$$Q \equiv q^2/2\mathrm{m_e}, \qquad Q_\pm = \frac{m}{\mathrm{m_e}}\left[E^{1/2} \pm (E-W)^{1/2}\right]^2. \quad \text{(A.36)}$$

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 $W = W_\mathrm{m}(Q)$ in the interval $0 < Q < Q_+(0)$. By solving the Eqs. $Q = Q_\pm(W_\mathrm{m})$ we obtain

$$W_\mathrm{m}(Q) = E + mc^2 - \sqrt{\left[\sqrt{E(E+2mc^2)} - \sqrt{Q(Q+2\mathrm{m_e} c^2)}\right]^2 + m^2 c^4}, \quad \text{(A.37)}$$

which, when $W \ll E$, reduces to

$$W_\mathrm{m}(Q) \simeq \beta\sqrt{Q(Q+2\mathrm{m_e} c^2)}. \quad \text{(A.38)}$$

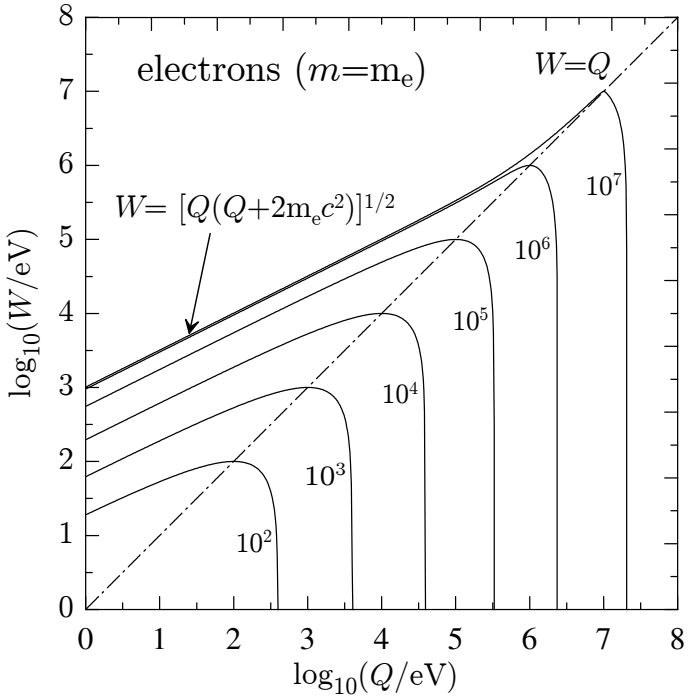

Figure A.3: Domains of kinematically allowed transitions in the (Q, W) plane for electrons/positrons. The curves represent the maximum allowed energy loss $W_{\rm m}(Q)$, given by Eq. (A.37), for electrons with the indicated kinetic energies (in eV). When E increases, $W_{\rm m}(Q)$ approaches the vacuum photon line, $W = [Q(Q+2{\rm m_e}c^2)]^{1/2}$, 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 $0 < W < W_{\rm m}(Q)$ (see Fig. A.2).

For a given energy loss W, the quantity

$$q_{\rm min} \equiv c^{-1}\sqrt{Q_-(Q_- + 2{\rm m_e}c^2)}, \tag{A.39}$$

is the minimum value of the momentum transfer in an inelastic collision, which occurs when $\theta = 0$. $q_{\rm min}$ is always larger than W/c. When the energy of the projectile increases, $\beta \to 1$, and $q_{\rm min}$ decreases and approaches (but never reaches) 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 $Q(Q + 2{\rm m_e}c^2) = W^2$ 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.2). Therefore, when the target is a single atom, the emission of photons by the projectile is not possible[1]. When the energy E of the projectile increases, Q_- decreases and tends to the photon line when β tends to unity. Hence, emission

[1]In a condensed medium, ultrarelativistic projectiles can emit real photons (Cerenkov radiation) under certain, quite restricting circumstances (Jackson, 1975).

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, in other words, as we approach the photon line. For a dilute gas, this causes a gradual increase of the cross section with the projectile energy when $\beta \to 1$.

The scattering angle θ is related to the energy loss through [see Eq. (A.29)]

$$\cos\theta = \frac{(cp)^2 + (cp')^2 - Q(Q + 2\mathrm{m_e}c^2)}{2(cp)(cp')}. \tag{A.40}$$

The recoil angle θ_r between $\mathbf{p}$ and $\mathbf{q}$ is given by

$$\cos\theta_\mathrm{r} = \frac{(cp)^2 - (cp')^2 + (cq)^2}{2(cp)(cq)}, \tag{A.41}$$

which can also be written in the form

$$\cos^2\theta_\mathrm{r} = \frac{W^2/\beta^2}{Q(Q + 2\mathrm{m_e}c^2)} \left(1 + \frac{Q(Q + 2\mathrm{m_e}c^2) - W^2}{2W(E + mc^2)}\right)^2. \tag{A.42}$$

For high-energy projectiles and collisions such that $Q \ll E$ and $W \ll E$,

$$\cos^2\theta_\mathrm{r} \simeq \frac{W^2/\beta^2}{Q(Q + 2\mathrm{m_e}c^2)} \simeq \frac{Q_-(Q_- + 2\mathrm{m_e}c^2)}{Q(Q + 2\mathrm{m_e}c^2)}. \tag{A.43}$$

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

$$f_i = f(x_i) \qquad (i = 1, \ldots, N). \tag{B.1}$$

The points (knots) x_i do not need to be equispaced, but we assume that they are in (strictly) increasing order

$$x_1 < x_2 < \cdots < x_N. \tag{B.2}$$

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 $x_i \leq x \leq x_{i+1}$

$$\varphi(x) = a_i + b_i x + c_i x^2 + d_i x^3 \equiv p_i(x) \qquad (i = 1, \ldots, N-1). \tag{B.3}$$

2) The polynomial $p_i(x)$ matches the values of $f(x)$ at the endpoints of the i-th interval,

$$p_i(x_i) = f_i, \quad p_i(x_{i+1}) = f_{i+1} \qquad (i = 1, \ldots, N-1), \tag{B.4}$$

so that $\varphi(x)$ is continuous in $[x_1, x_N]$.

3) The first and second derivatives of $\varphi(x)$ are continuous in $[x_1, x_N]$

$$p_i'(x_{i+1}) = p_{i+1}'(x_{i+1}) \qquad (i = 1, \ldots, N-2), \tag{B.5}$$

$$p_i''(x_{i+1}) = p_{i+1}''(x_{i+1}) \qquad (i = 1, \ldots, N-2). \tag{B.6}$$

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 $(i = 1, \ldots, N-1)$ we start from the fact that $\varphi''(x)$ is linear in $[x_i, x_{i+1}]$. Introducing the quantities

$$h_i \equiv x_{i+1} - x_i \qquad (i = 1, \ldots, N-1) \tag{B.7}$$

and

$$\sigma_i \equiv \varphi''(x_i) \qquad (i = 1, \ldots, N), \tag{B.8}$$

we can write the obvious identity

$$p_i''(x) = \sigma_i \frac{x_{i+1} - x}{h_i} + \sigma_{i+1} \frac{x - x_i}{h_i} \qquad (i = 1, \ldots, N-1). \tag{B.9}$$

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 $i = 1, \ldots, N-1$

$$p_i(x) = \sigma_i \frac{(x_{i+1} - x)^3}{6h_i} + \sigma_{i+1} \frac{(x - x_i)^3}{6h_i} + A_i(x - x_i) + B_i(x_{i+1} - x), \tag{B.10}$$

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.

$$\sigma_i \frac{h_i^2}{6} + B_i h_i = f_i \qquad \text{and} \qquad \sigma_{i+1} \frac{h_i^2}{6} + A_i h_i = f_{i+1}. \tag{B.11}$$

Finally, solving for A_i and B_i and substituting the result in (B.10), we obtain

$$\begin{aligned} p_i(x) \;=\; & \frac{\sigma_i}{6} \left[\frac{(x_{i+1} - x)^3}{h_i} - h_i(x_{i+1} - x) \right] + f_i \frac{x_{i+1} - x}{h_i} \\ & + \frac{\sigma_{i+1}}{6} \left[\frac{(x - x_i)^3}{h_i} - h_i(x - x_i) \right] + f_{i+1} \frac{x - x_i}{h_i}. \end{aligned} \tag{B.12}$$

To be able to use $\varphi(x)$ to approximate $f(x)$, we must find the second derivatives σ_i $(i = 1, \ldots, N)$. To this end, we impose the conditions (B.5). Differentiating (B.12) gives

$$p_i'(x) = \frac{\sigma_i}{6} \left[-\frac{3(x_{i+1} - x)^2}{h_i} + h_i \right] + \frac{\sigma_{i+1}}{6} \left[\frac{3(x - x_i)^2}{h_i} - h_i \right] + \delta_i, \tag{B.13}$$

where

$$\delta_i = \frac{f_{i+1} - f_i}{h_i}. \tag{B.14}$$

Hence,

$$p_i'(x_{i+1}) = \sigma_i \frac{h_i}{6} + \sigma_{i+1} \frac{h_i}{3} + \delta_i, \tag{B.15a}$$

$$p_i'(x_i) = -\sigma_i \frac{h_i}{3} - \sigma_{i+1} \frac{h_i}{6} + \delta_i \tag{B.15b}$$

and, similarly,

$$p'_{i+1}(x_{i+1}) = -\sigma_{i+1}\frac{h_{i+1}}{3} - \sigma_{i+2}\frac{h_{i+1}}{6} + \delta_{i+1}. \tag{B.15c}$$

Replacing (B.15a) and (B.15c) in (B.5), we obtain

$$h_i\sigma_i + 2(h_i + h_{i+1})\sigma_{i+1} + h_{i+1}\sigma_{i+2} = 6\,(\delta_{i+1} - \delta_i) \qquad (i = 1, \ldots, N-2). \tag{B.16}$$

The system (B.16) is linear in the N unknowns σ_i $(i = 1, \ldots, N)$. 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 $(i = 2, \ldots, N-1)$. The resulting system is, in matrix form,

$$\begin{pmatrix} H_2 & h_2 & 0 & \cdots & 0 & 0 & 0 \\ h_2 & H_3 & h_3 & \cdots & 0 & 0 & 0 \\ 0 & h_3 & H_4 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & H_{N-3} & h_{N-3} & 0 \\ 0 & 0 & 0 & \cdots & h_{N-3} & H_{N-2} & h_{N-2} \\ 0 & 0 & 0 & \cdots & 0 & h_{N-2} & H_{N-1} \end{pmatrix} \begin{pmatrix} \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \vdots \\ \sigma_{N-3} \\ \sigma_{N-2} \\ \sigma_{N-1} \end{pmatrix} = \begin{pmatrix} D_2 \\ D_3 \\ D_4 \\ \vdots \\ D_{N-3} \\ D_{N-2} \\ D_{N-1} \end{pmatrix}, \tag{B.17}$$

where

$$H_i = 2(h_{i-1} + h_i) \qquad (i = 2, \ldots, N-1) \tag{B.18}$$

and

$$\begin{aligned} D_2 &= 6(\delta_2 - \delta_1) - h_1\sigma_1 \\ D_i &= 6(\delta_i - \delta_{i-1}) \qquad\qquad (i = 3, \ldots, N-2) \\ D_{N-1} &= 6(\delta_{N-1} - \delta_{N-2}) - h_{N-1}\sigma_N. \end{aligned} \tag{B.19}$$

(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 on 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 $(i = 1, \ldots, N-1)$ —see Eq. (B.3)— can then be obtained by expanding the expressions (B.12):

$$\begin{aligned} a_i &= \frac{1}{6h_i}\left[\sigma_i x_{i+1}^3 - \sigma_{i+1}x_i^3 + 6\,(f_i x_{i+1} - f_{i+1}x_i)\right] + \frac{h_i}{6}\,(\sigma_{i+1}x_i - \sigma_i x_{i+1}), \\ b_i &= \frac{1}{2h_i}\left[\sigma_{i+1}x_i^2 - \sigma_i x_{i+1}^2 + 2\,(f_{i+1} - f_i)\right] + \frac{h_i}{6}\,(\sigma_i - \sigma_{i+1}), \\ c_i &= \frac{1}{2h_i}\,(\sigma_i x_{i+1} - \sigma_{i+1}x_i), \\ d_i &= \frac{1}{6h_i}\,(\sigma_{i+1} - \sigma_i). \end{aligned} \tag{B.20}$$

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 bear 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

$$\int_a^b f(x)\,\mathrm{d}x \simeq \int_a^b \varphi(x)\,\mathrm{d}x, \qquad x_1 \leq a \quad \text{and} \quad b \leq x_N, \tag{B.21}$$

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 $\varphi(x_\mathrm{c})$ —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 $x_k < x_\mathrm{c}$, 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

$$\int_A^B f(z)\,\mathrm{d}z, \tag{B.22}$$

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 (Abramowitz and Stegun, 1972), given by

$$\int_a^b f(z)\,\mathrm{d}z = \frac{b-a}{2}\sum_{i=1}^{20} w_i f(z_i) \tag{B.23}$$

with

$$z_i = \frac{b-a}{2}x_i + \frac{b+a}{2}. \tag{B.24}$$

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

$$w_i = \frac{2}{(1-x_i^2)\left[P'_{20}(x_i)\right]^2}. \tag{B.25}$$

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

$$\Delta_{20} = \frac{(b-a)^{41}(20!)^4}{41\,(40!)^3} f^{(40)}(\xi), \tag{B.26}$$

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 control is returned to the calling program.

Appendix C

Photon polarisation

The current version of PENELOPE can simulate the scattering of polarised photons according to the theory described in Section 2.7. Polarisation effects are accounted for by using DCSs that depend on the polarisation of the incident and scattered photons. Hence, the simulation algorithm requires defining the polarisation state of the primary photons and keeping track of the polarisation changes induced by scattering events. In the present Appendix we introduce elementary concepts on photon polarisation and we devise a consistent scheme for describing arbitrary polarisation states, which is tailored for Monte Carlo simulation of photon multiple scattering processes. Our treatment is based on the density matrix formalism and the representation of density matrices in terms of the so-called Stokes parameters, as described, *e.g.*, by Fano (1954a), McMaster (1954) and Tolhoek (1956).

C.1 Polarisation states

Let us consider classical electromagnetic plane waves that propagate in the direction given by the unit vector $\hat{\mathbf{k}}$. These waves are transverse, *i.e.*, their electric and magnetic fields are perpendicular to $\hat{\mathbf{k}}$. Hence, the polarisation of a wave can be described in terms of two orthogonal unit vectors $\hat{\boldsymbol{\epsilon}}_1$ and $\hat{\boldsymbol{\epsilon}}_2$ that lie in the plane orthogonal to $\hat{\mathbf{k}}$. It is convenient to select the polarisation vectors in such a way that $\hat{\boldsymbol{\epsilon}}_1$, $\hat{\boldsymbol{\epsilon}}_2$ and $\hat{\mathbf{k}}$ form a right-handed set of mutually orthogonal unit vectors, that is

$$\hat{\boldsymbol{\epsilon}}_1 \times \hat{\boldsymbol{\epsilon}}_2 = \hat{\mathbf{k}}, \quad \hat{\boldsymbol{\epsilon}}_1 \cdot \hat{\boldsymbol{\epsilon}}_2 = 0. \tag{C.1}$$

Clearly, the first polarisation vector, $\hat{\boldsymbol{\epsilon}}_1$, can be selected arbitrarily among the unit vectors orthogonal to $\hat{\mathbf{k}}$, and then the second polarisation vector is determined by the condition $\hat{\mathbf{k}} \times \hat{\boldsymbol{\epsilon}}_1 = \hat{\boldsymbol{\epsilon}}_2$. The electric field of a monochromatic plane wave of angular frequency ω that advances in the direction $\hat{\mathbf{k}}$ is

$$\boldsymbol{\mathcal{E}}(\mathbf{r}, t) = \mathcal{E}_0 \, \mathrm{Re} \left\{ [d_1 \hat{\boldsymbol{\epsilon}}_1 + d_2 \hat{\boldsymbol{\epsilon}}_2] \exp[\mathrm{i}(\mathbf{k} \cdot \mathbf{r} - \omega t)] \right\}, \tag{C.2a}$$

where $\mathbf{k} = (\omega/c)\hat{\mathbf{k}}$ is the wave vector, d_1 and d_2 are complex numbers such that $|d_1|^2 + |d_2|^2 = 1$, and the real parameter $\mathcal{E}_0$ determines the amplitude of the wave. The wave (C.2a) can be expressed in the equivalent form

$$\boldsymbol{\mathcal{E}}(\mathbf{r}, t) = \mathcal{E}_0 \,\mathrm{Re}\left\{\boldsymbol{\zeta} \exp[\mathrm{i}(\mathbf{k}\cdot\mathbf{r} - \omega t)]\right\}. \tag{C.2b}$$

where the polarisation vector $\boldsymbol{\zeta}$ is complex and has unit magnitude, *i.e.*, $\boldsymbol{\zeta}^*\cdot\boldsymbol{\zeta} = 1$. With all generality, we can write (apart from an irrelevant phase factor)

$$\boldsymbol{\zeta} = \cos(\alpha/2)\hat{\boldsymbol{\epsilon}}_1 + \sin(\alpha/2)\exp(\mathrm{i}\beta)\hat{\boldsymbol{\epsilon}}_2 \tag{C.3}$$

with $\alpha \in (0, \pi)$ and $\beta \in (0, 2\pi)$. The plane wave (C.2b) is elliptically polarised. When $\beta = 0$, the polarisation is linear; the electric field of the wave oscillates in a direction that forms an angle $\alpha/2$ with $\hat{\boldsymbol{\epsilon}}_1$. If $\alpha = \pi/2$ and $\beta = \pm\pi/2$, we have right-hand circular polarisation $(+)$ and left-hand circular polarisation $(-)$.

To facilitate the simulation of interactions of polarised electromagnetic radiation, and to avoid ambiguities, for each direction of propagation $\hat{\mathbf{k}}$ we define a "natural" basis of polarisation vectors as follows. Let ϑ and φ denote the polar and azimuthal angles of the direction $\hat{\mathbf{k}}$, that is,

$$\hat{\mathbf{k}} = \begin{pmatrix} \cos\varphi \sin\vartheta \\ \sin\varphi \sin\vartheta \\ \cos\vartheta \end{pmatrix}. \tag{C.4}$$

A rotation of angle ϑ about the $\hat{\mathbf{y}}$ axis followed by a rotation of angle φ about the $\hat{\mathbf{z}}$ axis transforms the vector $\hat{\mathbf{z}}$ into $\hat{\mathbf{k}}$. The matrix of this rotation is

$$\begin{aligned} \mathcal{R}(\vartheta, \varphi) \;&=\; \mathcal{R}(\varphi\hat{\mathbf{z}})\mathcal{R}(\vartheta\hat{\mathbf{y}}) = \begin{pmatrix} \cos\varphi & -\sin\varphi & 0 \\ \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\vartheta & 0 & \sin\vartheta \\ 0 & 1 & 0 \\ -\sin\vartheta & 0 & \cos\vartheta \end{pmatrix} \\ &= \begin{pmatrix} \cos\varphi\cos\vartheta & -\sin\varphi & \cos\varphi\sin\vartheta \\ \sin\varphi\cos\vartheta & \cos\varphi & \sin\varphi\sin\vartheta \\ -\sin\vartheta & 0 & \cos\vartheta \end{pmatrix}. \end{aligned} \tag{C.5}$$

We define the polarisation vectors $\hat{\boldsymbol{\epsilon}}_1(\hat{\mathbf{k}})$ y $\hat{\boldsymbol{\epsilon}}_2(\hat{\mathbf{k}})$ corresponding to the propagation direction $\hat{\mathbf{k}}$ as those that result from rotating the unit vectors $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ (see Fig. C.1), *i.e.*,

$$\hat{\boldsymbol{\epsilon}}_1(\hat{\mathbf{k}}) \equiv \mathcal{R}(\vartheta, \varphi)\hat{\mathbf{x}} = \begin{pmatrix} \cos\varphi\cos\vartheta \\ \sin\varphi\cos\vartheta \\ -\sin\vartheta \end{pmatrix} \tag{C.6a}$$

and

$$\hat{\boldsymbol{\epsilon}}_2(\hat{\mathbf{k}}) \equiv \mathcal{R}(\vartheta,\varphi)\hat{\mathbf{y}} = \begin{pmatrix} -\sin\varphi \\ \cos\varphi \\ 0 \end{pmatrix}. \tag{C.6b}$$

Evidently,

$$\hat{\boldsymbol{\epsilon}}_1(\hat{\mathbf{k}})\times\hat{\boldsymbol{\epsilon}}_2(\hat{\mathbf{k}}) = \hat{\mathbf{k}} \qquad \text{and} \qquad \hat{\boldsymbol{\epsilon}}_1(\hat{\mathbf{k}})\cdot\hat{\boldsymbol{\epsilon}}_2(\hat{\mathbf{k}}) = 0.$$

Note that $\hat{\mathbf{z}}\times\hat{\mathbf{k}} = \sin\vartheta\hat{\boldsymbol{\epsilon}}_2(\hat{\mathbf{k}})$. That is, $\hat{\boldsymbol{\epsilon}}_2(\hat{\mathbf{k}})$ is perpendicular to the plane that contains the vectors $\hat{\mathbf{z}}$ and $\hat{\mathbf{k}}$. Therefore, the vector $\hat{\boldsymbol{\epsilon}}_1(\hat{\mathbf{k}})$ lies in that plane. To simplify the writing of formulas, in what follows we will use the notation $\hat{\boldsymbol{\epsilon}}_1$ and $\hat{\boldsymbol{\epsilon}}_2$ to designate the polarisation vectors, by removing the explicit reference to the propagation direction $\hat{\mathbf{k}}$, which is implied by the context.

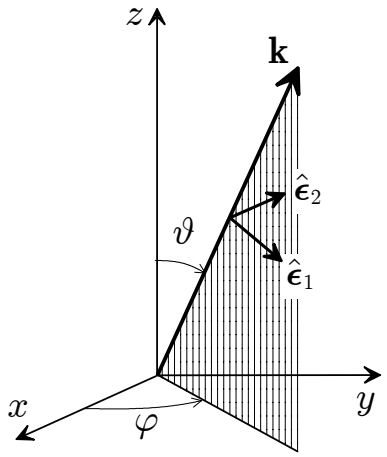

Figure C.1: Vectors $\hat{\boldsymbol{\epsilon}}_1(\hat{\mathbf{k}})$ and $\hat{\boldsymbol{\epsilon}}_2(\hat{\mathbf{k}})$ of the "natural" basis of linear polarisation for waves that propagate in the direction $\hat{\mathbf{k}}$.

In quantum mechanics, a classical wave (C.2) corresponds to a beam of photons in a pure polarisation state. In the basis $\{|\hat{\boldsymbol{\epsilon}}_1\rangle, |\hat{\boldsymbol{\epsilon}}_2\rangle\}$ of linear polarisation states, the state of a photon is expressed as a linear superposition of the basis states, with complex coefficients and unit norm,

$$|\boldsymbol{\zeta}\rangle = d_1|\hat{\boldsymbol{\epsilon}}_1\rangle + d_2|\hat{\boldsymbol{\epsilon}}_2\rangle = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}, \qquad |d_1|^2 + |d_2|^2 = 1. \tag{C.7a}$$

The quantities $|d_1|^2$ and $|d_2|^2$ are the probabilities that in a measurement of the polarisation we find the photon in the states $|\hat{\boldsymbol{\epsilon}}_1\rangle$ and $|\hat{\boldsymbol{\epsilon}}_2\rangle$, respectively. There is a one-to-one correspondence between polarisation states $|\boldsymbol{\zeta}\rangle$ and the polarisation vectors of classical waves, $\boldsymbol{\zeta} = d_1\hat{\boldsymbol{\epsilon}}_1 + d_2\hat{\boldsymbol{\epsilon}}_2$. Hereafter we write $\boldsymbol{\zeta}$ and $\hat{\boldsymbol{\epsilon}}_i$ instead of $|\boldsymbol{\zeta}\rangle$ and $|\hat{\boldsymbol{\epsilon}}_i\rangle$, *i.e.*, we use the same symbol to designate the quantum polarisation states and the unit polarisation vectors.

We will see that the quantum description of photon polarisation is mathematically identical (isomorphic) to that of the spin orientation of a spin-1/2 particle. An arbitrary polarisation state can be expressed in the form

$$\boldsymbol{\zeta} = \cos(\alpha/2)\hat{\boldsymbol{\epsilon}}_1 + \sin(\alpha/2)\exp(\mathrm{i}\beta)\hat{\boldsymbol{\epsilon}}_2 = \begin{pmatrix} \cos(\alpha/2) \\ \sin(\alpha/2)\exp(\mathrm{i}\beta) \end{pmatrix} \tag{C.7b}$$

with $\alpha \in (0, \pi)$ and $\beta \in (0, 2\pi)$. States with $\beta = 0$ correspond to linear polarisation in a direction that makes an angle $\alpha/2$ with $\hat{\boldsymbol{\epsilon}}_1$. If $\beta = \pm\pi/2$ and $\alpha = \pi/2$, we have right-hand (+) and left-hand (−) circularly polarised photons.

C.2 Density matrix and Stokes parameters

Usually, photons in a real beam cannot be assigned a definite polarisation state because a complete quantum description of the beam is not available. For instance, when a beam of photons in a pure polarisation state is scattered by the atoms of a gas, the scattered beam can be expressed as a linear combination of pure states, but with coefficients d_1 and d_2 that depend on the state of the target atom after the interaction. Such photon beams are said to have *partial polarisation* and can be described by using the density matrix formalism (Falkoff and MacDonald, 1951; Fano, 1954a; McMaster, 1954).

Let us consider a partially polarised beam, whose photons are in certain states $\boldsymbol{\zeta}_n$ of pure polarisation with corresponding probabilities p_n. The occupied states $\boldsymbol{\zeta}_n$ are only assumed to be normalised to unity, the number and nature of these states are arbitrary. The probabilities p_n are positive and add to unity,

$$\sum_n p_n = 1. \tag{C.8}$$

The density operator for such a radiation beam is defined as

$$\rho = \sum_n |\boldsymbol{\zeta}_n\rangle\, p_n \langle\boldsymbol{\zeta}_n|. \tag{C.9}$$

The matrix elements of the density operator in the basis of states of linear polarisation $\{\hat{\boldsymbol{\epsilon}}_1, \hat{\boldsymbol{\epsilon}}_2\}$ are

$$\rho_{\mu\nu} = \sum_n \langle\hat{\boldsymbol{\epsilon}}_\mu|\boldsymbol{\zeta}_n\rangle\, p_n \langle\boldsymbol{\zeta}_n|\hat{\boldsymbol{\epsilon}}_\nu\rangle \qquad (\mu, \nu = 1, 2). \tag{C.10}$$

If the beam photons are in a single pure state, Eq. (C.7a), the density matrix reduces to

$$\rho = |\boldsymbol{\zeta}\rangle\langle\boldsymbol{\zeta}| \qquad \text{and} \qquad \rho_{\mu\nu} = d_\mu d_\nu^*, \tag{C.11}$$

that is,

$$\rho = \begin{pmatrix} d_1 d_1^* & d_1 d_2^* \\ d_2 d_1^* & d_2 d_2^* \end{pmatrix}. \tag{C.12}$$

The density matrix is Hermitian,

$$\rho^\dagger = \rho, \qquad \rho_{\mu\nu} = \rho^*_{\nu\mu}, \tag{C.13}$$

and has unit trace,

$$\mathrm{Tr}\rho = \rho_{11} + \rho_{22} = \sum_n p_n = 1. \tag{C.14}$$

Hence ρ is determined by three real parameters. Since ρ is a 2×2 Hermitian matrix, it can be expressed as a linear combination of the Pauli matrices,

$$\sigma_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad \sigma_2 = \begin{pmatrix} 0 & -\mathrm{i} \\ \mathrm{i} & 0 \end{pmatrix}, \qquad \sigma_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \tag{C.15}$$

and the 2×2 identity matrix, with real coefficients,

$$\rho = \frac{1}{2}\left(I_2 + P_1\sigma_1 + P_2\sigma_2 + P_3\sigma_3\right) = \frac{1}{2}\begin{pmatrix} 1+P_3 & P_1 - \mathrm{i}P_2 \\ P_1 + \mathrm{i}P_2 & 1 - P_3 \end{pmatrix}. \tag{C.16}$$

The quantities P_i are the *Stokes parameters*, which are important because they provide a complete description of the polarisation of a beam and, as we will show below, they can be measured experimentally. From the properties of the Pauli matrices, we have

$$\mathrm{Tr}\rho\sigma_i = P_i. \tag{C.17}$$

This equality defines the Stokes parameters in terms of the density matrix elements $\rho_{\mu\nu}$,

$$P_1 = \rho_{12} + \rho_{21}, \qquad P_2 = \mathrm{i}(\rho_{12} - \rho_{21}), \qquad P_3 = \rho_{11} - \rho_{22}. \tag{C.18}$$

We can introduce the *Poincaré* vector, $\mathbf{P} \equiv (P_1, P_2, P_3)$, and write

$$\rho = \frac{1}{2}\left(I_2 + \mathbf{P}\cdot\boldsymbol{\sigma}\right). \tag{C.19}$$

The Poincaré vector is analogous to the direction of spin of a spin-$\frac{1}{2}$ particle. However, the spin direction belongs to ordinary space (because it transforms as a common vector under rotations), while $\mathbf{P}$ is in a different vector space (because its components transform in a special form under space rotations, see Section C.2.1). The determinant of the density matrix is

$$\det\rho = \frac{1}{2}(1 - P^2), \tag{C.20}$$

with $P^2 = P_1^2 + P_2^2 + P_3^2$.

If the Stokes parameters vanish, $P = 0$, we have $\rho = \frac{1}{2}I_2$. That is, all polarisation directions are equivalent. A density matrix of this type represents unpolarised photons.

The opposite case is that of a pure state (α, β), Eq. (C.7b), which has the following density matrix,

$$\rho = \begin{pmatrix} \cos^2(\alpha/2) & \cos(\alpha/2)\sin(\alpha/2)\exp(-\mathrm{i}\beta) \\ \cos(\alpha/2)\sin(\alpha/2)\exp(\mathrm{i}\beta) & \sin^2(\alpha/2) \end{pmatrix}. \tag{C.21}$$

The corresponding Stokes parameters are given by expressions (C.18),

$$P_1 = \sin\alpha\,\cos\beta, \qquad P_2 = \sin\alpha\,\sin\beta, \qquad P_3 = \cos\alpha. \tag{C.22}$$

Note that α and β are the polar and azimuthal angles of the Poincaré vector $\mathbf{P}$, and that

$$P^2 = P_1^2 + P_2^2 + P_3^2 = 1. \tag{C.23}$$

It is worth mentioning that in optics the Pauli matrices are usually given in a different order, namely, $\{\sigma_3, \sigma_1, \sigma_2\}$. We prefer the ordering (C.15) employed in quantum mechanics, not only because it allows the geometrical interpretation of the state angles α and β expressed by Eqs. (C.22) but also because the formalism is parallel to that of polarisation of spin-$\frac{1}{2}$ particles, which is studied in elementary quantum mechanics courses.

It is clear that P, the magnitude of the Poincaré vector, can take values from 0 (unpolarised photons) to 1 (pure polarisation states). It is natural to refer to P as the *polarisation degree* of the photons. The endpoints of the Poincaré vectors fill the interior of the unit sphere, the so-called *Poincaré sphere*. The pure states (complete polarisation) have the endpoints of their Poincaré vectors on the surface of that sphere.

In the case of pure states ($P = 1$), inverting the relations (C.22), we can obtain the state angles (α, β) from the Stokes parameters,

$$\alpha = \arccos P_3, \qquad \exp(\mathrm{i}\beta) = \frac{P_1 + \mathrm{i}P_2}{\sqrt{1 - P_3^2}}. \tag{C.24}$$

The pure states corresponding to the Poincaré vectors $\mathbf{P}$ and $-\mathbf{P}$, with respective directions (α, β) and $(\pi - \alpha, \beta + \pi)$, are

$$\boldsymbol{\zeta}(\mathbf{P}) \equiv \hat{\boldsymbol{\epsilon}}(\alpha, \beta) = \begin{pmatrix} \cos(\alpha/2) \\ \sin(\alpha/2)\exp(\mathrm{i}\beta) \end{pmatrix}, \tag{C.25a}$$

$$\boldsymbol{\zeta}(-\mathbf{P}) \equiv \hat{\boldsymbol{\epsilon}}(\pi - \alpha, \beta + \pi) = \begin{pmatrix} \sin(\alpha/2) \\ -\cos(\alpha/2)\exp(\mathrm{i}\beta) \end{pmatrix}. \tag{C.25b}$$

Note that these states are orthogonal,

$$\langle \boldsymbol{\zeta}(\mathbf{P}) |\, \boldsymbol{\zeta}(-\mathbf{P}) \rangle = 0. \tag{C.26}$$

Hence, by reversing the signs of the Stokes parameters of a pure state, we obtain its orthogonal state (except, possibly, an irrelevant phase factor). Thus, the state angles

(α, β), the Poincaré vectors and the density matrices of the states $\{\hat{\boldsymbol{\epsilon}}_1, \hat{\boldsymbol{\epsilon}}_2\}$ of the linear-polarisation basis are

$$\hat{\boldsymbol{\epsilon}}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} : \quad \begin{matrix} \alpha = 0, \quad \beta = 0, \\ \mathbf{P}(\hat{\boldsymbol{\epsilon}}_1) = (0, 0, 1), \end{matrix} \quad \rho(\hat{\boldsymbol{\epsilon}}_1) = \frac{1}{2} \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix}, \tag{C.27a}$$

$$\hat{\boldsymbol{\epsilon}}_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} : \quad \begin{matrix} \alpha = \pi, \quad \beta = \pi, \\ \mathbf{P}(\hat{\boldsymbol{\epsilon}}_2) = (0, 0, -1), \end{matrix} \quad \rho(\hat{\boldsymbol{\epsilon}}_2) = \frac{1}{2} \begin{pmatrix} 0 & 0 \\ 0 & 2 \end{pmatrix}. \tag{C.27b}$$

Similarly, for the states of the basis of circular polarisation, we have

$$\hat{\boldsymbol{\epsilon}}^{(\mathrm{r})} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ \mathrm{i} \end{pmatrix} : \quad \begin{matrix} \alpha = \pi/2, \quad \beta = \pi/2, \\ \mathbf{P}(\hat{\boldsymbol{\epsilon}}^{(\mathrm{r})}) = (0, 1, 0), \end{matrix} \quad \rho(\hat{\boldsymbol{\epsilon}}^{(\mathrm{r})}) = \frac{1}{2} \begin{pmatrix} 1 & -\mathrm{i} \\ \mathrm{i} & 1 \end{pmatrix}, \tag{C.28a}$$

$$\hat{\boldsymbol{\epsilon}}^{(\mathrm{l})} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -\mathrm{i} \end{pmatrix} : \quad \begin{matrix} \alpha = \pi/2, \quad \beta = -\pi/2, \\ \mathbf{P}(\hat{\boldsymbol{\epsilon}}^{(\mathrm{l})}) = (0, -1, 0), \end{matrix} \quad \rho(\hat{\boldsymbol{\epsilon}}^{(\mathrm{l})}) = \frac{1}{2} \begin{pmatrix} 1 & \mathrm{i} \\ -\mathrm{i} & 1 \end{pmatrix}. \tag{C.28b}$$

As a third example, to which we will refer below, we consider the basis of polarisation states obtained by rotating the vectors of the basis $\{\hat{\boldsymbol{\epsilon}}_1, \hat{\boldsymbol{\epsilon}}_2\}$ an angle of 45 degrees about the propagation direction $\hat{\mathbf{k}}$,

$$\hat{\boldsymbol{\epsilon}}_1^{(45)} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} : \quad \begin{matrix} \alpha = \pi/2, \quad \beta = 0, \\ \mathbf{P}(\hat{\boldsymbol{\epsilon}}_1^{(45)}) = (1, 0, 0), \end{matrix} \quad \rho(\hat{\boldsymbol{\epsilon}}_1^{(45)}) = \frac{1}{2} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, \tag{C.29a}$$

$$\hat{\boldsymbol{\epsilon}}_2^{(45)} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} : \quad \begin{matrix} \alpha = \pi/2, \quad \beta = \pi, \\ \mathbf{P}(\hat{\boldsymbol{\epsilon}}_2^{(45)}) = (-1, 0, 0), \end{matrix} \quad \rho(\hat{\boldsymbol{\epsilon}}_2^{(45)}) = \frac{1}{2} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}. \tag{C.29b}$$

The *intensity* of a photon beam is defined as the number of photons per unit volume in the beam. Let us consider two photon beams with intensities N_1 and N_2 and respective density matrices ρ_1 and ρ_2. The incoherent admixture of these two beams gives a beam with intensity $N = N_1 + N_2$ and density matrix

$$\rho = \frac{N_1}{N}\, \rho_1 + \frac{N_2}{N}\, \rho_2. \tag{C.30a}$$

The corresponding Stokes parameters are

$$\mathbf{P} = \frac{N_1}{N}\, \mathbf{P}_1 + \frac{N_2}{N}\, \mathbf{P}_2, \tag{C.30b}$$

where $\mathbf{P}_1$ and $\mathbf{P}_2$ are the Poincaré vectors of the initial beams.

A partially polarised beam with Stokes parameters $\mathbf{P}$ ($P < 1$) can be regarded as an incoherent admixture of an unpolarised beam and a completely polarised beam. To characterise these beams, we define the reduced Stokes parameters

$$P_i' \equiv P_i / P. \tag{C.31}$$

The matrix density of the partially polarised beam **P** can then be expressed as

$$\begin{aligned}\rho &= \frac{1}{2}\begin{pmatrix} 1+P_3 & P_1-\mathrm{i}P_2 \\ P_1+\mathrm{i}P_2 & 1-P_3 \end{pmatrix} \\ &= (1-P)\frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + P\frac{1}{2}\begin{pmatrix} 1+P'_3 & P'_1-\mathrm{i}P'_2 \\ P'_1+\mathrm{i}P'_2 & 1-P'_3 \end{pmatrix}. \end{aligned} \tag{C.32}$$

The second term in the last expression describes a completely polarised beam, because $P'^2_1 + P'^2_2 + P'^2_3 = 1$. Hence, the result of mixing an unpolarised beam and a completely polarised beam $\mathbf{P}'$, with relative intensities $(1-P)$ and P, respectively, has the same density matrix as the original partially polarised beam. Therefore, any polarisation measurement on the mixed beam will yield the same results as for the original beam ρ (see below).

The density matrices of beams in pure polarisation states with Poincaré vectors **P** and $-\mathbf{P}$ ($P = 1$) are

$$\rho(\mathbf{P}) = \frac{1}{2}\begin{pmatrix} 1+P_3 & P_1-\mathrm{i}P_2 \\ P_1+\mathrm{i}P_2 & 1-P_3 \end{pmatrix} \qquad \text{and} \qquad \rho(-\mathbf{P}) = \frac{1}{2}\begin{pmatrix} 1-P_3 & -P_1+\mathrm{i}P_2 \\ -P_1-\mathrm{i}P_2 & 1+P_3 \end{pmatrix}, \tag{C.33}$$

respectively. The density matrix of the beam obtained by mixing these two beams, with equal intensities, is

$$\rho = \frac{1}{2}\rho(\mathbf{P}) + \frac{1}{2}\rho(-\mathbf{P}) = \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \tag{C.34}$$

which is the density matrix of an unpolarised beam. This result implies that an unpolarised beam can be regarded as an admixture of two completely polarised beams with equal intensities and "opposite" polarisations **P** and $-\mathbf{P}$, independently of the direction of **P**.

C.2.1 Rotations of the base vectors

A unitary[1] transformation U in the space of polarisation states (*i.e.*, a change of basis, $\{\hat{\boldsymbol{\epsilon}}_1, \hat{\boldsymbol{\epsilon}}_2\}$) induces a transformation of Poincaré vectors and density matrices. The transformed density matrix is $\rho' = U\rho U^{-1}$ and, from Eq. (C.20),

$$P'^2 = 1 - 4\det\rho' = 1 - 4\det\rho = P^2. \tag{C.35}$$

That is, the transformation leaves the length of the Poincaré vectors (the degree of polarisation) unaltered. As we are going to show, rotations of the unit polarisation vectors induce rotations of the Poincaré vectors.

[1]We recall that a 2×2 matrix U is unitary if $U^\dagger = U^{-1}$, *i.e.*, if its Hermitian conjugate $U^\dagger$ coincides with its inverse U^{-1}.

Let us assume that we change the zero-azimuth plane by an angle ϕ, *i.e.*, we rotate the vectors of the basis of linear polarisation an angle ϕ about the $\hat{\mathbf{k}}$ axis. The vectors of the new basis are

$$\begin{aligned}\hat{\boldsymbol{\epsilon}}'_1 &= \mathcal{R}(\phi\hat{\mathbf{k}})\hat{\boldsymbol{\epsilon}}_1 = \cos\phi\,\hat{\boldsymbol{\epsilon}}_1 + \sin\phi\,\hat{\boldsymbol{\epsilon}}_2,\\ \hat{\boldsymbol{\epsilon}}'_2 &= \mathcal{R}(\phi\hat{\mathbf{k}})\hat{\boldsymbol{\epsilon}}_2 = -\sin\phi\,\hat{\boldsymbol{\epsilon}}_1 + \cos\phi\,\hat{\boldsymbol{\epsilon}}_2.\end{aligned} \tag{C.36}$$

The rotation matrix has elements $\mathcal{R}_{ij}(\phi\hat{\mathbf{k}}) = \left\langle \hat{\boldsymbol{\epsilon}}_i | \hat{\boldsymbol{\epsilon}}'_j \right\rangle$, *i.e.*,

$$\mathcal{R}(\phi\hat{\mathbf{k}}) = \begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix} = \cos\phi I_2 - \mathrm{i}\sin\phi\sigma_2. \tag{C.37}$$

The matrix of the inverse rotation is

$$\mathcal{R}(-\phi\hat{\mathbf{k}}) = \mathcal{R}^{-1}(\phi\hat{\mathbf{k}}) = \begin{pmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{pmatrix} = \cos\phi I_2 + \mathrm{i}\sin\phi\sigma_2. \tag{C.38}$$

Note that $\mathcal{R}^{-1}(\phi\hat{\mathbf{k}}) = \mathcal{R}^{\dagger}(\phi\hat{\mathbf{k}})$. The coefficients of the expansion of the polarisation state $\boldsymbol{\zeta} = d_i\hat{\boldsymbol{\epsilon}}_i$ in the new basis are

$$d'_j = \left\langle \hat{\boldsymbol{\epsilon}}'_j \,|\boldsymbol{\zeta}\right\rangle = \left\langle \mathcal{R}(\phi\hat{\mathbf{k}})\hat{\boldsymbol{\epsilon}}_j \middle| \, d_i\hat{\boldsymbol{\epsilon}}_i \right\rangle = \left\langle \hat{\boldsymbol{\epsilon}}_j \middle| \mathcal{R}(-\phi\hat{\mathbf{k}}) \middle| \, \hat{\boldsymbol{\epsilon}}_i \right\rangle d_i = \mathcal{R}_{ji}(-\phi\hat{\mathbf{k}})d_i, \tag{C.39}$$

Or, more explicitly,

$$d'_1 = \cos\phi\, d_1 + \sin\phi\, d_2, \qquad d'_2 = -\sin\phi\, d_1 + \cos\phi\, d_2. \tag{C.40}$$

We see that the transformation (C.36) is equivalent to rotating the electromagnetic field an angle $-\phi$ about the $\hat{\mathbf{k}}$ axis, leaving the orientation of the axes fixed.

Let us consider a beam with the density matrix $\rho_{\mu\nu}$. In the new polarisation basis, the density matrix of this beam is

$$\begin{aligned}\rho'_{\mu\nu} &= \sum_n \langle\hat{\boldsymbol{\epsilon}}'_\mu|\boldsymbol{\zeta}_n\rangle\, p_n \langle\boldsymbol{\zeta}_n|\hat{\boldsymbol{\epsilon}}'_\nu\rangle = \sum_n \sum_{i,j} \langle\mathcal{R}(\phi\hat{\mathbf{k}})\hat{\boldsymbol{\epsilon}}_\mu|\hat{\boldsymbol{\epsilon}}_i\rangle\langle\hat{\boldsymbol{\epsilon}}_i|\boldsymbol{\zeta}_n\rangle\, p_n \langle\boldsymbol{\zeta}_n|\hat{\boldsymbol{\epsilon}}_j\rangle\langle\hat{\boldsymbol{\epsilon}}_j|\mathcal{R}(\phi\hat{\mathbf{k}})\hat{\boldsymbol{\epsilon}}_\nu\rangle\\ &= \sum_n \sum_{i,j} \langle\hat{\boldsymbol{\epsilon}}_\mu|\mathcal{R}(-\phi\hat{\mathbf{k}})|\hat{\boldsymbol{\epsilon}}_i\rangle\langle\hat{\boldsymbol{\epsilon}}_i|\boldsymbol{\zeta}_n\rangle\, p_n \langle\boldsymbol{\zeta}_n|\hat{\boldsymbol{\epsilon}}_j\rangle\langle\hat{\boldsymbol{\epsilon}}_j|\mathcal{R}(\phi\hat{\mathbf{k}})|\hat{\boldsymbol{\epsilon}}_\nu\rangle.\end{aligned} \tag{C.41}$$

That is,

$$\begin{aligned}\rho' &= \mathcal{R}(-\phi\hat{\mathbf{k}})\rho\mathcal{R}(\phi\hat{\mathbf{k}}) = (\cos\phi I_2 + \mathrm{i}\sin\phi\sigma_2)\frac{1}{2}\left(I_2 - \mathbf{P}\cdot\boldsymbol{\sigma}\right)(\cos\phi I_2 - \mathrm{i}\sin\phi\sigma_2)\\ &= \frac{1}{2}\left\{I_2 + [-\sin(2\phi)P_3 + \cos(2\phi)P_1]\sigma_1 + P_2\sigma_2 + [\cos(2\phi)P_3 + \sin(2\phi)P_1]\sigma_3\right\}.\end{aligned} \tag{C.42}$$

Hence,

$$P'_3 = \cos(2\phi)P_3 + \sin(2\phi)P_1$$

$$P_1' = -\sin(2\phi)P_3 + \cos(2\phi)P_1. \tag{C.43}$$

Thus, a rotation of the zero-azimuth plane by an angle ϕ is equivalent to a rotation of the photon polarisation state by an angle $-\phi$ and to a rotation of the Stokes parameters P_3 and P_1 by an angle -2ϕ. In the case $\phi = \pi$, we have $P_3' = P_3$ and $P_1' = P_1$. This feature is consistent with the fact that linearly polarised waves with polarisation vectors $\boldsymbol{\zeta}$ and $-\boldsymbol{\zeta}$ correspond to the same polarisation state. Note that $P_2' = P_2$, because rotations about the $\hat{\mathbf{k}}$ axis do not affect the states of circular polarisation.

C.3 Polarisation analysers

The polarisation of a photon beam can be measured using polarisation filters, which transmit light of various polarisations with different efficiencies. An ideal filter is completely transparent for a certain polarisation state and absolutely opaque to the orthogonal state, working similarly to a Stern-Gerlach analyser with one of the output channels (spin values) blocked. Let us consider an ideal filter that transmits photons in the state $\boldsymbol{\zeta}(\mathbf{Q})$, corresponding to the Poincaré vector $\mathbf{Q}$, and absorbs photons in the state $\boldsymbol{\zeta}(-\mathbf{Q})$ of opposite polarisation, $-\mathbf{Q}$. This filter can be considered as a measuring apparatus with an associated quantum operator (observable) $\boldsymbol{\mathcal{F}}[\boldsymbol{\zeta}(\mathbf{Q})]$ that has the eigenstates $\boldsymbol{\zeta}(\mathbf{Q})$ and $\boldsymbol{\zeta}(-\mathbf{Q})$ with eigenvalues 1 and 0, respectively. We recall that $|\langle\boldsymbol{\zeta}(\mathbf{Q})|\boldsymbol{\zeta}(\mathbf{P})\rangle|^2$ is the probability that an incident photon in the pure state $\boldsymbol{\zeta}(\mathbf{P})$ is transmitted through the filter, emerging in the polarisation state $\boldsymbol{\zeta}(\mathbf{Q})$.

To determine the operator $\boldsymbol{\mathcal{F}}[\boldsymbol{\zeta}(\mathbf{Q})]$, we recall that the pure state $\hat{\boldsymbol{\epsilon}}_1$ corresponds to the Poincaré vector $\mathbf{Q} = (0, 0, 1)$, see Eq. (C.27). Then, if a polariser transmits photons with polarisation $\hat{\boldsymbol{\epsilon}}_1$, we must have

$$\boldsymbol{\mathcal{F}}(\hat{\boldsymbol{\epsilon}}_1) = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}. \tag{C.44}$$

This matrix can also be expressed as

$$\boldsymbol{\mathcal{F}}(\hat{\boldsymbol{\epsilon}}_1) = \frac{1}{2}\left[I_2 + \sigma_3\right] = \frac{1}{2}\left[I_2 + \mathbf{Q}\cdot\boldsymbol{\sigma}\right]. \tag{C.45}$$

Now, the operator $\boldsymbol{\mathcal{F}}[\boldsymbol{\zeta}(\mathbf{Q})]$ for any other polarisation direction $\mathbf{Q}$ can be obtained by means of a unitary transformation of $\{\hat{\boldsymbol{\epsilon}}_1, \hat{\boldsymbol{\epsilon}}_2\}$. Therefore, the most general form of that operator is

$$\boldsymbol{\mathcal{F}}[\boldsymbol{\zeta}(\mathbf{Q})] = \frac{1}{2}\left[I_2 + \mathbf{Q}\cdot\boldsymbol{\sigma}\right]. \tag{C.46}$$

Using this formula, we see that the operators of polarisation filters that transmit photons with the polarisations $\hat{\boldsymbol{\epsilon}}^{(\mathrm{r})}$ and $\hat{\boldsymbol{\epsilon}}_1^{(45)}$ are [see Eqs. (C.28a) and (C.29a)]

$$\boldsymbol{\mathcal{F}}(\hat{\boldsymbol{\epsilon}}^{(\mathrm{r})}) = \begin{pmatrix} 1 & -\mathrm{i} \\ \mathrm{i} & 1 \end{pmatrix} \tag{C.47}$$

and

$$\boldsymbol{\mathcal{F}}(\hat{\boldsymbol{\epsilon}}_1^{(45)}) = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, \tag{C.48}$$

respectively. It can be easily verified that, indeed, the eigenstates of these operators are those of the bases $\{\hat{\boldsymbol{\epsilon}}^{(\mathrm{r})}, \hat{\boldsymbol{\epsilon}}^{(\mathrm{l})}\}$ and $\{\hat{\boldsymbol{\epsilon}}_1^{(45)}, \hat{\boldsymbol{\epsilon}}_2^{(45)}\}$ defined above.

Note that the matrix of the filter operator $\boldsymbol{\mathcal{F}}[\boldsymbol{\zeta}(\mathbf{Q})]$ is equal to the density matrix of a pure state with Poincaré vector $\mathbf{Q}$, Eq. (C.19),

$$\boldsymbol{\mathcal{F}}[\boldsymbol{\zeta}(\mathbf{Q})] = \rho(\mathbf{Q}). \tag{C.49}$$

Furthermore, $\boldsymbol{\mathcal{F}}^2[\boldsymbol{\zeta}(\mathbf{Q})] = \boldsymbol{\mathcal{F}}[\boldsymbol{\zeta}(\mathbf{Q})]$, that is, the operator of an ideal polariser is a projector. Therefore,

$$\boldsymbol{\mathcal{F}}[\boldsymbol{\zeta}(\mathbf{Q})] = |\boldsymbol{\zeta}(\mathbf{Q})\rangle\langle\boldsymbol{\zeta}(\mathbf{Q})|. \tag{C.50}$$

Evidently,

$$\boldsymbol{\mathcal{F}}[\boldsymbol{\zeta}(\mathbf{Q})] + \boldsymbol{\mathcal{F}}[\boldsymbol{\zeta}(-\mathbf{Q})] = |\boldsymbol{\zeta}(\mathbf{Q})\rangle\langle\boldsymbol{\zeta}(\mathbf{Q})| + |\boldsymbol{\zeta}(-\mathbf{Q})\rangle\langle\boldsymbol{\zeta}(-\mathbf{Q})| = I_2. \tag{C.51}$$

The transmission probability of photons in a pure state $\boldsymbol{\zeta}(\mathbf{P})$, with Poincaré vector $\mathbf{P}$ ($P = 1$) and density matrix $\rho(\mathbf{P})$, through the polariser $\boldsymbol{\mathcal{F}}[\boldsymbol{\zeta}(\mathbf{Q})]$ is

$$\begin{aligned} \langle\boldsymbol{\mathcal{F}}[\boldsymbol{\zeta}(\mathbf{Q})]\rangle &\equiv \langle\boldsymbol{\zeta}(\mathbf{P})|\boldsymbol{\mathcal{F}}[\boldsymbol{\zeta}(\mathbf{Q})]|\boldsymbol{\zeta}(\mathbf{P})\rangle \\ &= \sum_{i,j}\langle\boldsymbol{\zeta}(\mathbf{P})|\hat{\boldsymbol{\epsilon}}_i\rangle\,\langle\hat{\boldsymbol{\epsilon}}_i|\boldsymbol{\mathcal{F}}[\boldsymbol{\zeta}(\mathbf{Q})]|\hat{\boldsymbol{\epsilon}}_j\rangle\,\langle\hat{\boldsymbol{\epsilon}}_j|\boldsymbol{\zeta}(\mathbf{P})\rangle \\ &= \sum_{i,j}\langle\hat{\boldsymbol{\epsilon}}_i|\boldsymbol{\mathcal{F}}[\boldsymbol{\zeta}(\mathbf{Q})]|\hat{\boldsymbol{\epsilon}}_j\rangle\,\langle\hat{\boldsymbol{\epsilon}}_j|\boldsymbol{\zeta}(\mathbf{P})\rangle\,\langle\boldsymbol{\zeta}(\mathbf{P})|\hat{\boldsymbol{\epsilon}}_i\rangle \\ &= \mathrm{Tr}\{\boldsymbol{\mathcal{F}}[\boldsymbol{\zeta}(\mathbf{Q})]\,\rho(\mathbf{P})\}. \end{aligned} \tag{C.52}$$

In the case of a partially polarised beam, with Poincaré vector $\mathbf{P}$ ($P \leq 1$), the transmission probability can be readily obtained from the decomposition (C.32). The unpolarised component is transmitted with probability 1/2, while the transmission probability of the completely polarised component is given by (C.52). Hence,

$$\begin{aligned} \langle\boldsymbol{\mathcal{F}}[\boldsymbol{\zeta}(\mathbf{Q})]\rangle &= (1-P)\,\frac{1}{2} + P\,\mathrm{Tr}\{\boldsymbol{\mathcal{F}}[\boldsymbol{\zeta}(\mathbf{Q})]\,\rho(\mathbf{P}')\} \\ &= \mathrm{Tr}\{\boldsymbol{\mathcal{F}}[\boldsymbol{\zeta}(\mathbf{Q})]\,\rho(\mathbf{P})\}. \end{aligned} \tag{C.53}$$

This result can also be derived, in a more general fashion, from elementary considerations on the density matrix, Eq. (C.9) (Sakurai, 1997; Ballentine, 1998). Using the relation

$$(\mathbf{Q}\cdot\boldsymbol{\sigma})(\mathbf{P}\cdot\boldsymbol{\sigma}) = \mathbf{Q}\cdot\mathbf{P} + \mathrm{i}\boldsymbol{\sigma}\cdot(\mathbf{Q}\times\mathbf{P}), \tag{C.54}$$

we see that

$$\langle\boldsymbol{\mathcal{F}}[\boldsymbol{\zeta}(\mathbf{Q})]\rangle = \mathrm{Tr}\left\{\frac{1}{2}\,[I_2 + \mathbf{Q}\cdot\boldsymbol{\sigma}]\;\frac{1}{2}\,[I_2 + \mathbf{P}\cdot\boldsymbol{\sigma}]\right\}$$

$$= \frac{1}{2}(1 + \mathbf{Q}\cdot\mathbf{P}). \tag{C.55}$$

That is, if $\mathbf{P}$ is parallel (antiparallel) to $\mathbf{Q}$, the transmission probability is unity (zero). When $\mathbf{P}$ and $\mathbf{Q}$ are mutually orthogonal, the fraction of transmitted photons is 1/2.

Now we are ready to show how the Stokes parameters can be measured experimentally. Let us consider a photon beam in an arbitrary (pure or mixed) polarisation state, described by the Poincaré vector $\mathbf{P}$ ($P \leq 1$), and assume that we perform measurements with ideal analysers. The probability of transmission through a linear filter with polarisation vector $\hat{\boldsymbol{\epsilon}}_1$ [$\mathbf{Q} = (0, 0, 1)$] is

$$\langle \boldsymbol{\mathcal{F}}(\hat{\boldsymbol{\epsilon}}_1)\rangle = \mathrm{Tr}\left\{ \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \frac{1}{2} \begin{pmatrix} 1+P_3 & P_1 - \mathrm{i}P_2 \\ P_1 + \mathrm{i}P_2 & 1-P_3 \end{pmatrix} \right\} = \tfrac{1}{2}(1 + P_3). \tag{C.56a}$$

Similarly, for a linear filter of polarisation $\hat{\boldsymbol{\epsilon}}_1^{(45)}$ [$\mathbf{Q} = (1, 0, 0)$],

$$\langle \boldsymbol{\mathcal{F}}(\hat{\boldsymbol{\epsilon}}_1^{(45)})\rangle = \mathrm{Tr}\left\{ \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \frac{1}{2} \begin{pmatrix} 1+P_3 & P_1 - \mathrm{i}P_2 \\ P_1 + \mathrm{i}P_2 & 1-P_3 \end{pmatrix} \right\} = \tfrac{1}{2}(1 + P_1). \tag{C.56b}$$

Finally, the transmission probability through a right-hand circular polariser [with $\mathbf{Q} = (0, 1, 0)$] is

$$\langle \boldsymbol{\mathcal{F}}(\hat{\boldsymbol{\epsilon}}^{(\mathrm{r})})\rangle = \mathrm{Tr}\left\{ \frac{1}{2}\begin{pmatrix} 1 & -\mathrm{i} \\ \mathrm{i} & 1 \end{pmatrix} \frac{1}{2} \begin{pmatrix} 1+P_3 & P_1 - \mathrm{i}P_2 \\ P_1 + \mathrm{i}P_2 & 1-P_3 \end{pmatrix} \right\} = \tfrac{1}{2}(1 + P_2). \tag{C.56c}$$

That is, to determine the Stokes parameters of a photon beam, we only have to perform measurements with the three ideal polarisers $\hat{\boldsymbol{\epsilon}}_1$, $\hat{\boldsymbol{\epsilon}}_1^{(45)}$ and $\hat{\boldsymbol{\epsilon}}^{(\mathrm{r})}$. The parameter P_3 characterises the polarisation along $\{\hat{\boldsymbol{\epsilon}}_1, \hat{\boldsymbol{\epsilon}}_2\}$: if $P_3 = 1$, the photon is in the pure state $\hat{\boldsymbol{\epsilon}}_1$, while $P_3 = -1$ corresponds to total polarisation in the orthogonal direction $\hat{\boldsymbol{\epsilon}}_2$. The second Stokes parameter describes circular polarisation: $P_2 = 1$ corresponds to the pure state $\hat{\boldsymbol{\epsilon}}^{(\mathrm{r})}$, and $P_2 = -1$ describes photons with left-hand circular polarisation, $\hat{\boldsymbol{\epsilon}}^{(\mathrm{l})}$. Similarly, P_1 measures the polarisation in a direction that makes an angle of 45 degrees with $\hat{\boldsymbol{\epsilon}}_1$. The values $P_1 = 1$ and $P_1 = -1$ correspond to the pure polarisation states $\hat{\boldsymbol{\epsilon}}_1^{(45)}$ and $\hat{\boldsymbol{\epsilon}}_2^{(45)}$, respectively.

From the results in Eqs. (C.56), our ordering of the Stokes parameters, with P_3 measuring linear polarisation in the "natural" directions $\hat{\boldsymbol{\epsilon}}_1$ and $\hat{\boldsymbol{\epsilon}}_2$, may seem awkward. The ordering used in optics, (P_3, P_1, P_2), assigns more relevance to linear polarisations, but alters the relationship (C.22) between the components of the Poincaré vector and the state angles (α, β) obscuring the geometrical interpretation of the latter.

C.3.1 Real analysers

Real polarisation filters have maximum efficiency W_+ (transmission probability, ≤ 1) for pure states corresponding to a certain Poincaré vector $\mathbf{Q}$ ($Q = 1$) and minimum

efficiency W_- for pure states polarised in the opposite direction, $-\mathbf{Q}$. A real filter can be regarded as the "superposition" of two ideal filters with weights W_+ and W_-, and represented by the operator

$$\begin{aligned}\boldsymbol{\mathcal{F}}[\boldsymbol{\zeta}(\mathbf{Q}); W_+, W_-] &\equiv W_+\boldsymbol{\mathcal{F}}[\boldsymbol{\zeta}(\mathbf{Q})] + W_-\boldsymbol{\mathcal{F}}[\boldsymbol{\zeta}(-\mathbf{Q})] \\ &= \frac{1}{2}\left[(W_+ + W_-)I_2 + (W_+ - W_-)\mathbf{Q}\cdot\boldsymbol{\sigma}\right]. \end{aligned} \tag{C.57}$$

The states $|\boldsymbol{\zeta}(\mathbf{Q})\rangle$ and $|\boldsymbol{\zeta}(-\mathbf{Q})\rangle$ are eigenstates of this operator with respective eigenvalues W_+ and W_-. Obviously, the transparent filter (*i.e.*, the filter that transmits all photons) corresponds to the operator

$$\boldsymbol{\mathcal{F}}[\boldsymbol{\zeta}(\mathbf{Q}); 1, 1) = \boldsymbol{\mathcal{F}}[\boldsymbol{\zeta}(\mathbf{Q})] + \boldsymbol{\mathcal{F}}[\boldsymbol{\zeta}(-\mathbf{Q})] = I_2. \tag{C.58}$$

The transmission probability of photons in a pure state with Poincaré vector $\mathbf{P}$ ($P = 1$), and density matrix $\rho(\mathbf{P})$, through a polariser $\boldsymbol{\mathcal{F}}[\boldsymbol{\zeta}(\mathbf{Q}); W_+, W_-]$ is given by the trace of the matrix $\boldsymbol{\mathcal{F}}[\boldsymbol{\zeta}(\mathbf{Q}); W_+, W_-]\,\rho(\mathbf{P})$ (Sakurai, 1997; Ballentine, 1998),

$$\begin{aligned}\langle\boldsymbol{\mathcal{F}}[\boldsymbol{\zeta}(\mathbf{Q}); W_+, W_-]\rangle &= \mathrm{Tr}\{\boldsymbol{\mathcal{F}}[\boldsymbol{\zeta}(\mathbf{Q}); W_+, W_-]\,\rho(\mathbf{P})\} \\ &= \mathrm{Tr}\left\{\frac{1}{2}\left[(W_+ + W_-)I_2 + (W_+ - W_-)\mathbf{Q}\cdot\boldsymbol{\sigma}\right]\ \frac{1}{2}\left[I_2 + \mathbf{P}\cdot\boldsymbol{\sigma}\right]\right\}.\end{aligned}$$

Using the property (C.54) of the Pauli matrices, and the fact that these matrices are traceless, we obtain

$$\langle\boldsymbol{\mathcal{F}}[\boldsymbol{\zeta}(\mathbf{Q}); W_+, W_-]\rangle = \frac{1}{2}\left[(W_+ + W_-) + (W_+ - W_-)\mathbf{Q}\cdot\mathbf{P}\right]. \tag{C.59}$$

It is clear that, if $\mathbf{P}$ is parallel (antiparallel) to $\mathbf{Q}$, the transmission probability is W_+ (W_-). When $\mathbf{P}$ and $\mathbf{Q}$ are mutually orthogonal, the fraction of incident photons that is transmitted is $(W_+ + W_-)/2$. Using the relation (C.54), we see that

$$\boldsymbol{\mathcal{F}}^2[\boldsymbol{\zeta}(\mathbf{Q}); W_+, W_-] = \frac{1}{2}\left[(W_+^2 + W_-^2)I_2 + (W_+^2 - W_-^2)\mathbf{Q}\cdot\boldsymbol{\sigma}\right]. \tag{C.60}$$

This operator represents a composite filter consisting of two identical polarisers of the type $\boldsymbol{\mathcal{F}}[\boldsymbol{\zeta}(\mathbf{Q}); W_+, W_-]$. It has the same form as the operator $\boldsymbol{\mathcal{F}}[\boldsymbol{\zeta}(\mathbf{Q}); W_+, W_-]$, but with efficiencies W_+^2 and W_-^2, as could be expected from the fact that the attenuation of the composite filter is the product of attenuations of the individual filters.

C.4 Simulation and polarisation states

In the simulations, polarisation states are described by means of the Stokes parameters. Pure states with polarisation $\mathbf{P}$ are represented by the four-component vector of Stokes

parameters,

$$\mathsf{P} \equiv (1, \mathbf{P})^{\mathrm{T}} = \begin{pmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{pmatrix}, \qquad \text{with } P_0 = 1. \tag{C.61}$$

Alternatively, we can characterise the polarisation by means of the density-matrix four-vector

$$\mathsf{D}_{\mathbf{P}} \equiv \begin{pmatrix} \rho_{11} \\ \rho_{22} \\ \rho_{12} \\ \rho_{21} \end{pmatrix}, \tag{C.62}$$

whose components are the elements of the density matrix (notice the order). These two representations are related by the linear transformation

$$\mathsf{D}_{\mathbf{P}} = \mathcal{X}\mathsf{P}, \qquad \mathsf{P} = \mathcal{X}^{-1}\mathsf{D}_{\mathbf{P}}, \tag{C.63}$$

with

$$\mathcal{X} = \frac{1}{2}\begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & -\mathrm{i} & 0 \\ 0 & 1 & \mathrm{i} & 0 \end{pmatrix}, \qquad \mathcal{X}^{-1} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & \mathrm{i} & -\mathrm{i} \\ 1 & -1 & 0 & 0 \end{pmatrix}. \tag{C.64}$$

For brevity, the four-vector P will be referred to as the *Stokes vector*. Similarly, an ideal filter $\boldsymbol{\mathcal{F}}[\boldsymbol{\zeta}(\mathbf{Q})]$ that accepts photons with polarisation $\mathbf{Q}$ can be represented by the Stokes vector $\mathsf{Q} = (1, \mathbf{Q})^{\mathrm{T}}$. Photons with polarisation P are transmitted through the filter with probability [see Eq. (C.55)]

$$\langle \boldsymbol{\mathcal{F}}[\boldsymbol{\zeta}(\mathbf{Q})] \rangle = \frac{1}{2}(1 + \mathbf{Q}\cdot\mathbf{P}) \equiv \frac{1}{2}\,\mathsf{Q}^{\mathrm{T}}\mathsf{P}. \tag{C.65}$$

Or, in terms of the density-matrix four-vectors,

$$\langle \boldsymbol{\mathcal{F}}[\boldsymbol{\zeta}(\mathbf{Q})] \rangle = \mathrm{Tr}[\rho(\mathbf{Q})\rho(\mathbf{P})] = \mathsf{D}_{\mathbf{Q}}^{\mathrm{T}}\mathcal{Z}_{34}\mathsf{D}_{\mathbf{P}}, \tag{C.66}$$

where the matrix

$$\mathcal{Z}_{34} \equiv \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{C.67}$$

exchanges the third and fourth components of the vector $\mathsf{D}_{\mathbf{P}}$, as required to get the correct value of the trace.

A rotation of the zero-azimuth plane by an angle ϕ causes the rotation of the Stokes vector given by Eq. (C.43), *i.e.*,

$$\mathsf{P}' = \mathcal{R}(\phi)\mathsf{P} \qquad \text{with} \qquad \mathcal{R}(\phi) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(2\phi) & 0 & -\sin(2\phi) \\ 0 & 0 & 1 & 0 \\ 0 & \sin(2\phi) & 0 & \cos(2\phi) \end{pmatrix}. \tag{C.68}$$

Note that the Stokes vector of a filter transforms in the same way, *i.e.*, $\mathsf{Q}' = \mathcal{R}(\phi)\mathsf{Q}$. Indeed, from Eq. (C.65) we see that, when we rotate both the beam and the filter, the results of measurements remain unaltered,

$$\mathsf{Q}'^{\mathrm{T}}\mathsf{P}' = \mathsf{Q}^{\mathrm{T}}\mathcal{R}^{\mathrm{T}}(\phi)\mathcal{R}(\phi)\mathsf{P} = \mathsf{Q}^{\mathrm{T}}\mathsf{P}.$$

A beam of photons with polarisation $\mathbf{P}$ and intensity N (number of photons per unit volume in the beam) will be represented by the Stokes vector $N\mathsf{P}$. Similarly, the density matrix of the beam will be defined as $N\rho(\mathbf{P})$, so that the relations (C.63) are still valid. Note, however, that with this convention the trace of the density matrix, and the zeroth component of the Stokes vector, are equal to the intensity N of the beam. If the beam $N\mathsf{P}$ is passed through an ideal filter $\mathcal{F}[\boldsymbol{\zeta}(\mathbf{Q})]$, the transmitted beam has the intensity

$$N' = N\,\frac{1}{2}(1 + \mathbf{Q}\cdot\mathbf{P}) \equiv N\,\frac{1}{2}\,\mathsf{Q}^{\mathrm{T}}\mathsf{P}, \tag{C.69}$$

and its Stokes vector is $N'\mathsf{Q}$. The Stokes vector of the transparent filter ($\mathcal{F} = I_2$) is

$$\mathsf{Q}_I = \mathcal{X}^{-1}\begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{C.70}$$

Evidently, for the transparent filter, Eq. (C.69) gives $N' = N$.

Appendix D

Electron/positron transport in electromagnetic fields

In this Appendix, we consider the transport of electrons (and 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 $\boldsymbol{\mathcal{E}}$ and a magnetic field $\boldsymbol{\mathcal{B}}$, which are set up by external sources and do not vary with time. For practical purposes, we also consider that both $\boldsymbol{\mathcal{E}}$ and $\boldsymbol{\mathcal{B}}$ 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 (hard interaction or hinge), the electron loses a discrete amount of kinetic energy and changes its direction of motion. In the absence of EM fields, the electron travels between consecutive interaction subject only to the stopping power due to soft energy-loss interactions, *i.e.*, following a straight trajectory segment and slowing down with constant stopping power $S_{\rm soft}$ (see Section 4.3.3). 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 segment the electron is driven by the EM force.

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 does not vary much along a segment. In practice, however, we can always limit the maximum segment length by means of the parameter $s_{\rm max}$ so that the energy variation caused by the electric field in each trajectory segment is still small. The effect of the EM field can then 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 the electrons leave the field volume). Injection of a single electron in the medium may

give rise to a complex cascade of delta rays, which 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 (Jackson, 1975); this approximation is not valid for very strong magnetic and electric fields.

The simulation algorithm described below is designed to track charged particles in the presence of EM fields that are constant with time *and* vary slowly with position. More specifically, we assume that the particle trajectory can be described as a succession of short segments such that the electric and magnetic fields are essentially constant along each segment. When this assumption is not valid (*i.e.*, when the fields vary too rapidly with position or time), our simple scheme for solving the equation of motion [Eq. (D.8)] is not applicable.

D.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. The force acting on an electron ($Z_0 = -1$) or positron ($Z_0 = +1$) is the Lorentz force [1],

$$\mathbf{F} = Z_0 e \left(\boldsymbol{\mathcal{E}} + \frac{1}{c}\, \mathbf{v} \times \boldsymbol{\mathcal{B}} \right) = Z_0 e \left(\boldsymbol{\mathcal{E}} + \beta\, \hat{\mathbf{v}} \times \boldsymbol{\mathcal{B}} \right) \tag{D.1}$$

with $\hat{\mathbf{v}} = \mathbf{v}/v$ and $\beta = v/c$. The momentum and the kinetic energy of the particle are, respectively, $\mathbf{p} = \mathrm{m_e} c\, \gamma\beta \hat{\mathbf{v}}$ and $E = (\gamma - 1)\mathrm{m_e}c^2$, were $\gamma = (1-\beta^2)^{-1/2}$. We notice that

$$\frac{\mathrm{d}\gamma}{\mathrm{d}t} = \gamma^3 \beta\, \frac{\mathrm{d}\beta}{\mathrm{d}t} \qquad \text{and} \qquad \gamma^2\beta^2 + 1 = \gamma^2. \tag{D.2}$$

The relativistic equation of motion,

$$\frac{\mathrm{d}\mathbf{p}}{\mathrm{d}t} = \mathbf{F} \tag{D.3}$$

can be reduced to a more tractable form by writing it as

$$\frac{\mathrm{d}(\gamma\beta\hat{\mathbf{v}})}{\mathrm{d}t} = \frac{\mathbf{F}}{\mathrm{m_e}c}, \tag{D.4}$$

and considering the equality, which follows directly from Eqs. (D.2),

$$\frac{\mathrm{d}(\gamma\beta\hat{\mathbf{v}})}{\mathrm{d}t} = \gamma^3 \frac{\mathrm{d}\beta}{\mathrm{d}t}\hat{\mathbf{v}} + \gamma\beta \frac{\mathrm{d}\hat{\mathbf{v}}}{\mathrm{d}t}\,, \tag{D.5}$$

[1] In this Appendix, electromagnetic quantities are expressed in the Gaussian system of units.

where the vectors $\hat{\mathbf{v}}$ and $\mathrm{d}\hat{\mathbf{v}}/\mathrm{d}t$ are orthogonal. Then, projecting Eq. (D.4) into the directions of these two vectors, we conclude that

$$\frac{\mathrm{d}\beta}{\mathrm{d}t} = \frac{1}{\gamma^3}\,\frac{\mathbf{F}\cdot\hat{\mathbf{v}}}{\mathrm{m_e}c} \qquad \text{and} \qquad \frac{\mathrm{d}\hat{\mathbf{v}}}{\mathrm{d}t} = \frac{1}{\gamma\beta}\,\frac{\mathbf{F}-(\mathbf{F}\cdot\hat{\mathbf{v}})\hat{\mathbf{v}}}{\mathrm{m_e}c}. \tag{D.6}$$

Therefore,

$$\begin{aligned}\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = \frac{\mathrm{d}(c\beta\hat{\mathbf{v}})}{\mathrm{d}t} &= c\,\frac{\mathrm{d}\beta}{\mathrm{d}t}\,\hat{\mathbf{v}} + c\beta\,\frac{\mathrm{d}\hat{\mathbf{v}}}{\mathrm{d}t} \\ &= \frac{1}{\gamma^3}\,\frac{\mathbf{F}\cdot\hat{\mathbf{v}}}{\mathrm{m_e}}\,\hat{\mathbf{v}} + \frac{1}{\gamma}\,\frac{\mathbf{F}-(\mathbf{F}\cdot\hat{\mathbf{v}})\hat{\mathbf{v}}}{\mathrm{m_e}},\end{aligned} \tag{D.7}$$

which we cast in the familiar form

$$\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = \mathbf{A}, \qquad \mathbf{A} \equiv \frac{1}{\mathrm{m_e}\gamma}\left\{(1-\beta^2)\,(\mathbf{F}\cdot\hat{\mathbf{v}})\,\hat{\mathbf{v}} + \left[\mathbf{F}-(\mathbf{F}\cdot\hat{\mathbf{v}})\hat{\mathbf{v}}\right]\right\}. \tag{D.8}$$

In the case of the Lorentz force, Eq. (D.1), the acceleration is

$$\mathbf{A} = \frac{Z_0 e}{\mathrm{m_e}\gamma}\left[\boldsymbol{\mathcal{E}} - \beta^2(\boldsymbol{\mathcal{E}}\cdot\hat{\mathbf{v}})\hat{\mathbf{v}} + \beta\hat{\mathbf{v}}\times\boldsymbol{\mathcal{B}}\right]. \tag{D.9}$$

Notice that, for arbitrary fields $\boldsymbol{\mathcal{E}}$ and $\boldsymbol{\mathcal{B}}$, $\mathbf{A}$ is a function of the particle's position $\mathbf{r}$, energy E, and direction of motion $\hat{\mathbf{v}}$.

The equation of motion (D.8) can be solved numerically, *e.g.*, by using a Runge-Kutta algorithm (DeVries, 1994), but this procedure would be too lengthy for straight implementation in a simulation code. 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 segments such that *the acceleration* $\mathbf{A}$ *does not change much over the course of a segment.* Along each segment, we then have

$$\mathbf{v}(t) = \mathbf{v}_0 + t\,\mathbf{A}(\mathbf{r}_0, E_0, \hat{\mathbf{v}}_0) \tag{D.10}$$

$$\mathbf{r}(t) = \mathbf{r}_0 + t\,\mathbf{v}_0 + \frac{1}{2}\,t^2\mathbf{A}(\mathbf{r}_0, E_0, \hat{\mathbf{v}}_0), \tag{D.11}$$

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

$$t = \int_0^s \frac{\mathrm{d}s'}{v}, \tag{D.12}$$

which to first order becomes

$$t = \frac{s}{v_0} = \frac{s}{c\beta_0}. \tag{D.13}$$

Then, to first order in the force,

$$\mathbf{v}(s) = \mathbf{v}_0 + s\,c\beta_0\,\frac{\mathbf{A}(\mathbf{r}_0, E_0, \hat{\mathbf{v}}_0)}{c^2\beta_0^2} \tag{D.14}$$

and

$$\mathbf{r}(s) = \mathbf{r}_0 + s\,\hat{\mathbf{v}}_0 + \frac{1}{2}\,s^2\frac{\mathbf{A}(\mathbf{r}_0, E_0, \hat{\mathbf{v}}_0)}{c^2\beta_0^2}. \tag{D.15}$$

In the case of the Lorentz force, these formulas give

$$\mathbf{r}(s) = \mathbf{r}_0 + s\,\hat{\mathbf{v}}_0 + \frac{1}{2}\,s^2\,\frac{Z_0 e\,[\boldsymbol{\mathcal{E}}_0 - \beta_0^2(\boldsymbol{\mathcal{E}}_0\cdot\hat{\mathbf{v}}_0)\hat{\mathbf{v}}_0 + \beta_0\hat{\mathbf{v}}_0\times\boldsymbol{\mathcal{B}}_0]}{\mathrm{m_e}c^2\,\gamma_0\beta_0^2} \tag{D.16}$$

and

$$\mathbf{v}(s) = \mathbf{v}_0 + \Delta\mathbf{v} \tag{D.17}$$

with

$$\Delta\mathbf{v} = s\,c\beta_0\,\frac{Z_0 e\,[\boldsymbol{\mathcal{E}}_0 - \beta_0^2(\boldsymbol{\mathcal{E}}_0\cdot\hat{\mathbf{v}}_0)\hat{\mathbf{v}}_0 + \beta_0\hat{\mathbf{v}}_0\times\boldsymbol{\mathcal{B}}_0]}{\mathrm{m_e}c^2\,\gamma_0\beta_0^2}. \tag{D.18}$$

In our tracking algorithm, the velocity is used to determine the direction vector at the end of the segment,

$$\hat{\mathbf{v}}(s) = \frac{\mathbf{v}_0 + \Delta\mathbf{v}}{|\mathbf{v}_0 + \Delta\mathbf{v}|}. \tag{D.19}$$

Owing to the action of the EM force, the kinetic energy E of the particle varies along the segment. 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 $E(s)$ as

$$E(s) = E_0 + Z_0 e\,[\varphi(\mathbf{r}_0) - \varphi(\mathbf{r}(s))] \tag{D.20}$$

where $\varphi(\mathbf{r})$ is the scalar 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 segment

$$\frac{|\boldsymbol{\mathcal{E}}(\mathbf{r}(s)) - \boldsymbol{\mathcal{E}}(\mathbf{r}_0)|}{|\boldsymbol{\mathcal{E}}(\mathbf{r}_0)|} < \delta_{\boldsymbol{\mathcal{E}}} \ll 1, \qquad \frac{|\boldsymbol{\mathcal{B}}(\mathbf{r}(s)) - \boldsymbol{\mathcal{B}}(\mathbf{r}_0)|}{|\boldsymbol{\mathcal{B}}(\mathbf{r}_0)|} < \delta_{\boldsymbol{\mathcal{B}}} \ll 1 \tag{D.21}$$

and
2) the relative changes in kinetic energy and velocity (or direction of motion) are small

$$\left|\frac{E(s) - E_0}{E_0}\right| < \delta_E \ll 1, \qquad \frac{|\Delta\mathbf{v}|}{v_0} < \delta_{\mathrm{v}} \ll 1. \tag{D.22}$$

These conditions set an upper limit on the allowed segment 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 segment 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 segment 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.

D.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 Bielajew, 1988)

$$\mathbf{r}(t) = \mathbf{r}_0 + t\mathbf{v}_{0\perp} + \frac{1}{a}\left[\cosh(act) - 1 + \frac{v_{0\parallel}}{c}\sinh(act)\right]\hat{\boldsymbol{\mathcal{E}}}, \tag{D.23}$$

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,

$$\mathbf{v}_{0\parallel} = (\mathbf{v}_0\cdot\hat{\boldsymbol{\mathcal{E}}})\hat{\boldsymbol{\mathcal{E}}}, \qquad \mathbf{v}_{0\perp} = \mathbf{v}_0 - (\mathbf{v}_0\cdot\hat{\boldsymbol{\mathcal{E}}})\hat{\boldsymbol{\mathcal{E}}} \tag{D.24}$$

and

$$a \equiv \frac{Z_0 e\mathcal{E}}{\mathrm{m_e}c^2\gamma_0} = \frac{Z_0 e\mathcal{E}}{E_0 + \mathrm{m_e}c^2}. \tag{D.25}$$

The velocity of the particle is

$$\begin{aligned}\mathbf{v}(t) &= \mathbf{v}_{0\perp} + \left[c\sinh(act) + v_{0\parallel}\cosh(act)\right]\hat{\boldsymbol{\mathcal{E}}} \\ &= \mathbf{v}_0 + \left\{c\sinh(act) + v_{0\parallel}\left[\cosh(act) - 1\right]\right\}\hat{\boldsymbol{\mathcal{E}}}.\end{aligned} \tag{D.26}$$

Since the scalar potential for the constant field is $\varphi(\mathbf{r}) = -\boldsymbol{\mathcal{E}}\cdot\mathbf{r}$, the kinetic energy of the particle varies with time and is given by

$$E(t) = E_0 - Z_0 e\boldsymbol{\mathcal{E}}\cdot\left[\mathbf{r}_0 - \mathbf{r}(t)\right]. \tag{D.27}$$

Figure D.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 all trajectories lie on the xz-plane. Continuous curves represent exact trajectories obtained from the analytical formula (D.23). The dashed curves are the results from the first-order tracking algorithm described above [Eqs. (D.16)–(D.22)] with $\delta_{\boldsymbol{\mathcal{E}}} = \delta_E = \delta_{\mathrm{v}} = 0.02$. 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 segments, *i.e.*, smaller δ-values.

D.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 $\boldsymbol{\mathcal{B}}$. 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

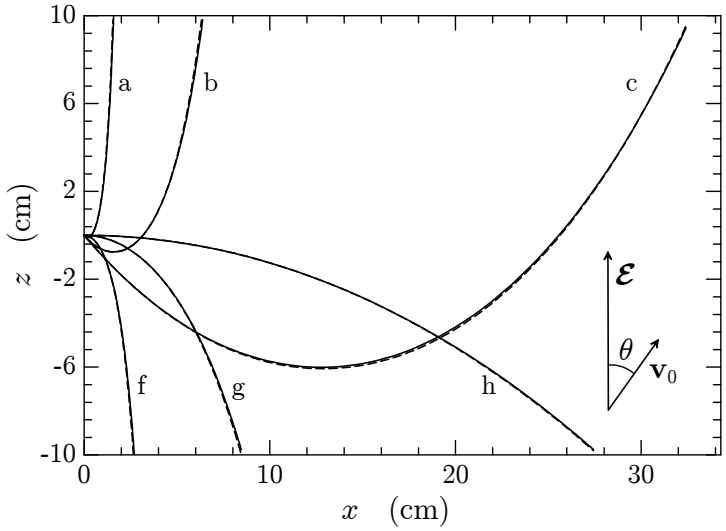

Figure D.1: Trajectories of electrons and positrons in a uniform electric field of 511 kV/cm. Continuous curves represent exact trajectories obtained from Eq. (D.23). The dashed lines are obtained by using the first-order numerical tracking method described by Eqs. (D.16)–(D.22) with $\delta_{\boldsymbol{\mathcal{E}}} = \delta_E = \delta_{\rm v} = 0.02$. 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.

a constant of motion. It is convenient to introduce the precession frequency vector $\boldsymbol{\omega}$, defined by (notice the sign)

$$\boldsymbol{\omega} \equiv -\frac{Z_0 e \boldsymbol{\mathcal{B}}}{\mathrm{m_e}\gamma c} = -\frac{Z_0 e c \boldsymbol{\mathcal{B}}}{E_0 + \mathrm{m_e}c^2}, \tag{D.28}$$

and split the velocity $\mathbf{v}$ into its components parallel and perpendicular to $\boldsymbol{\omega}$,

$$\mathbf{v}_{\parallel} = (\mathbf{v}\cdot\hat{\boldsymbol{\omega}})\hat{\boldsymbol{\omega}}, \qquad \mathbf{v}_{\perp} = \mathbf{v} - (\mathbf{v}\cdot\hat{\boldsymbol{\omega}})\hat{\boldsymbol{\omega}}. \tag{D.29}$$

Then, the equation of motion (D.8) becomes

$$\frac{\mathrm{d}\mathbf{v}_{\parallel}}{\mathrm{d}t} = 0, \qquad \frac{\mathrm{d}\mathbf{v}_{\perp}}{\mathrm{d}t} = \boldsymbol{\omega}\times\mathbf{v}_{\perp}. \tag{D.30}$$

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 $\boldsymbol{\mathcal{B}}$, the particle describes a circle with angular frequency ω and speed $v_{0\perp}$ (which is a constant of motion). The radius of the circle is $R = v_{0\perp}/\omega$. That is, the trajectory is an helix with central axis along the $\boldsymbol{\mathcal{B}}$ 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. D.2).

In terms of the path length $s = tv_0$, the equation of motion takes the form

$$\mathbf{r}(s) = \mathbf{r}_0 + \frac{s}{v_0}\mathbf{v}_{0\parallel} + R\left[1 - \cos(s_\perp/R)\right](\hat{\boldsymbol{\omega}}\times\hat{\mathbf{v}}_{0\perp}) + R\,\sin(s_\perp/R)\hat{\mathbf{v}}_{0\perp}, \tag{D.31}$$

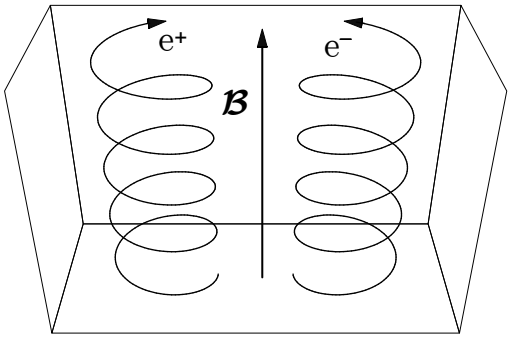

Figure D.2: Trajectories of electrons and positrons in a uniform magnetic field. The two particles start from the base plane with equal initial velocities.

where $\hat{\mathbf{v}}_{0\perp} \equiv \mathbf{v}_{0\perp}/v_{0\perp}$ and $s_\perp = sv_{0\perp}/v_0$. Equivalently,

$$\mathbf{r}(s) = \mathbf{r}_0 + s\hat{\mathbf{v}}_0 - \frac{s}{v_0}\mathbf{v}_{0\perp} + \frac{1}{\omega}\left[1-\cos(s\omega/v_0)\right](\hat{\boldsymbol{\omega}}\times\mathbf{v}_{0\perp}) + \frac{1}{\omega}\sin(s\omega/v_0)\mathbf{v}_{0\perp}. \quad \text{(D.32)}$$

After the path length s, the particle velocity is

$$\mathbf{v}(s) = v_0\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}s} = \mathbf{v}_0 + \left[\cos(s\omega/v_0)-1\right]\mathbf{v}_{0\perp} + \sin(s\omega/v_0)(\hat{\boldsymbol{\omega}}\times\mathbf{v}_{0\perp}). \quad \text{(D.33)}$$

In Fig. D.3 we compare exact trajectories of electrons and positrons in a uniform magnetic field obtained from the analytical formula (D.32) with results from the first-order tracking algorithm [Eqs. (D.16)-(D.22)] with $\delta_{\boldsymbol{\mathcal{B}}} = \delta_E = \delta_{\mathrm{v}} = 0.02$. 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 segment. In practice, however, this is not convenient because it implies a considerable increase in numerical work. The formulas (D.32) and (D.33) provide a more convenient tracking algorithm, because their accuracy is not limited by the segment length as long as the magnetic field stays essentially constant along the segment.

D.2 Tracking particles in matter with an EM field

In class II (mixed) simulations, along each trajectory segment a particle slows down because of soft energy-loss interactions. PENELOPE simulates the slowing down due to soft interactions by means of an effective stopping power S_{soft} (see Section 4.3.3), which is constant along the trajectory segment. In the absence of EM fields, we have

$$S_{\mathrm{soft}} = -\frac{\mathrm{d}E}{\mathrm{d}s} = -\frac{1}{v}\frac{\mathrm{d}E}{\mathrm{d}t} = -\frac{1}{v}\frac{\mathrm{d}(\mathrm{m_e}c^2\gamma)}{\mathrm{d}t} = -\mathrm{m_e}c\gamma^3\frac{\mathrm{d}\beta}{\mathrm{d}t}, \quad \text{(D.34)}$$

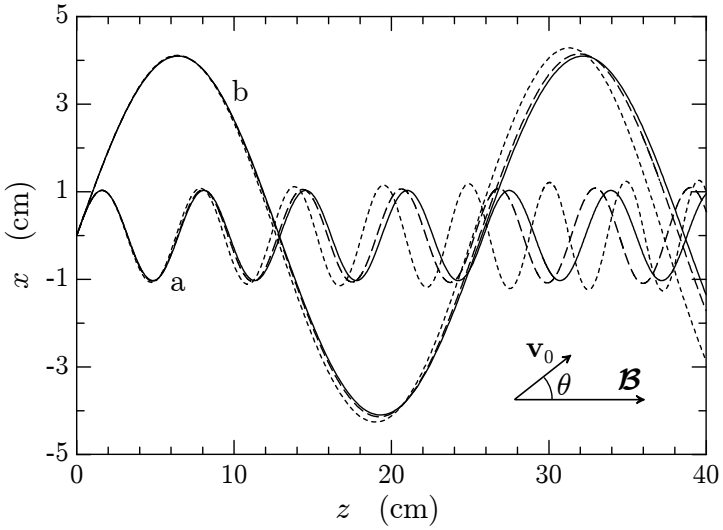

Figure D.3: Trajectories of electrons and positrons in a uniform magnetic field of 0.2 tesla. Continuous curves are exact trajectories calculated from Eq. (D.32). The short-dashed lines are obtained by using the numerical tracking method described in the text with $\delta_{\rm v} = 0.02$. Long-dashed curves are the results from the tracking algorithm with $\delta_{\rm v} = 0.005$. a: electrons, $E_0 = 0.5$ MeV, $\theta = 45$ deg. b: positrons, $E_0 = 3$ MeV, $\theta = 45$ deg.

where we have used the first of the equalities (D.2). Hence

$$\frac{\mathrm{d}\beta}{\mathrm{d}t} = -\frac{S_{\rm soft}}{\gamma^3 \mathrm{m_e} c}. \tag{D.35}$$

This equality shows that the stopping power acts as a retarding force [cf. Eqs. (D.6) and (D.7)]

$$F_{\rm soft} = -\, S_{\rm soft}\, \hat{\mathbf{v}}, \tag{D.36}$$

which should be added to the Lorentz force (D.1). That is, under the action of an EM field and soft energy-loss interactions, the acceleration of the transported particle along a trajectory segment is

$$\mathbf{A} = \frac{Z_0 e}{\mathrm{m_e}\gamma}\left[\boldsymbol{\mathcal{E}} - \beta^2(\boldsymbol{\mathcal{E}}\cdot\hat{\mathbf{v}})\hat{\mathbf{v}} + \beta\hat{\mathbf{v}}\times\boldsymbol{\mathcal{B}}\right] - \frac{1}{\mathrm{m_e}\gamma}\,(1-\beta^2)\,S_{\rm soft}\,\hat{\mathbf{v}}. \tag{D.37}$$

With our first-order tracking algorithm [see Eqs. (D.16) and (D.18)], the effects of the electric field, the magnetic field, and the stopping force are uncoupled, *i.e.*, they can be evaluated separately. Because the stopping force is properly accounted for by PENELOPE, we only need to consider the alterations of trajectories caused by the EM field.

For uniform electric fields, the algorithm offers a satisfactory solution since it usually admits relatively large segment lengths. In the case of uniform magnetic fields (with $\boldsymbol{\mathcal{E}} = 0$), the kinetic energy is a constant of the motion and the only effective constraint

on the segment 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 $\boldsymbol{\mathcal{B}}$ are circles and the first-order algorithm generates each segment as a parabolic segment, we need to move in sub-segments of length much less than the radius R (*i.e.*, $\delta_{\rm v}$ 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. (D.32) and (D.33)], 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 EM field is essentially constant along each trajectory segment and write

$$\mathbf{r}(s) = \mathbf{r}_0 + s\hat{\mathbf{v}}_0 + (\Delta\mathbf{r})_{\boldsymbol{\mathcal{E}}} + (\Delta\mathbf{r})_{\boldsymbol{\mathcal{B}}}, \tag{D.38}$$

where $(\Delta\mathbf{r})_{\boldsymbol{\mathcal{E}}}$ and $(\Delta\mathbf{r})_{\boldsymbol{\mathcal{B}}}$ are the displacements caused by the electric field and the magnetic field, respectively. For $(\Delta\mathbf{r})_{\boldsymbol{\mathcal{E}}}$ we use the first-order approximation [see Eq. (D.16)],

$$(\Delta\mathbf{r})_{\boldsymbol{\mathcal{E}}} = \frac{1}{2}\, s^2\, \frac{Z_0 e\,[\boldsymbol{\mathcal{E}}_0 - \beta_0^2(\boldsymbol{\mathcal{E}}_0\cdot\hat{\mathbf{v}}_0)\hat{\mathbf{v}}_0]}{\mathrm{m_e}c^2\,\gamma_0\beta_0^2}. \tag{D.39}$$

The displacement caused by the magnetic field is evaluated using the result (D.32), *i.e.*,

$$(\Delta\mathbf{r})_{\boldsymbol{\mathcal{B}}} = -\frac{s}{v_0}\mathbf{v}_{0\perp} + \frac{1}{\omega}\,[1-\cos(s\omega/v_0)]\,(\hat{\boldsymbol{\omega}}\times\mathbf{v}_{0\perp}) + \frac{1}{\omega}\sin(s\omega/v_0)\mathbf{v}_{0\perp} \tag{D.40}$$

with

$$\boldsymbol{\omega} \equiv -\frac{Z_0 ec\boldsymbol{\mathcal{B}}_0}{E_0 + \mathrm{m_e}c^2}, \qquad \text{and} \qquad \mathbf{v}_{0\perp} = \mathbf{v}_0 - (\mathbf{v}_0\cdot\hat{\boldsymbol{\omega}})\hat{\boldsymbol{\omega}}. \tag{D.41}$$

Similarly, the particle velocity along the segment is expressed as

$$\mathbf{v}(s) = \mathbf{v}_0 + (\Delta\mathbf{v})_{\boldsymbol{\mathcal{E}}} + (\Delta\mathbf{v})_{\boldsymbol{\mathcal{B}}} \tag{D.42}$$

with [see Eqs. (D.18) and (D.33)]

$$(\Delta\mathbf{v})_{\boldsymbol{\mathcal{E}}} = s\,c\beta_0\,\frac{Z_0 e\,[\boldsymbol{\mathcal{E}}_0 - \beta_0^2(\boldsymbol{\mathcal{E}}_0\cdot\hat{\mathbf{v}}_0)\hat{\mathbf{v}}_0]}{\mathrm{m_e}c^2\,\gamma_0\beta_0^2} \tag{D.43}$$

and

$$(\Delta\mathbf{v})_{\boldsymbol{\mathcal{B}}} = [\cos(s\omega/v_0) - 1]\,\mathbf{v}_{0\perp} + \sin(s\omega/v_0)(\hat{\boldsymbol{\omega}}\times\mathbf{v}_{0\perp}). \tag{D.44}$$

The energy $E(s)$ at the end of the segment is [cf. Eq. (D.20)]

$$E(s) = E_0 + Z_0 e\,[\varphi(\mathbf{r}_0) - \varphi(\mathbf{r}(s))]\,. \tag{D.45}$$

In our implementation of this tracking algorithm, the allowed segment lengths s are limited by the following constraints [see Eqs. (D.21) and (D.22)]

$$\frac{|\boldsymbol{\mathcal{E}}(\mathbf{r}(s)) - \boldsymbol{\mathcal{E}}(\mathbf{r}_0)|}{|\boldsymbol{\mathcal{E}}(\mathbf{r}_0)|} < \delta_{\boldsymbol{\mathcal{E}}} \ll 1, \qquad \frac{|\boldsymbol{\mathcal{B}}(\mathbf{r}(s)) - \boldsymbol{\mathcal{B}}(\mathbf{r}_0)|}{|\boldsymbol{\mathcal{B}}(\mathbf{r}_0)|} < \delta_{\boldsymbol{\mathcal{B}}} \ll 1 \tag{D.46}$$

and

$$\left|\frac{E(s)-E_0}{E_0}\right| < \delta_E \ll 1, \qquad \frac{\left|(\Delta\mathbf{v})_{\boldsymbol{\mathcal{E}}}+(\Delta\mathbf{v})_{\boldsymbol{\mathcal{B}}}\right|}{v_0} < \delta_{\mathrm{v}} \ll 1. \tag{D.47}$$

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 $\delta_{\boldsymbol{\mathcal{E}}} = \delta_{\boldsymbol{\mathcal{B}}} = \delta_E = \delta_{\mathrm{v}} = 0.02$. Notice that, in the case of a uniform magnetic field, the tracking algorithm is now exact, irrespective of the segment length.

This tracking algorithm has been implemented in the subroutine package PENFIELD, which is devised to work linked to PENELOPE and PENGEOM. 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 (1972), *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.

Acosta, E., X. Llovet, and F. Salvat (2002), "Monte Carlo simulation of bremsstrahlung emission by electrons," *Appl. Phys. Lett.* **80**, 3228–3230.

Agostinelli, S., J. Allison, K. Amako, J. Apostolakis, H. Araujo, P. Arce, and M. Asai (2003), "Geant4—a simulation toolkit," *Nucl. Instrum. Meth. A* **506**, 250–303.

Allison, J., K. Amako, J. Apostolakis, H. Araujo, P. Arce Dubois, and M. Asai (2006), "Geant4 developments and applications," *IEEE Trans. Nucl. Sci.* **53**, 270–278.

Almansa, J., F. Salvat-Pujol, G. Díaz-Londoño, A. Carnicer, A. M. Lallena, and F. Salvat (2016), "PENGEOM – A general-purpose geometry package for Monte Carlo simulation of radiation transport in complex material structures," *Comput. Phys. Commun.* **199**, 102–113.

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," *Radiat. Res.* **100**, 16–29.

Badal, A. and J. Sempau (2006), "A package of Linux scripts for the parallelization of Monte Carlo simulations," *Comput. Phys. Commun.* **175**, 440–450.

Ballentine, L. E. (1998), *Quantum Mechanics. A Modern Development* (World Scientific Publ. Co., Singapore), 1st edition.

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.

Baym, G. (1974), *Lectures in Quantum Mechanics* (Westview Press, Boulder, Colorado).

Bé, M.-M., N. Coursol, B. Duchemin, F. Lagoutine, J. Legrand, K. Debertin, and E. Schönfeld (2011), *Table of Radionuclides. Introduction*, Technical Report LNHB 211/53, Laboratoire National Henri Becquerel, Saclay, France, evaluated decay data files can be downloaded from the NUCLEIDE web site, `www.nucleide.org/DDEP_WG/DDEPdata.htm`.

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 B. Alder, S. Fernbach, and M. Rotenberg (editors), "*Methods in Computational Physics*," volume 1, pages 135–215 (Academic Press, New York).

Berger, M. J. (1992), *ESTAR, PSTAR and ASTAR: computer programs for calculating stopping-power and range tables for electrons, protons and helium ions*, Technical Report NISTIR 4999, National Institute of Standards and Technology, Gaithersburg, MD, available from `www.nist.gov/pml/data/star/index.cfm`.

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*, Technical Report NBSIR 87-3597, National Bureau of Standards, Gaithersburg, MD.

Berger, M. J., J. H. Hubbell, S. M. Seltzer, J. Chang, J. S. Coursey, R. Sukumar, and D. S. Zucker (2005), *XCOM: Photon Cross Sections Database*, Technical report, Institute of Standards and Technology, Gaithersburg, MD, available from `http://physics.nist.gov/PhysRefData/Xcom/Text/XCOM.html`.

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*, Technical Report NBSIR 82-2550, National Bureau of Standards, Gaithersburg, MD.

Berger, M. J. and S. M. Seltzer (1988a), "An overview of ETRAN Monte Carlo methods," in T. M. Jenkins, W. R. Nelson, and A. Rindi (editors), "*Monte Carlo Transport of Electrons and Photons*," chapter 7 (Plenum, New York).

Berger, M. J. and S. M. Seltzer (1988b), "Applications of ETRAN Monte Cario codes," in T. M. Jenkins, W. R. Nelson, and A. Rindi (editors), "*Monte Carlo Transport of Electrons and Photons*," chapter 9 (Plenum, New York).

Berger, M. J. and S. M. Seltzer (1988c), "ETRAN — Experimental Benchmarks," in T. M. Jenkins, W. R. Nelson, and A. Rindi (editors), "*Monte Carlo Transport of Electrons and Photons*," chapter 8 (Plenum, New York).

Bethe, H. A. (1930), "Zur Theorie des Durchgangs schneller Korpuskularstrahlen durch Materie," *Ann. Physik* **397**, 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. A* **146**, 83–112.

Bethe, H. A. and E. E. Salpeter (1957), *Quantum Mechanics of One- and Two-Electron Atoms* (Springer-Verlag, Berlin).

Bhabha, H. J. (1936), "The scattering of positrons by electrons with exchange on Dirac's theory of electrons," *Proc. Phys. Soc. A* **154**, 195–196.

Bielajew, A. F. (1988), "Electron transport in $\vec{E}$ and $\vec{B}$ fields," in T. M. Jenkins, W. R. Nelson, and A. Rindi (editors), "*Monte Carlo Transport of Electrons and Photons*," chapter 19 (Plenum, New York).

Bielajew, A. F. (1995), `HOWFAR` *and* `HOWNEAR`*: Geometry modeling for Monte Carlo particle transport*, Technical 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 T. M. Jenkins, W. R. Nelson, and A. Rindi (editors), "*Monte Carlo Transport of Electrons and Photons*," chapter 18 (Plenum, New York).

Bielajew, A. F. and F. Salvat (2001), "Improved elctron 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).

Bote, D. and F. Salvat (2008), "Calculations of inner-shell ionization by electron impact with the distorted-wave and plane-wave Born approximations," *Phys. Rev. A* **77**, 042701.

Bote, D., F. Salvat, A. Jablonski, and C. J. Powell (2009), "Cross sections for ionization of K, L and M shells of atoms by impact of electrons and positrons with energies up to 1 GeV: Analytical formulas," *At. Data and Nucl. Data Tables* **96**, 871–909.

Bote, D., F. Salvat, A. Jablonski, and C. J. Powell (2011), "Erratum to Cross sections for ionization of K, L and M shells of atoms by impact of electrons and positrons with energies up to 1 GeV: Analytical formulas," *At. Data and Nucl. Data Tables* **97**, 186.

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. Zanarini (1987), *GEANT3*, Technical 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.

Büermann, L., B. Grosswendt, H.-M. Kramer, H.-J. Selbach, M. Gerlach, M. Hoffmann, and M. Krumrey (2006), "Measurement of the x-ray mass energy-absorption coefficient of air using 3 keV to 10 keV synchrotron radiation," *Phys. Med. Biol.* **51**, 167–175.

Bush, K., S. F. Zavgorodni, and W. A. Beckham (2007), "Azimuthal particle redistribution for the reduction of latent phase-space variance in Monte Carlo simulations," *Phys. Med. Biol.* **52**, 4345–4360.

Campbell, J. L. and T. Papp (2001), "Widths of the atomic K–N7 levels," *At. Data Nucl. Data Tables* **77**, 1–56.

Carlson, T. A. (1975), *Photoelectron and Auger Spectroscopy* (Plenum Press, New York).

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.

Chetty, I. J., B. Curran, J. E. Cygler, J. J. DeMarco, G. Ezzell, B. A. Faddegon, I. K. wra kow, P. J. Keall, H. Liu, C.-M. C. Ma, D. W. O. Rogers, J. Seuntjens, D. Sheikh-Bagheri, and J. V. Siebers (2007), "Report of the AAPM Task Group No. 105: Issues associated with clinical implementation of Monte Carlo-based photon and electron external beam treatment planning," *Med. Phys.* **34**, 4818–4853.

Coleman, W. A. (1968), "Mathematical verification of a certain Monte Carlo sampling technique and applications of the technique to radiation transport problems," *Nucl. Sci. and Engineering* **32**, 76–81.

Condon, E. U. and H. Odabaşi (1980), *Atomic Structure* (Cambridge University Press, Cambridge, UK).

Cooper, M. J. (1985), "Compton scattering and electron momentum determination," *Rep. Prog. Phys.* **48**, 415–481.

Cristy, M. and K. F. Eckerman (1987), *Specific absorbed fractions of energy at various ages from internal photon sources I. Methods*, Technical Report ORNL/TM 8381/Vi, Oak Ridge National Laboratory, Oak Ridge, TN.

Cromer, D. T. and D. Liberman (1970), "Relativistic calculation of anomalous scattering factors for x rays," *J. Chem. Phys.* **53**, 1891–1898.

Cullen, D. E., J. H. Hubbell, and L. Kissel (1997), *EPDL97 The Evaluated Photon Data Library, '97 Version*, Technical Report UCRL-50400, Lawrence Livermore National Laboratory, Livermore, California.

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.

Desclaux, J. P. (1977), "Erratum notice," *Comput. Phys. Commun.* **13**, 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.

DeVries, P. L. (1994), *A First Course in Computational Physics* (John Wiley and Sons, Inc., New York).

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* (Princeton University Press, Princeton, NJ).

Falkoff, D. L. and J. E. MacDonald (1951), "On the Stokes parameters for polarized radiation," *J. Opt. Soc. Am.* **41**, 861–862.

Fano, U. (1954a), "A Stokes-parameter technique for the treatment of polarization in quantum mechanics," *Phys. Rev.* **93**, 121–123.

Fano, U. (1954b), "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.

Fano, U., L. V. Spencer, and M. J. Berger (1959), "Penetration and diffusion of x rays," in S. Flügge (editor), "Encyclopedia of Physics," volume 38/2, pages 660–817 (Springer Verlag, Berlin).

Fernández, J. E., M. Bastiano, and A. Tartari (1998), "The ionization loss of energy in gases and in condensed materials," *X-Ray Spectrom.* **27**, 325–331.

Fernández-Varea, J. M., R. Mayol, J. Baró, and F. Salvat (1993a), "On the theory and simulation of multiple elastic scattering of electrons," *Nucl. Instrum. Meth. B* **73**, 447–473.

Fernández-Varea, J. M., R. Mayol, D. Liljequist, and F. Salvat (1993b), "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., F. Salvat, M. Dingfelder, and D. Liljequist (2005), "A relativistic optical-data model for inelastic scattering of electrons and positrons in condensed matter," *Nucl. Instrum. Meth. B* **229**, 187–218.

Ferrari, A., P. R. Sala, A. Fassò, and J. Ranft (2005), *Fluka: a multi-particle transport code*, Technical Report CERN200500X, INFN TC 05/11, SLACR773, CERN, Geneva.

Furness, J. B. and I. E. McCarthy (1973), "Semiphenomenological optical model for electron scattering on atoms," *J. Phys. B: Atom. Mol. Phys.* **6**, 2280–2291.

García-Toraño, E., V. Peyres, M.-M. Bé, C. Dulieu, M.-C. Lépy, and F. Salvat (2017), "Simulation of decay processes and radiation transport times in radioactivity measurements," *Nucl. Instrum. Meth. B* **396**, 43–49.

Glassner, A. S. (1984), "Space subdivision for fast ray tracing," *IEEE Computer Graphics and Applications* **4**, 15–22.

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.

Hahn, B., D. G. Ravenhall, and R. Hofstadter (1956), "High-energy electron scattering," *Phys. Rev.* **101**, 1131–1142.

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*, Technical Report SAND91-1634, Sandia National Laboratories, Albuquerque, NM.

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 (Eds.) (1991), *Electron Probe Quantitation* (Plenum Press, New York).

Heitler, W. (1954), *The Quantum Theory of Radiation* (Oxford University Press, London).

Hellekalek, P. (1998), "Good random number generators are (not so) easy to find," *Mathematics and Computers in Simulation* **46**, 485–505.

Hirayama, H., Y. Namito, A. F. Bielajew, S. J. Wilderman, and W. R. Nelson (2006), *The EGS5 Code System*, Technical Report SLAC-R-730 (KEK 2005-8), Stanford Linear Accelerator Center, Menlo Park, California.

Hubbell, J. H. (1989), *Bibliography and current status of K, L, and higher shells fluorescence yields for computations of photon energy-absorption coefficients*, Technical 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 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., W. 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.

Hubbell, J. H., W. J. Veigele, E. A. Briggs, R. T. Brown, D. T. Cromer, and R. J. Howerton (1977), "Erratum: Atomic form factors, incoherent scattering functions, and photon scattering cross sections," *J. Phys. Chem. Ref. Data* **6**, 615–616.

ICRU Report 37 (1984), *Stopping Powers for Electrons and Positrons* (ICRU, Bethesda, MD).

ICRU Report 77 (2007), *Elastic Scattering of Electrons and Positrons* (ICRU, Bethesda, MD).

ICRU Report 85 (2011), *Fundamental Quantities and Units for Ionizing Radiation* (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), "Adenda: 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.* **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), available from `www.nist.gov/srd/nist64.cfm`.

Jackson, J. D. (1975), *Classical Electrodynamics* (John Wiley and Sons, New York), 2nd edition.

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), *Monte Carlo Transport of Electrons and Photons* (Plenum, New York).

Kalos, M. H. and P. A. Whitlock (2008), *Monte Carlo Methods* (Wiley-VCH, Weinheim).

Kawrakow, I. and D. W. O. Rogers (2001), *The EGSnrc code system: Monte Carlo simulation of electron and photon transport*, Technical Report PIRS-701, National Research Council of Canada, Ottawa.

Kellerer, S., K. Hahn, and H. H. Rossi (1992), "Intermediate dosimetric quantities," *Radiation Research* **130**, 15–25.

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.* **87**, 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.

Kissel, L., B. Zhou, S. C. Roy, S. K. S. Gupta, and R. H. Pratt (1995), "The validity of form-factor, modified-form-factor and anomalous-scattering-factor approximations in elastic-scattering calculations," *Acta Cryst.* **A51**, 271–288.

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.

Kövér, L. (2010), "X-ray photoelectron spectroscopy using hard x-rays," *J. Elect. Spect. Relat. Phenom.* **178–179**, 241–257.

Landau, L. D. (1944), "On the energy loss of fast particles by ionization," *Journal of Physics-USSR* **8**, 201–205.

L'Ecuyer, P. (1988), "Efficient and portable combined random number generators," *Commun. ACM* **31**, 742–749.

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. (1954), "On the properties of a gas of charged particles," *Dan. Mat. Fys. Medd.* **28**, 1–57.

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.

Llovet, X., C. J. Powell, A. Jablonski, and F. Salvat (2014a), "Cross sections for inner-shell ionization by electron impact," *J. Phys. Chem. Ref. Data* **43**, 013102.

Llovet, X., F. Salvat, F. Salvat-Pujol, A. Jablonski, and C. J. Powell (2014b), "NIST Database of Cross Sections for Inner-Shell Ionization by Electron or Positron Impact," NIST NSRDS 164, National Institute of Standards and Technology, Gaithersburg, MD, available from `http://dx.doi.org/10.6028/NIST.NSRDS.164`.

Manson, S. T. (1972), "Inelastic collisions of fast charged particles with atoms: ionization of the aluminum L shell," *Phys. Rev. A* **6**, 1013–1024.

Maron, M. J. (1982), *Numerical Analysis: A Practical Approach* (Macmillan, New York).

McMaster, W. H. (1954), "Polarization and the Stokes parameters," *A. J. Phys.* **22**, 351–362.

Meagher, D. (1982), "Geometric modeling using octree encoding," *Computer Graphics and Image Processing* **19**, 129–147.

Merzbacher, E. (1970), *Quantum Mechanics* (John Wiley and Sons, New York), 3rd edition.

Møller, C. (1932), "Zur Theorie des Durchgangs schneller Elektronen durch Materie," *Ann. Physik* **14**, 531–585.

Mohr, P. J., B. N. Taylor, and D. B. Newell (2010), "CODATA recommended values of the fundamental physical constants: 2010," *Rev. Mod. Phys.* **84**, 1527–1605.

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.

Mott, N. F. and H. S. W. Massey (1965), *The Theory of Atomic Collisions* (Oxford University 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 (1993), "Implementation of linearly-polarized photon scattering into the EGS4 code," *Nucl. Instrum. Meth. A* **332**, 277–283.

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.

Negreanu, C., X. Llovet, R. Chawla, and F. Salvat (2005), "Calculation of multiple-scattering angular distributions of electrons and positrons," *Radiat. Phys. Chem.* **74**, 264–281.

Nelson, W. R., H. Hirayama, and D. W. O. Rogers (1985), *The EGS4 Code System*, Technical Report SLAC-265, Stanford Linear Accelerator Center, Stanford, California.

Page, C. G. (1988), *Professional Programmer's Guide to Fortran 77* (University of Leicester, Leicester, UK), available from `www.star.le.ac.uk/~cgp/fortran.html`.

Palik, E. D. (editor) (1985), *Handbook of Optical Constants of Solids* (Academic Press, San Diego, CA).

Palik, E. D. (editor) (1991), *Handbook of Optical Constants of Solids II* (Academic Press, San Diego, CA).

Palik, E. D. (editor) (1998), *Handbook of Optical Constants of Solids III* (Academic Press, San Diego, 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*, Technical Report UCRL-ID-50400, Lawrence Livermore National Laboratory, Livermore, California.

Poškus, A. (2018), "BREMS: A program for calculating spectra and angular distributions of bremsstrahlung at electron energies less than 3 MeV," *Comput. Phys. Commun.* **232**, 237–255, the computer code is available from `http://dx.doi.org/10.17632/mvd57skzd9.1`.

Pratt, R. H. (1960a), "Atomic photoelectric effect at high energies," *Phys. Rev.* **117**, 1017–1028.

Pratt, R. H. (1960b), "Photoeffect from the L shell," *Phys. Rev.* **119**, 1619–1626.

Pratt, R. H., A. Ron, and H. K. Tseng (1973a), "Atomic photoelectric effect above 10 keV," *Rev. Mod. Phys.* **45**, 273–325.

Pratt, R. H., A. Ron, and H. K. Tseng (1973b), "Erratum: Atomic photoelectric effect above 10 keV," *Rev. Mod. Phys.* **45**, 663–664.

Pratt, R. H., H. K. Tseng, C. M. Lee, and L. Kissel (1977), "Bremsstrahlung energy spectra from electrons of kinetic energy 1 keV $\leq T_1 \leq$ 2000 keV incident on neutral atoms $2 \leq Z \leq 92$," *At. Data Nucl. Data Tables* **20**, 175–209.

Pratt, R. H., H. K. Tseng, C. M. Lee, and L. Kissel (1981), "Erratum: Bremsstrahlung energy spectra from electrons of kinetic energy 1 keV $\leq T_1 \leq$ 2000 keV incident on neutral atoms $2 \leq Z \leq 92$," *At. Data Nucl. Data Tables* **26**, 477–481.

Press, W. H. and S. A. Teukolski (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).

Reimer, L., U. Zepke, J. Moesch, S. Schulze-Hillert, M. Ross-Messemer, W. Probst, and E. Weimer (1992), *EEL Spectroscopy* (Carl Zeiss, Oberkochen, Germany).

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 $(\mathrm{d}\sigma/\mathrm{d}\Omega')_{\mathrm{incoh}}$ by means of a pocket calculator," *Phys. Rev. A* **26**, 3325–3333.

Richtmyer, F. K., S. W. Barnes, and E. Ramberg (1934), "The widths of the L-series lines and of the energy levels of Au(79)," *Phys. Rev.* **15**, 843–860.

Rubinstein, R. Y. (1981), *Simulation and the Monte Carlo Method* (Wiley, New York).

Sabbatucci, L. and F. Salvat (2016), "Theory and calculation of the atomic photoeffect," *Radiat. Phys. Chem.* **121**, 122–140.

Sakurai, J. J. (1967), *Advanced Quantum Mechanics* (Addison and Wesley, New York).

Sakurai, J. J. (1997), *Modern Quantum Mechanics* (University of Bangalore Press, New Delhi), 3rd edition.

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. (2015), "The PENELOPE code system. Specific features and recent improvements," *Annals of Nuclear Energy* **82**, 98–109.

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

Salvat, F. and J. M. Fernández-Varea (2019), "RADIAL: a Fortran subroutine package for the solution of the radial Schrdinger and Dirac wave equations," *Comput. Phys. Commun.* **240**, 165–177.

Salvat, F., J. M. Fernández-Varea, J. Sempau, and X. Llovet (2006), "Monte Carlo simulation of bremsstrahlung emission by electrons," *Radiat. Phys. Chem.* **75**, 1201–1219.

Salvat, F., A. Jablonski, and C. J. Powell (2005), "ELSEPA–Dirac partial-wave calculation of elastic scattering of electrons and positrons by atoms, positive ions and molecules," *Comput. Phys. Commun.* **165**, 157–190.

Salvat, F., J. D. Martínez, R. Mayol, and J. Parellada (1987), "Analytical Dirac-Hartree-Fock-Slater screening function for atoms (Z = 1–92)," *Phys. Rev. A* **36**, 467–474.

Sauter, F. (1931), "Über den atomaren Photoeffekt in der K-Schale nach der relativistischen Wellenmechanik Diracs," *Ann. Phys.* **11**, 454–488.

Schiff, L. I. (1951), "Energy-angle distribution of thin target bremsstrahlung," *Phys. Rev.* **83**, 252–253.

Schiff, L. I. (1968), *Quantum Mechanics* (McGraw-Hill, Tokyo).

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. (1973), *Theoretical photoionization cross sections from 1 to 1500 keV*, Technical Report UCRL-51326, Lawrence Livermore Laboratory, Livermore, California.

Scofield, J. H. (1978), "K- and L-shell ionization of atoms by relativistic electrons," *Phys. Rev. A* **18**, 963–970.

Seltzer, S. M. (1993), "Calculation of photon mass energy-transfer and mass energy-absorption coefficients," *Radiat. Research* **136**, 147–170, tabulations of μ/ρ and $\mu_{\rm en}/\rho$ for the elements and for some compounds can be obtained from `www.nist.gov/pml/data/xraycoef/index.cfm`.

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* **13**, 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.

Sempau, J., S. Wilderman, and A. F. Bielajew (2000), "DPM, a fast, accurate Monte Carlo code optimized for photon and electron radiotherapy treatment planning dose calculations," *Phys. Med. Biol.* **45**, 2263–2291.

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.

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.

Sternheimer, R. M., S. M. Seltzer, and M. J. Berger (1983), "Erratum: Density effect for the ionization loss of charge particles in various substances," *Phys. Rev. B* **27**, 6971.

Sternheimer, R. M., S. M. Seltzer, and M. J. Berger (1984), "Density effect for the ionization loss of charge particles in various substances," *At. Data and Nucl. Data Tables* **26**, 261–271.

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.

Tolhoek, H. A. (1956), "Electron polarization, theory and experiment," *Rev. Mod. Phys.* **28**, 277–298.

Tsai, Y. S. (1974), "Pair production and bremsstrahlung of charged leptons," *Rev. Mod. Phys.* **46**, 815–851.

Tsai, Y. S. (1977), "Erratum: Pair production and bremsstrahlung of charged leptons," *Rev. Mod. Phys.* **49**, 421–423.

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.

Waller, I. and D. R. Hartree (1929), "On the intensity of total scattering of X-rays," *Proc. R. Soc. London A* **124**, 119–142.

Wentzel, G. (1927), "Zwei Bemerkungen über die Zerstreuung korpuskularer Strahlen als Beugungserscheinung," *Z. Phys.* **42**, 590–593.

Wightman, A. (1948), "Note on polarization effects in Compton scattering," *Phys. Rev.* **74**, 1813–1817.

Woodcock, E., T. Murphy, P. Hemmings, and S. Longworth (1965), "Techniques used in the GEM code for Monte Carlo neutronics calculations in reactors and other systems of complex geometry", Technical Report ANL-7050, *Proc. Conf. on Applications of Computing Methods to Reactor Problems*, Argonne National Laboratories, Argonne, IL.

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 B. Alder, S. Fernbach, and M. Rotenberg (editors), "*Methods in Computational Physics*," volume 1, pages 89–134 (Academic Press, New York).

Zheng-Ming, L. and A. Brahme (1993), "An overview of the transport theory of charged particles," *Radiat. Phys. Chem.* **41**, 673–703.