Fundamentals of the Monte Carlo method
for
neutral and charged particle transport

Alex F Bielajew
The University of Michigan
Department of Nuclear Engineering and Radiological Sciences
2927 Cooley Building (North Campus)
2355 Bonisteel Boulevard
Ann Arbor, Michigan 48109-2104
U. S. A.
Tel: 734 764 6364
Fax: 734 763 4540
email: bielajew@umich.edu

February 11, 2000

Preface

This book arises out of a course I am teaching for a two-credit (26 hour) graduate-level course *Monte Carlo Methods* being taught at the Department of Nuclear Engineering and Radiological Sciences at the University of Michigan.

AFB, February 11, 2000

Contents

Chapter 1

What is the Monte Carlo method?

The Monte Carlo method is a numerical solution to a problem that models objects interacting with other objects or their environment based upon simple object-object or object-environment relationships[1]. It represents an attempt to model nature through direct simulation of the essential dynamics of the system in question. In this sense the Monte Carlo method is essentially simple in its approach—a solution to a macroscopic system through simulation of its microscopic interactions.

A solution is determined by random sampling of the relationships, or the microscopic interactions, until the result converges. Thus, the mechanics of executing a solution involves repetitive action or calculation. To the extent that many microscopic interactions can be modelled mathematically, the repetitive solution can be executed on a computer. However, the Monte Carlo method predates the computer (more on this later) and is not essential to carry out a solution although in most cases computers make the determination of a solution much faster.

There are many examples of the use of the Monte Carlo method that can be drawn from social science, traffic flow, population growth, finance, genetics, quantum chemistry, radiation sciences, radiotherapy, and radiation dosimetry but our discussion will concentrate on the simulation of neutrons, photons and electrons being transported in condensed materials, gases and vacuum. We will make brief excursions into other kinds of Monte Carlo methods when they they serve to elucidate some point or when there may be a deeper connection to particle-matter interactions or radiation transport in general.

In some cases, the microscopic interactions are not well known. For example, a Monte Carlo calculation of the seating patterns of the members of an audience in an auditorium may

[1] This presupposes that all uses of the Monte Carlo are for the purposes of understanding physical phenomena. There are others uses of the Monte Carlo method for purely mathematical reasons, such as the determination of multi-dimensional integrals, a topic that will be discussed later in Chapter 2. Often these integrals are motivated by physical models. However, there are examples where the motivation is entirely mathematical in which case our definition of the Monte Carlo method would have to be generalized somewhat.

require that the researcher make assumptions about where an audience member would like to sit and attempt to factor in other phenomena such as: a) for some type of performances, people arrive predominantly in pairs, b) audience members prefer an unobstructed view of the stage, c) audience members prefer to sit in the middle, close to the front, *etc.* Each one of these assumptions could then be tested through measurement and then refined. The Monte Carlo method in this case is an adjunct to the basic theory, providing a mechanism to facilitate its development. An example is given in Figure 1.1.

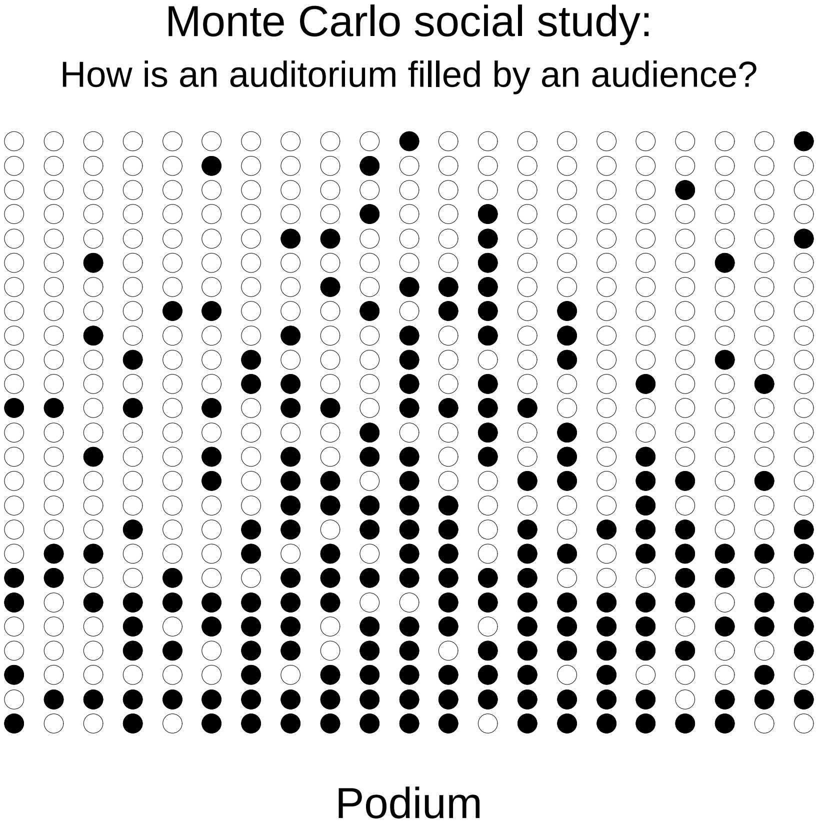
Monte Carlo social study:
How is an auditorium filled by an audience?
Podium

Figure 1.1: Simulation of a seating arrangement in a partially filled small auditorium. An occupied seat is represented by a solid circle and an empty seat by an open circle. The audience members were given a preference to sit in the middle and towards the front with the constraint that only one person could occupy a seat. (This constraint is what makes the mathematical solution difficult but is easy to simulate using Monte Carlo methods.)

The important role that Monte Carlo methods have to play in this sort of study is illustrated in Figure 1.2. Basic science attempts to understand the basic working mechanisms of a phenomenon. The "theory" is a set of assumptions (with perhaps a mathematical formulation of these assumptions) that can by a measured in an "experiment". Ideally, the connection between theory and experiment is direct so that the interpretation of the experiment in unambiguous. This happens when the mathematical description of the microscopic interactions and the macroscopic measurement involves no further approximation. When this is not

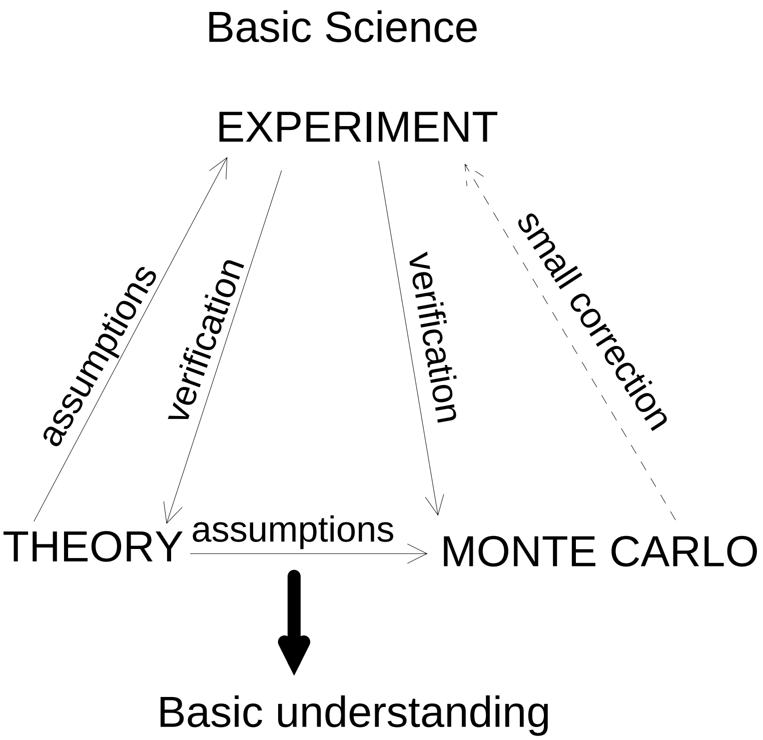

Figure 1.2: The role of Monte Carlo methods in basic science.

possible, as in the example of Figure 1.1 where two people can not occupy the same seat, a Monte Carlo simulation enters the picture in a useful way and can serve a two-fold purpose. It can either provide a small correction to an otherwise useful theory or it can be employed directly to verify or disprove the theory of microscopic interactions.

In some cases, the microscopic interactions are well-known as in the electromagnetic interactions of high-energy electrons and photons and verified by experiment. Monte Carlo techniques in this field are useful for predicting the trajectories of high-energy particles through detectors and other complex assemblies of materials. As an example, consider the experiment by MacPherson, Ross and Rogers [MRR95, MRR96] to measure the stopping power of electrons. The stopping power is the differential amount of energy, dE deposited in a differential pathlength dx through some material. Stopping power is a function of the electron energy and the material. The experimental set-up is depicted in Figure 1.3.

Nearly-monoenergetic electrons from a linear accelerator (LINAC) are first scattered in the forward direction by the thin exit window of the LINAC (this produces negligible energy loss). The energy of the electrons are this point is known from a separate experiment. The electron pass through a “thin” foil and are collected in a large NaI detector. The NaI detector completely absorbs the electrons (except the occasional one that is backscattered) although the bremsstrahlung photons produced in slowing down the electrons can escape.

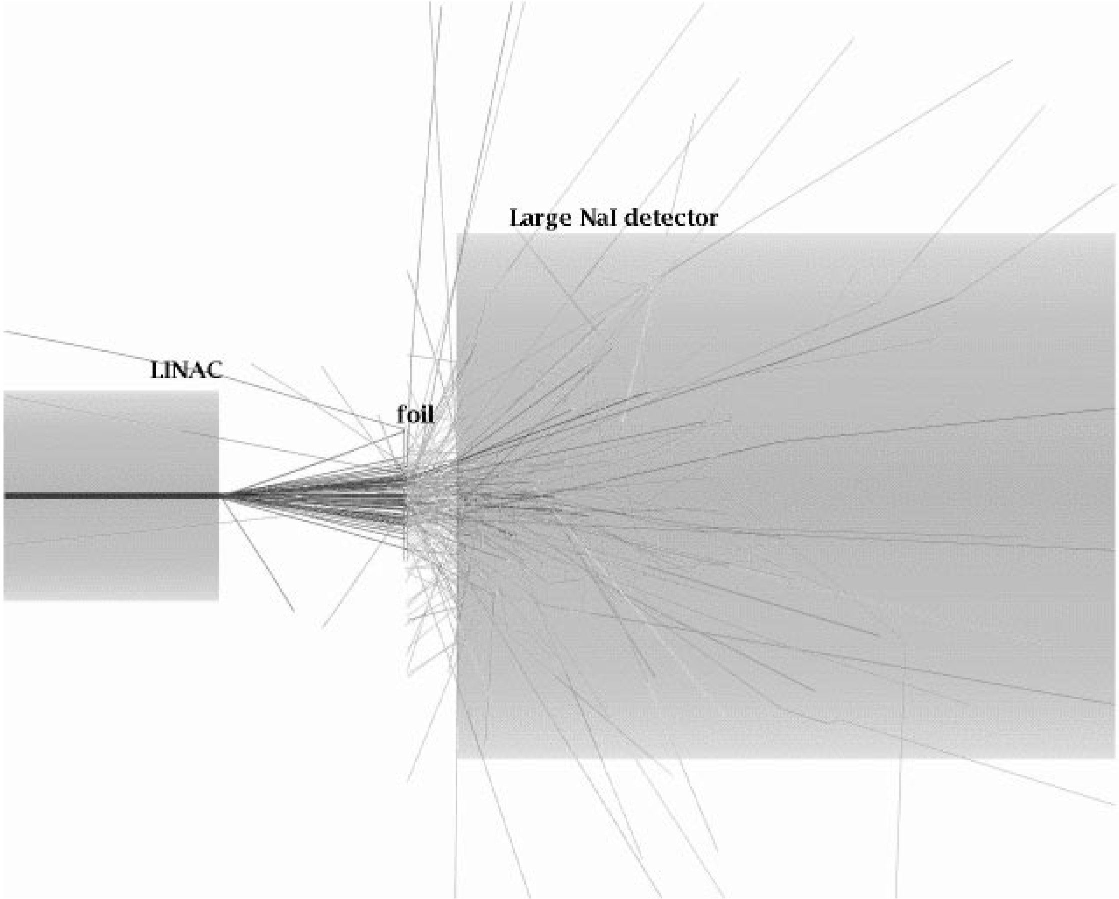

Figure 1.3: Experimental set-up of the MacPherson *et al.* stopping-power measurement.

The photon trajectories are represented by straight-line segments in the figure[2]. A close-up of the electron and photon trajectories in the foil is represented in Figure 1.4. In this case, 20 MeV electrons passed through a thin plastic foil.

We can note several features: 1) the electron trajectories through the foil are nearly straight (there is little elastic scattering) although the angular distribution following the accelerator's exit window is evident, 2) some electrons backscatter from the foil and the NaI detector, 3) some bremsstrahlung, which is produced in either the foil or the NaI detector, escapes detection, 4) some electrons scatter into large angles in the foil and escape detection. Since the amount of energy lost in the foil is determined by how much energy is captured by the NaI detector *less* the known input energy, items 2)–4) are corrections to the experiment that can be determined by Monte Carlo methods and the skewness of the electron trajectories in

[2] The trajectories depicted in Figure 1.3 and some subsequent ones were produced by the EGS4 code [NHR85, BHNR94] and the system for viewing the trajectories is called EGS_Windows [BW91]. Color renditions of this and subsequent figures make it easier to distinguish the particle species and color versions may be viewed on the web at `http://www-ners.engin.umich.edu/info/bielajew/EWarchive.html`.

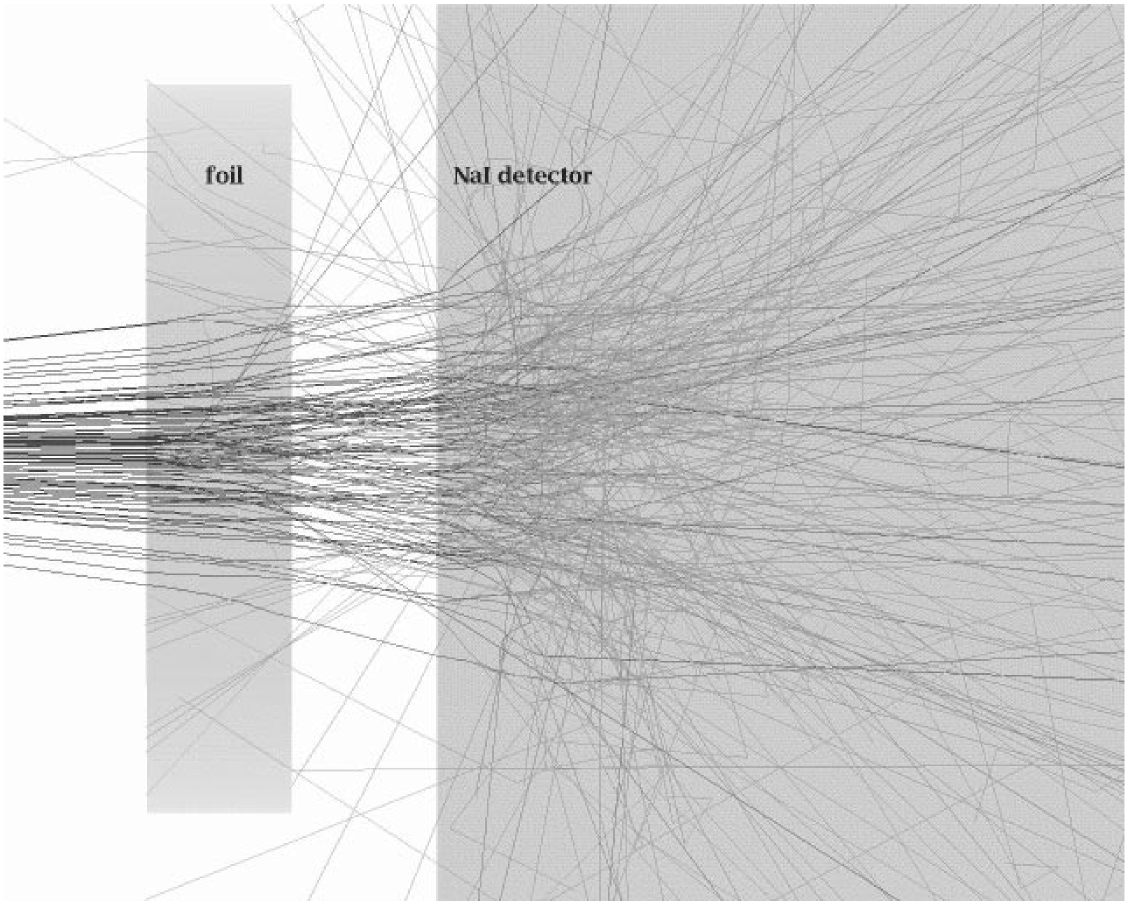

Figure 1.4: A close up of the electron and photon trajectories in a thin foil in the MacPherson *et al.* stopping-power experiment.

the foil would have to be accounted for. However, these corrections are small in this case and the Monte Carlo methods assist in the determination of the stopping power, a basic physics parameter, and the scientific method outlined in Figure 1.2 still applies[3].

If one makes the foil thicker and/or reduces the energy, the electron trajectories become considerably more complicated. A close up of a tungsten foil irradiated by 10 MeV electrons in the same experiment is depicted in Figure 1.5. The electrons develop very curved trajectories owing to multiple Coulomb elastic scattering. Indeed, some of them even stop *in the foil!* Since the corrections in this case would be large (The dx in the stopping power is a measure of the electron pathlength, **not** the thickness of the foil.) and to some extent circular, this measurement is not useful as a determination of the stopping power. However, this does not mean that the energy deposition in the thick foil is not accurate! Assuming

[3] There is some circular logic in using Monte Carlo calculations to assist in the determination of stopping power since the Monte Carlo calculations itself relies upon knowledge of the stopping power. However, if the foil is thin enough, the sensitivity to the assumed stopping power in the Monte Carlo calculations is minimal.

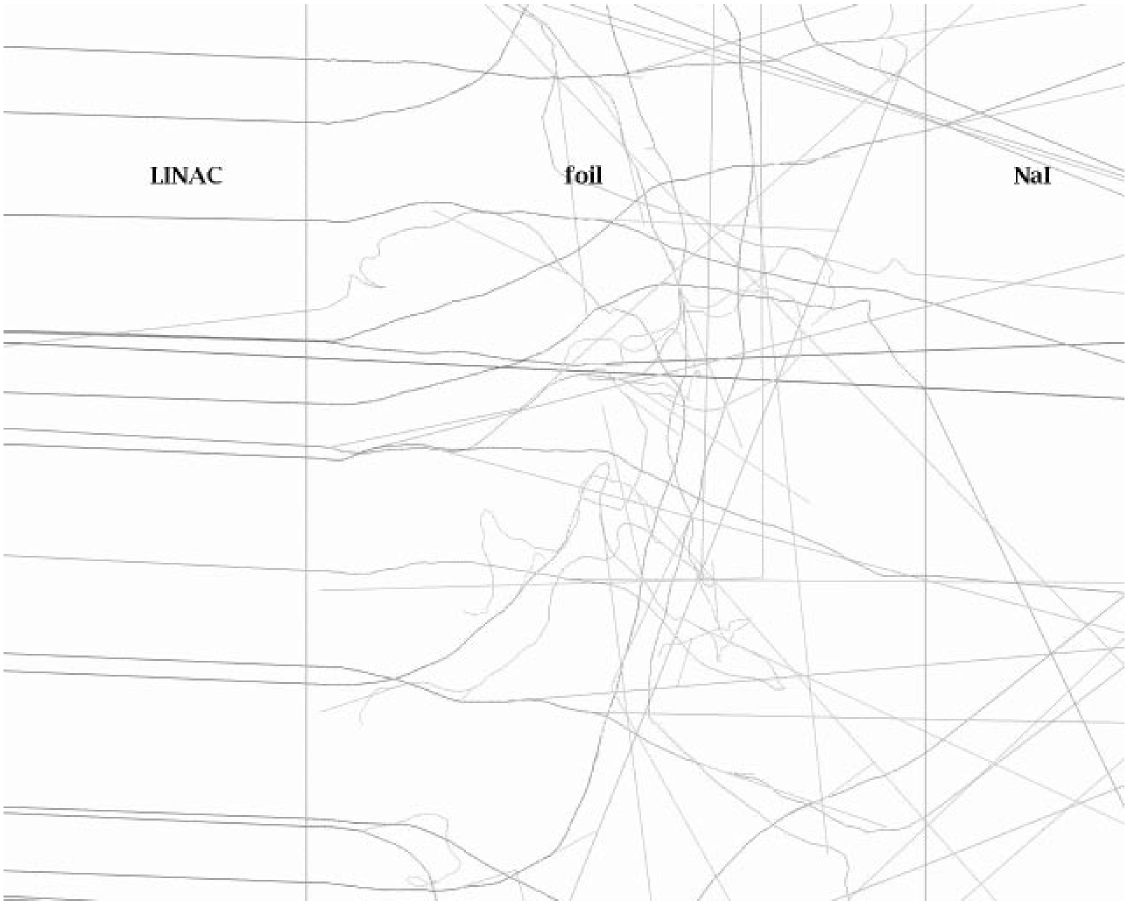

Figure 1.5: A close up of the electron and photon trajectories in a thick foil in the MacPherson *et al.* stopping-power experiment.

we believe the accepted value of the stopping power, the energy deposited in the foil may be predicted with accuracy. However, the calculated result depends in a complicated and coupled way on all the physics of electron and photon transport.

This use of the Monte Carlo method is depicted in Figure 1.6. In this case, theory can not provide a sufficiently precise and entire mathematical description of the microscopic and macroscopic physics. Theory can, however, provide intuition for the design of the measurement. Monte Carlo methods are an adjunct to this process as well, serving in the analysis of the experiment and verifying or invalidating the design.

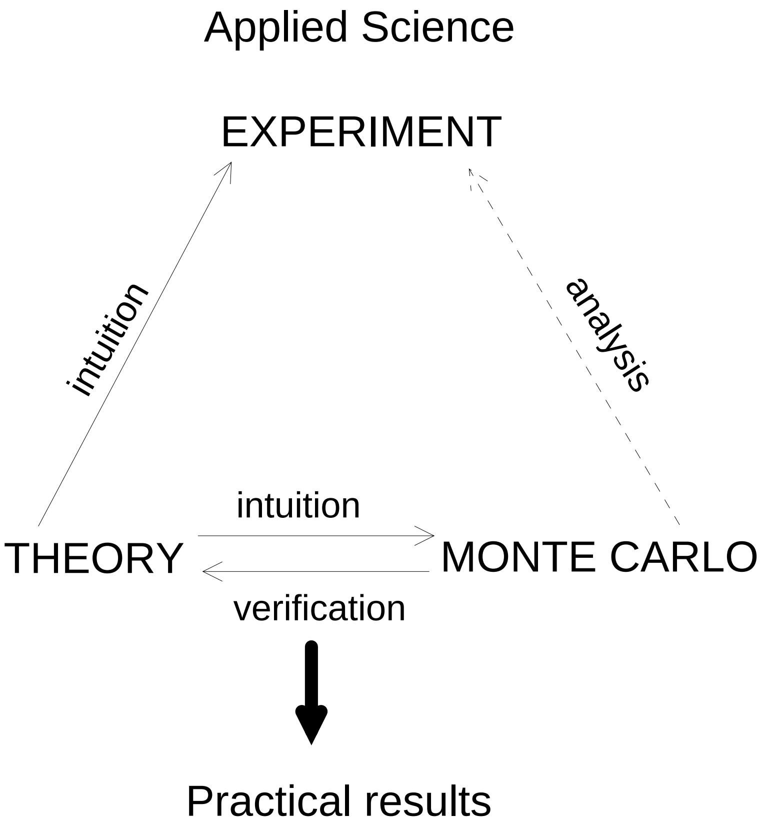

Figure 1.6: The role of Monte Carlo methods in applied science.

1.1 Why is Monte Carlo?

If Monte Carlo did not exist there would be strong motivation to invent it! As argued previously, the products of both basic and applied science are dependent upon the trinity of measurement, theory and Monte Carlo. Monte Carlo is often seen as a "competitor" to other methods of macroscopic calculation, which we will call deterministic and/or analytic methods. Although the proponents of either method sometimes approach a level of fanaticism in their debates, a practitioner of science should first ask, "What do I want to accomplish?" followed by "What is the most efficient way to do it?" Sometimes the correct answer will be "Deterministic" and other times it will be "Monte Carlo". The most successful scientist will avail himself or herself of more than one avenue attack on a problem.

There are, however, two inescapable realities. The first is that macroscopic theory, particularly transport theory, provides deep insight and allows one to develop sophisticated intuition as to how macroscopic particle fields can be expected to behave. Monte Carlo can not compete very well with this. In discovering the properties of macroscopic field behaviour, Monte

Carloists operate very much like experimentalists. Without theory to provide guidance the process of discovery is trial and error, guided perhaps, by some brilliant intuition.

However, when it comes to complexity of a problem, however that is measured, Monte Carlo techniques become advantageous as the complexity of a problem increases. Later in this book, in section 11, a mathematical proof is given. This "idea" is expressed in Figure 1.7.

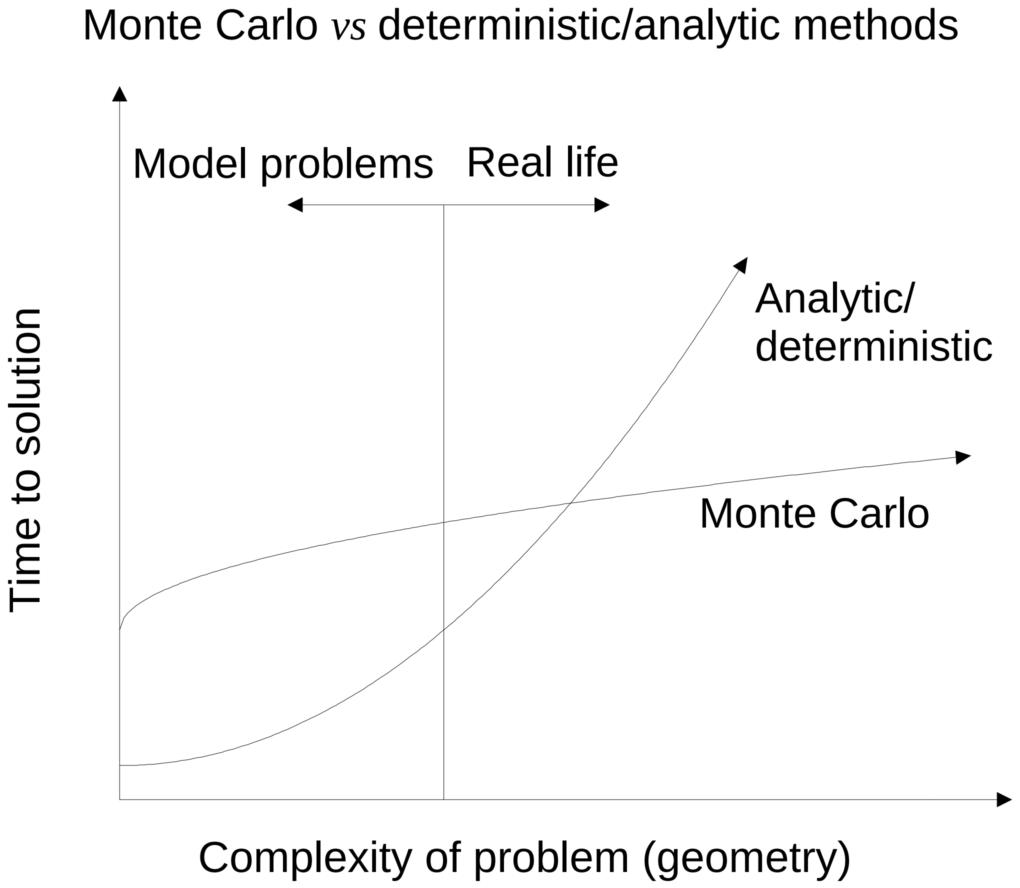

Figure 1.7: Time to solution of Monte Carlo *vs.* deterministic/analytic approaches.

The other inescapable reality is that computers are getting faster and cheaper at an geometric rate. This is known as Moore's Law[4].

A demonstration of Moore's Law for the radiotherapy application has been maintained for about 10 years now, through a timing benchmark of the XYZDOS code [BR92, Bie95], a "user

[4] Gordon E Moore (one of the co-founders of Intel) predicted in 1965 that the transistor density of semiconductor chips would double roughly every 12. This was based upon observation of silicon manufacturing during the previous few years. Moore's Law is not actually a law—it is a prediction based on the observation. Computers doubled in speed/unit cost every 12 months from 1962–1970 and every 18 months thereafter.

code" for the EGS4 Monte Carlo code system [NHR85, NBRH94, BHNR94]. The result for Intel processors is shown in Figure 1.8.

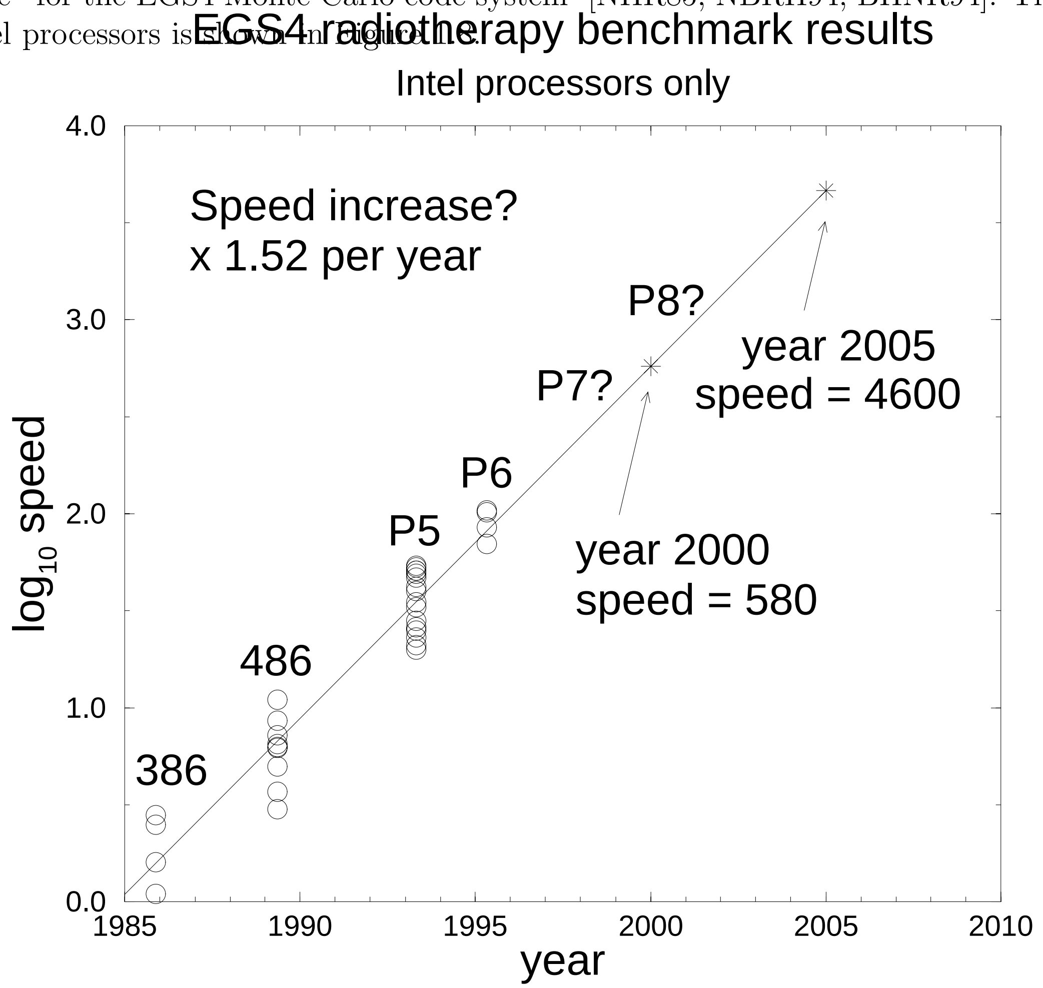

Figure 1.8: An example of Moore's "Law".

The question as to when this geometric growth in computer speed will stop is a topic of some hot debate. The increase in speed is achieved in two ways. As the technology advances, computer circuits can be made smaller and smaller and it takes less time for signals to move within a chip. Smaller size also allows chips to be placed closer together and interchip communication times are proportionally less[5]. The problems associated with smaller size is that heat becomes a problem. This is dealt with by driving these circuits at lower voltages and by mounting these chips in elaborate heat-dissipating assemblies, sometimes with cooling fans mounted directly on the assemblies. The other difficulty with small size is the "quantum limit". Circuit components can become so small that the state of a switch is no longer well-defined (as in the classical case) and there is a probability that a switch, once thrown, could

[5] There's an old joke in computer circles that goes like this: "Company XYZ's computer manufacturing business has become so successful that they are relocating into smaller premises!"

spontaneously revert back causing undesirable effects. This is a more difficult problem that we may see as soon as 2005 or earlier.

Another benefit of small size is that more circuitry can be packed into the same silicon real estate and more sophisticated signal processing algorithms can be implemented. In addition, the design of these algorithms is undergoing continuing research resulting in more efficient processing per transistor. All these effects combine to give us the geometric growth we see in computer speed per unit cost.

Other skeptics argue that market forces do not favor the development of faster, cheaper computers. Historically, the development fast computing was based upon the need for it from science, big industry and the military. The growth of the personal computer industry is related to its appeal to the home market and its accessibility to small business. So successful has the personal computer been that the mainframe computer industry has been squeezed into niche markets. Some predict that eventually consumers, particularly the home markets, will stop driving the demand so relentlessly. *How fast does a home computer have to be?* is a typical statement heard from this perspective. However, applications usually grow to keep up with new technologies, so perhaps this argument is not well-founded. “Keeping up with the Joneses” is still a factor in the home market.

One trend that should not be ignored in this argument is the emergence of new technologies, like semi-optical (optical backplane computers) or completely optical computers. The widespread introduction of this technology might cause computer speeds to exceed the present-day geometric growth!

Another factor weighing in favor of Monte Carlo is that the Monte Carlo technique is one based upon a minimum amount of data and a maximum amount of floating-point operation. Deterministic calculations are often maximum data and minimal floating-point operation procedures. Since impediments to data processing are often caused by communication bottlenecks, either from the CPU to cache memory, cache memory to main memory or main memory to large storage devices (typically disk), modern computer architecture favors the Monte Carlo model which emphasizes iteration and minimizes data storage.

Although the concluding remarks of this section seem to favor the Monte Carlo approach, a point made previously should be re-iterated. Analytic theory development and its realizations in terms of deterministic calculations are our only way of making theories regarding the behaviour of macroscopic fields, and our only way of modelling particle fluences in a symbolic way. Monte Carlo is simply another tool in the theoretician’s or the experimentalist’s toolbox. The importance of analytic development must never be understated.

1.2 Some history

The usual first reference to the Monte Carlo method is usually that of Comte de Buffon [dB77] who proposed a Monte Carlo-like method to evaluate the probability of tossing a needle onto a ruled sheet. This reference goes back to 1777, well before the contemplation of automatic calculating machines. Buffon calculated that a needle of length L tossed randomly on a plane ruled with parallel lines of distance d apart where $d > L$ would have a probability

$$p = \frac{2L}{\pi d} \ . \tag{1.1}$$

A computer simulation of 50 needles (where $d/L = \frac{3}{4}$) on a finite grid of 5 lines is shown in Figure 1.9.

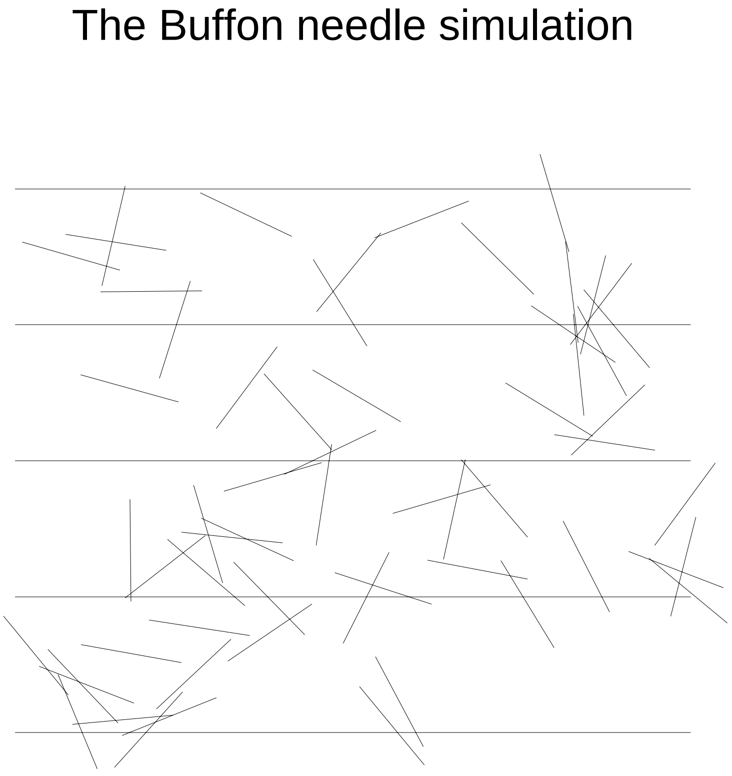

Figure 1.9: A computer simulation of the Buffon needle problem.

Later, Laplace [Lap86] suggested that this procedure could be employed to determine the value of π, albeit slowly. Several other historical uses of Monte Carlo predating computers are cited by Kalos and Whitlock [KW86]. The modern Monte Carlo age was ushered in by von Neumann and Ulam during the initial development of thermonuclear weapons[6]. Ulam and von Neumann coined the phrase "Monte Carlo" and were pioneers in the development of the Monte Carlo technique and its realizations on digital computers[7].

[6] The two books by Richard Rhodes, *The making of the Atomic Bomb* and *Dark Sun* are excellent historical sources for this period.

[7] Despite its chequered history, thermonuclear weapons have never been deployed in conflict. So Monte Carlo calculations have not been employed in a destructive way. In contrast, Monte Carlo calculations are employed for beneficial purposes, such as in the prediction of dose in cancer treatments [Bie94] and should be credited with saving lives.

Bibliography

[BHNR94] A. F. Bielajew, H. Hirayama, W. R. Nelson, and D. W. O. Rogers. History, overview and recent improvements of EGS4. *National Research Council of Canada Report PIRS-0436*, 1994.

[Bie94] A. F. Bielajew. Monte Carlo Modeling in External Electron-Beam Radiotherapy — Why Leave it to Chance? *In "Proceedings of the XI'th Conference on the Use of Computers in Radiotherapy" (Medical Physics Publishing, Madison, Wisconsin)*, pages 2 – 5, 1994.

[Bie95] A. F. Bielajew. EGS4 timing benchmark results: Why Monte Carlo is a viable option for radiotherapy treatment planning. *In "Proceedings of the International Conference on Mathematics and Computations, Reactor Physics, and Environmental Analyses" (American Nuclear Society Press, La Grange Park, Illinois, U.S.A.)*, pages 831 – 837, 1995.

[BR92] A. F. Bielajew and D. W. O. Rogers. A standard timing benchmark for EGS4 Monte Carlo calculations. *Medical Physics*, 19:303 – 304, 1992.

[BW91] A. F. Bielajew and P. E. Weibe. EGS-Windows - A Graphical Interface to EGS. *NRCC Report: PIRS-0274*, 1991.

[dB77] G. Comte de Buffon. *Essai d'arithmétique morale*, volume 4. Supplément à l'Histoire Naturelle, 1777.

[KW86] M. H. Kalos and P. A. Whitlock. *Monte Carlo methods, Volume I: Basics.* John Wiley and Sons, New York, 1986.

[Lap86] P. S. Laplace. Theorie analytique des probabilités, Livre 2. In *Oeuvres complétes de Laplace*, volume 7, Part 2, pages 365 – 366. L'académie des Sciences, Paris, 1886.

[MRR95] M. S. MacPherson, C. K. Ross, and D. W. O. Rogers. A technique for accurate measurement of electron stopping powers. *Med. Phys. (abs)*, 22:950, 1995.

[MRR96] M. S. MacPherson, C. K. Ross, and D. W. O. Rogers. Measured electron stopping powers for elemental absorbers. *Med. Phys. (abs)*, 23:797, 1996.

[NBRH94] W. R. Nelson, A. F. Bielajew, D. W. O. Rogers, and H. Hirayama. EGS4 in '94: A decade of enhancements. *Stanford Linear Accelerator Report SLAC-PUB-6625 (Stanford, Calif)*, 1994.

[NHR85] W. R. Nelson, H. Hirayama, and D. W. O. Rogers. The EGS4 Code System. Report SLAC–265, Stanford Linear Accelerator Center, Stanford, Calif, 1985.

Chapter 2

Elementary probability theory

Fundamental to understanding the operation of a Monte Carlo process and interpreting the results of Monte Carlo calculations, is some understanding of elementary probability theory. In this chapter we introduce some elementary probability theory to facilitate the discussions in the later chapters.

2.1 Continuous random variables

2.1.1 One-dimensional probability distributions

A probability distribution function, $p(x)$ is a measure of the likelihood of observing x. For example, x could be the position at which a photon interacts via the Compton interaction. If $p(x_1) = 2p(x_2)$ then what this actually means is that an observation of x in a differential interval $\mathrm{d}x$ is twice as likely to be observed than in an interval the same size about x_2. This really only has meaning in the limit that $\mathrm{d}x$ goes to zero, or if $\mathrm{d}x$ is finite, when $p(x)$ varies slowly over the interval in the vicinity of x_1 and x_2. A more concrete example for $p(x) = \exp(-x)$ is shown in Figure 2.1.

In general, $p(x)$ has some special properties that distinguishes it from other functions:

- $p(x) \geq 0$ since negative probabilities have no meaning (at least none that we will mention herein),

- $p(x)$ is normalized in the following fashion,

$$\int_{x_{\min}}^{x_{\max}} \mathrm{d}x \; p(x) = 1 \; , \tag{2.1}$$

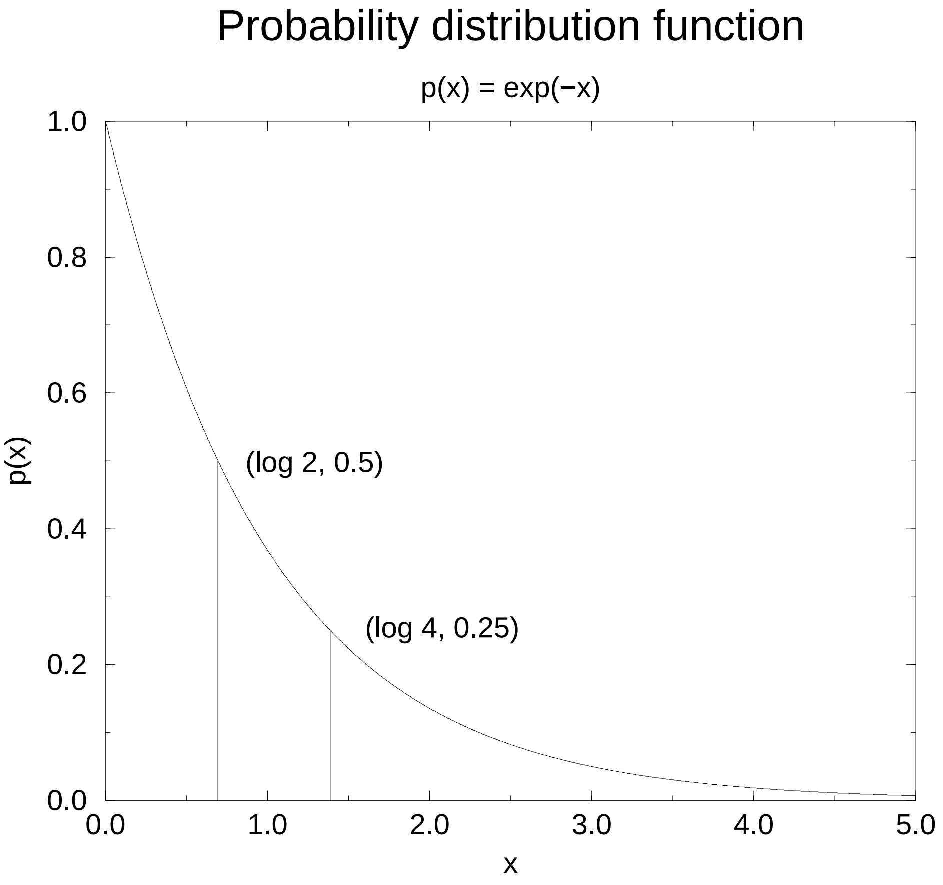

Figure 2.1: The probability function $p(x) = \exp(-x)$. $p(\log 2) = 2p(\log 4)$

- $-\infty < x_{\min} < x_{\max} < +\infty$, that is, $x_{\min}$ and $x_{\max}$ can be any real number so long as $x_{\min}$ is less than $x_{\max}$.

These are the only restrictions on $p(x)$. Note that the statement of normalization above implies that $p(x)$ is integrable over its range of definition. Other than that, it can be discontinuous and even infinite! (The above definition in terms of intervals preserves our interpretation[1]).

Certain probability functions can be characterized in terms of their moments,

$$\langle x^n \rangle = \int_{x_{\min}}^{x_{\max}} \mathrm{d}x \; x^n p(x) \; , \tag{2.2}$$

however, the existence of these moments is not guaranteed nor even necessary. Assuming that $\langle x \rangle$ and $\langle x^2 \rangle$ exist, the variance associated with the probability function may be defined

[1] An example of an infinite probability would be $p(x) = a\delta(x - x_1) + (1 - a)\delta(x - x_2)$, where $\delta()$ is the Dirac delta function.

to be:

$$\mathrm{var}\{x\} = \langle x^2 \rangle - \langle x \rangle^2 \ . \tag{2.3}$$

The variance is a measure of the width of the distribution of x. $\mathrm{var}\{x\}$ is zero for the Dirac delta function and greater than zero for all other probability distribution functions, even combinations of delta functions.

An example of a probability distribution function that has no moments is the Cauchy or Lorentz distribution function[2] defined over the infinite range:

$$p(x) = \frac{1}{\pi} \frac{\Gamma}{\Gamma^2 + x^2} \qquad -\infty < x < \infty. \tag{2.4}$$

The mean value of the Cauchy/Lorentz distribution can be defined to be zero if the "principle value" of the first moment is obtained in the following way:

$$\langle x \rangle = \lim_{a \longrightarrow \infty} \int_{-a}^{a} \frac{1}{\pi} \frac{\Gamma}{\Gamma^2 + x^2} \ , \tag{2.5}$$

but the second moment and hence the variance can not be defined in any fashion.

Another interesting probability distribution function of great importance in the simulation of electron transport is the screened Rutherford or Wentzel distribution function:

$$p(\mu) = \frac{a(2+a)}{2} \frac{1}{(1-\mu+a)^2} \quad ; \quad -1 \le \mu \le 1, \tag{2.6}$$

where μ is the cosine of the scattering angle, $\cos\Theta$.

The conventional small-angle form of the screened Rutherford or Wentzel distribution function is:

$$p(\Theta) = 4a \frac{\Theta}{(\Theta^2 + 2a)^2} \quad ; \quad 0 \le \Theta < \infty, \tag{2.7}$$

Its first moment exists, $\langle \Theta \rangle = \pi\sqrt{a/2}$ but its second moment is infinite! This strange behavior, the nonexistence of an angular variance is responsible for the theory of electron transport being so problematic. Of course, one could restrict the range of integration to physical angles, $0 \le \Theta \le \pi$, but the problems persist.

2.1.2 Two-dimensional probability distributions

Consideration of two and higher-dimensional probability distributions follows from a generalization of one-dimensional distributions with the added features of correlation between observables and conditional probabilities.

[2] More pathological examples could be invented but this distribution arises from an interesting application—the intrinsic probability distribution of the energy of a quantum from an excited atomic state of finite lifetime.

Consider a two-dimensional (or joint) probability function p(x,y). A tangible example is the distribution in energy and angle of a photon undergoing an inelastic collision with an atom[3].

Another example is the two-dimensional probability distribution presented in Figure 2.2. The meaning of two-dimensional probability distributions is as follows: Hold one variable, say x fixed, and the resulting distribution is a probability distribution function in the other variable, y. It as if you cut through the two-dimensional probability distribution at a given point in x and then displayed the “cross cuts”. Several examples of these cross cuts are shown in Figure 2.3.

Figure 2.2: A two-dimensional probability function $f(x,y)$

The notions of normalization and moments follow directly. Thus,

$$\langle x^n y^m \rangle = \int_{y_{\min}}^{y_{\max}} \int_{x_{\min}}^{x_{\max}} \mathrm{d}x\mathrm{d}y \; x^n y^m p(x,y) \; , \tag{2.8}$$

with the normalization condition $\langle x^0 y^0 \rangle = 1$, the only “moment” that must be defined.

[3] “Inelastic” in this definition relates to the energy of the photon changing, not the energy of the atom.

Figure 2.3: Cross cuts of the two-dimensional probability function given in Figure 2.2.

Higher order moments may or may not exist. If they exist, we define the covariance:

$$\mathrm{cov}\{x, y\} = \langle xy \rangle - \langle x \rangle \langle y \rangle \ , \tag{2.9}$$

which can be positive or negative. Note that $\mathrm{cov}\{x, x\} = \mathrm{var}\{x\}$.

The covariance is a measure of the independence of observing x or y. If x and y are independent random variables, then $p(x, y) = p_1(x)p_2(y)$ and $\mathrm{cov}\{x, y\} = 0$. A related function is the correlation coefficient:

$$\rho\{x, y\} = \frac{\mathrm{cov}\{x, y\}}{\sqrt{\mathrm{var}\{x\}\mathrm{var}\{y\}}} \ , \tag{2.10}$$

where $-1 \leq \rho\{x, y\} \leq 1$.

Some interesting relations involving the variances and the covariances may be found. For example,

$$\mathrm{var}\{x \pm y\} = \mathrm{var}\{x\} + \mathrm{var}\{y\} \pm 2\ \mathrm{cov}\{x, y\} \ , \tag{2.11}$$

or simply $\mathrm{var}\{x\} + \mathrm{var}\{y\}$ if x and y are independent.

The marginal probabilities are defined by integrating out the other variables.

$$m(x) = \int_{y_{\mathrm{min}}}^{y_{\mathrm{max}}} \mathrm{d}y \; p(x, y) \quad ; \quad m(y) = \int_{x_{\mathrm{min}}}^{x_{\mathrm{max}}} \mathrm{d}x \; p(x, y) \; . \tag{2.12}$$

Note that the marginal probability distributions are properly normalized. For the example of joint energy and angular distributions, one marginal probability distribution relates to the distribution in energy irrespective of angle and the other refers to the distribution in angle irrespective of energy. This the joint probability distribution function may be written:

$$p(x, y) = m(x)p(y|x) \; , \tag{2.13}$$

where the *conditional* probability is defined by

$$p(y|x) = \frac{p(x, y)}{m(x)} \; . \tag{2.14}$$

The interpretation of the *conditional* probability is that given x, what is the probability that y occurs. The appearance of $m(x)$ in the denominator guaranteed the normalization of $p(y|x)$.

2.1.3 Cumulative probability distributions

Associated with each one-dimensional probability distribution function is its cumulative probability distribution function

$$c(x) = \int_{x_{\mathrm{min}}}^{x} \mathrm{d}x' \; p(x') \; . \tag{2.15}$$

Cumulative probability distribution functions have the following properties which follow directly from its definition and the properties of probability distribution functions:

- $p(x)$ and $c(x)$ are related by a derivative:

$$p(x) = \frac{\mathrm{d}c(x)}{\mathrm{d}x} \; , \tag{2.16}$$

- $c(x)$ is zero at the beginning of its range of definition

$$c(x_{\mathrm{min}}) = 0 \; , \tag{2.17}$$

- and unity at the end of its range of definition

$$c(x_{\mathrm{max}}) = 1 \; , \tag{2.18}$$

- $c(x)$ is a monotonically increasing function of x as a result of $p(x)$ always being positive and the definition of $c(x)$ in Equation 2.15.

Cumulative probability distribution functions can be related to uniform random numbers to provide a way for sampling these distributions. We will complete this discussion in Chapter 4.

Cumulative probability distribution functions for multi-dimensional probability distribution functions are usually defined in terms of the one-dimensional forms of the marginal and conditional probability distribution functions.

2.2 Discrete random variables

A more complete discussion of probability theory would include some discussion of discrete random variables. An example would be the results of flipping a coin or a card game. We will have some small use for this in Chapter 5 and will introduce what we need at that point.

Bibliography

Problems

1. Which of the following are candidate probability distributions? For those that are not, explain. For those that are, determine the normalization constant N. Those that are proper probability distributions, which contain moments that do not exist?

$$f(x) = N \exp(-\mu x) \quad 0 \leq x < \infty$$

$$f(x) = N \exp(-\mu x) \quad 0 \leq x < \Lambda/\mu$$

$$f(x) = N \sin(x) \quad 0 \leq x < \pi$$

$$f(x) = N \sin(x) \quad 0 \leq x < 2\pi$$

$$f(x) = \frac{N}{\sqrt{x}} \quad 0 \leq x < 1$$

$$f(x) = \frac{N}{\sqrt{x}} \quad 1 \leq x < \infty$$

$$f(x) = N \frac{x}{(x^2 + a^2)^{3/2}} \quad 0 \leq x < \infty$$

2. Verify that Equations 2.4, 2.6, and 2.7 are true probability distributions.

3. Prove Equation 2.11. Simplify in the case that x and y are independent.

Chapter 3

Random Number Generation

> *Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin.*
>
> John von Neumann (1951)

The "pseudo" random number generator (RNG) is the "soul" or "heartbeat" of a Monte Carlo simulation. It is what generates the pseudo-random nature of Monte Carlo simulations thereby imitating the true stochastic or random nature of particle interactions. Consequently, much mathematical study has been devoted to RNG's [Ehr81, Knu81, Jam88]. These three references are excellent reviews of RNG theory and methods up to about 1987. The following references contain more modern material [MZT90, MZ91, L9̈4, Jam94, Knu97]. It must also be noted that random number generation is an area of active research. Information that was given last year may be proven to be misleading this year. The best way to stay in tune is to track the discussions concerning this topic on the web sites of organizations for which random number generation is critical. A particularly good one is CERN's site (www.cern.ch). CERN is the European Laboratory for Particle Physics. Monte Carlo applications are quite important in particle physics.

Sometimes one hears the opinion that Monte Carlo codes should be connected somehow to a source of "true" random numbers. Such true random numbers could be produced by the noise in an electronic circuit or the time intervals between decays of a radioactive substance. There are two good reasons **NOT** to do this. Either a piece of hardware producing the random numbers would have to be interfaced to a computer somehow, or an array containing enough random numbers would have to stored. Neither would be practical. However, the most compelling reason for using mathematically-derived pseudo-random numbers is repeatability—essential for code debugging. When a Monte Carlo code matures, error discovery becomes less frequent, errors become more subtle and are revealed after the simulation has run for a long time. Replaying the precise sequence of events that leads to the fault is essential.

We will not endeavor to explain the theory behind random number generation, merely give some guidelines for good use. The operative phrase to be used when considering RNG's is "use extreme caution". **DO USE** an RNG that is known to work well and is widely tested. **DO NOT FIDDLE** with RNG's unless you understand thoroughly the underlying mathematics and have the ability to test the new RNG thoroughly. **DO NOT TRUST** RNG's that come bundled with standard mathematical packages. For example, DEC's `RAN` RNG (a system utility) and IBM's `RANDU` (part of the SSP mathematical package) are known to give strong triplet correlations. This would affect, for example, the "random" seeding of an isotropic distribution of point sources in a 3-dimensional object. A picture of an artefact generated by these RNG's is given in Figure 3.1. This is known as the "spectral" property of LCRNG's.

The gathering of random numbers into planes is a well-known artefact of RNG's. Marsaglia's classic paper [Mar68] entitled "Random numbers fall mainly in the planes", describes how random numbers gather into $(n-1)$-dimensional hyperplanes in n-space. Good RNG's either maximise the number of planes that are constructed to give the illusion of randomness or practically eliminate this artefact entirely. One must be aware of this behaviour in case anomalies do occur. In some cases, despite the shortcoming of RNG's, no anomalies are detected. An example of this is the same data that produced the obvious artefact in Figure 3.1 but displayed with a 10° rotation about the z-axis does not exhibit the artefact. This is shown in Figure 3.2.

3.1 Linear congruential random number generators

Most computer architectures support 32-bit 2's-complement integer arithmetic[1]. The following equation describes a linear congruential random number generator (LCRNG) suitable for machines that employ 2's-complement integer arithmetic:

$$X_{n+1} = \mathrm{mod}(aX_n + c, 2^{32}) \ . \tag{3.1}$$

This LCRNG generates a 32-bit string of random bits X_{n+1} from another representation one step earlier in the cycle, X_n. Upon multiplication or addition, the high-order bits (greater than position 32) are simply lost leaving the low-order bits scrambled in a pseudo-random

[1] In 32-bit 2's-complement integer arithmetic
$00000000000000000000000000000000 = 0$
$00000000000000000000000000000001 = 1$
$00000000000000000000000000000010 = 2 \cdots$
$01111111111111111111111111111111 = 2^{31} - 1 = 2147483647$
$10000000000000000000000000000000 = -2^{31} = -2147483648$
$10000000000000000000000000000001 = -2^{31} + 1 = -2147483647$
$10000000000000000000000000000010 = -2^{31} + 2 = -2147483646 \cdots$
$11111111111111111111111111111111 = -1.$

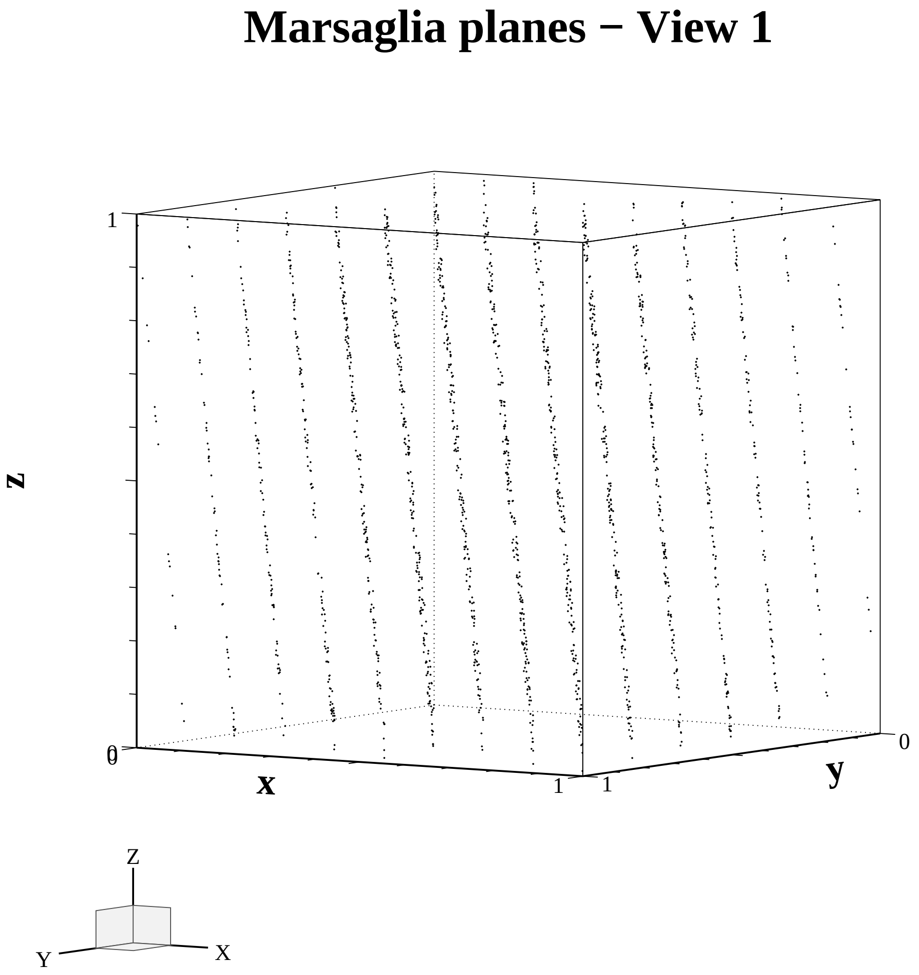
Marsaglia planes − View 1

Figure 3.1: The gathering of random numbers into two-dimensional planes when a three-dimensional cube is seeded.

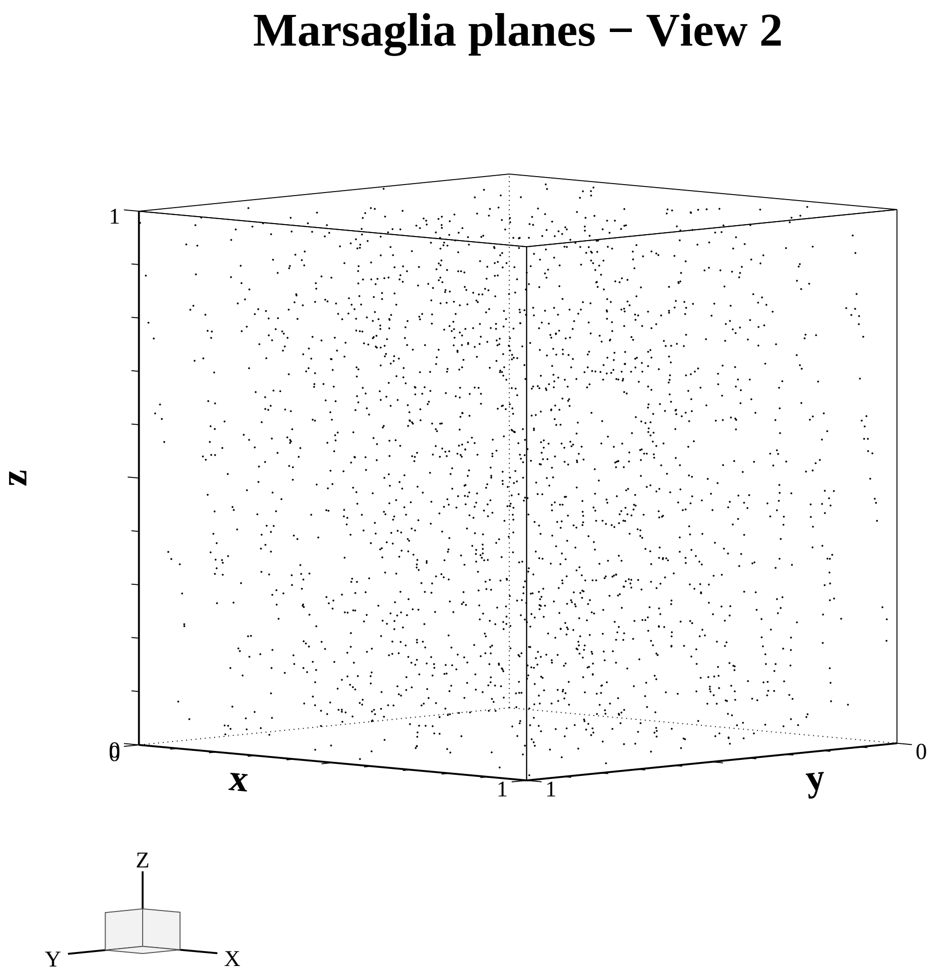

Figure 3.2: The identical data in Figure 3.1 but rotated by 10° about the z-axis.

fashion. In this equation a is a "magic" multiplier and c is an odd number. The operations of addition and multiplication take place as if X_{n+1}, X_n, a, and c are all 32-bit integers.

It is dangerous to consider all the bits as individually random. In fact, the higher-order bits are more random than the lower-order bits. Random bits are more conveniently produced by techniques we will discuss later.

The multiplier a is a **magic** number. Although there are guidelines to follow to determine a good multiplier, optimum ones are determined experimentally. Particularly good examples are $a = 663608941$ and $a = 69069$. The latter has been suggested by Donald Knuth as the "best" 32-bit multiplier. It is also easy to remember!

When c is an odd number, the cycle length of the of the LCRNG is 2^{32} (about 4 billion, in effect, creating every integer in the realm of possibility and an artefactually uniform random number when converted to floating point numbers. When c is set to zero, the LCRNG becomes what is known as a multiplicative congruential random number generator (MCRNG) with a cycle of 2^{30} (about 1 billion) but with faster execution, saving a fetch and an integer addition. The seed, X_0 can be any integer in the case of an LCRNG. In the case of a MCRNG, it is particularly critical to seed the RNG with an odd number. Conventional practice is to use either a large odd number (a "random" seed) or a large prime number. If one seeds the MCRNG with an even integer a reduced cycle length will be obtained. The severity of the truncation is proportional to the number of times the initial seed can be divided by two.

The conversion of the random integer to uniform on the range 0 to 1 contains its subtleties as well and is, to some extent, dependent on architecture. The typical mathematical conversion is:

$$r_n = 1/2 + X_n/2^{32} \ . \tag{3.2}$$

One subtlety to note is that this should be expected to produce a range $0 \leq r_n < 1$ because the integer representation is asymmetric—it has one more negative integer than positive integer.

One might code the initialization, generation and conversion in the following fashion:

```
:Initialisation:    i = 987654321

:Iteration:         i = i * 663608941
                    r = 0.5 + i * 0.23283064e-09
```

Some computers will generate one exact floating-point zero with this algorithm, others may generate 2! One way out of this situation is to make the conversion factor slightly smaller, say 0.2328306e-09. This makes the range to be approximately $10^{-7} \leq r_n < 1 - 10^{-7}$ and appears to work on all computers. Another benefit of this is that it avoids an exact zero which can cause problems in certain sampling algorithms. We will see more on this later. Although the endpoints of the distribution are truncated, it usually has no effect. Moreover, if you are depending upon good uniformity at the endpoints of the random number distribution, then you probably have to resort to special techniques to obtain them.

3.2 Long sequence random number generators

For many practical applications, cycle lengths of one or four billion are just simply inadequate. In fact, a modern workstation just calculating random numbers with these RNG's would cycle them in just a few minutes!

One approach is to employ longer integers! The identical algorithm may be employed with 64-bit integers producing a sequence length of 2^{64} or about 1.84×10^{19} in the case of the LCRNG or 2^{62} or about 4.61×10^{18} for the MCRNG. A well-studied multiplier for this purpose is $a = 6364136223846793005$. This 64-bit RNG is an excellent choice for the emerging 64-bit architectures (assuming that they support 64-bit 2's complement integer arithmetic) or it can be "faked" using 32-bit architectures [Bie86]. However, this latter approach is no longer recommended since more powerful long-sequence RNG's have been developed.

A new approach called the "subtract-with-borrow" algorithm was developed by George Marsaglia and co-workers [MZT90, MZ91]. This algorithm was very attractive. It was implemented in floating-point arithmetic and was portable to all machines. Initialization and restart capabilities are more involved than for LCRNG's but the long sequence lengths and speed of execution make it worthwhile.

Tezuka, l'Ecuyer and Couture [TlC93, Tez94, Cl96] proved that Marsaglia's algorithm is in fact equivalent to a linear congruential generator but with a very big integer word length which greatly reduced the "spectral" anomalies. Since LCRNG's are so well-studied this is actually a comfort since for a while the "subtract-with-borrow" algorithm were being employed with little theoretical understanding, something many researchers felt uncomfortable with.

The spectral property of this new class of generator has been addressed by Lüscher [L9̈4] who used Kolmogorov's chaos theory to show how the algorithm could be improved by skipping some random numbers.

A version of Lüscher's algorithm [Jam94] will be distributed with the source library routines associated with this course. However, keep in mind that the mathematics of random number generation is still in its infancy. If you take up a Monte Carlo project at some time in the future, make sure you become well-informed as to the "state-of-the-art".

An example of Marsaglia and Zaman's "subtract-with-borrow" algorithm is given as follows:

```
C
C     Declarations:
C     =============

C     These variables, 100 floating point numbers and 2 integers define the
C     "state" of the RNG at any point

      real u(97), c, cd, cm
      integer ixx, jxx

C
C     Initialisation:
C     ===============

      if((ixx.le.0).or.(ixx.gt.31328)) ixx = 1802 ! Sets Marsaglia default
      if((jxx.le.0).or.(jxx.gt.30081)) jxx = 9373 ! Sets Marsaglia default

      i = mod(ixx/177,177) + 2
      j = mod(ixx,    177) + 2
      k = mod(jxx/169,178) + 1
      l = mod(jxx,    169)
      do ii = 1,97
          s = 0.0
          t = 0.5
          do jj = 1,24
              m = mod(mod(i*j,179)*k,179)
              i = j
              j = k
              k = m
              l = mod(53*l + 1,169)
              if(mod(l*m,64).ge.32) s = s + t
              t = 0.5*t
          end do
          u(ii) = s
      end do
      c   =   362436./16777216.
      cd  =  7654321./16777216.
      cm  = 16777213./16777216.
      ixx = 97
      jxx = 33

C
```

```
C     Iteration:
C     ==========

      rng = u(ixx) - u(jxx)
      if (rng.lt.0.) rng = rng + 1.
      u(ixx) = rng
      ixx = ixx - 1
      if(ixx.eq.0) ixx = 97
      jxx = jxx - 1
      if(jxx.eq.0) jxx = 97
      c = c - cd
      if (c.lt.0.) c = c + cm
      rng = rng - c
      if (rng.lt.0.) rng = rng + 1.
```

Although the initialization scheme is quite involved, it only has to be done once. The "state" of the random number generator at any time is defined by the 100 floating point numbers, `u(97), c, cd, cm` and the array indices `ixx, jxx` which are employed as pointers into the `u()` array. `ixx, jxx` also serve a double role as initializing seeds. The sequence length is 2^{144} (about 2×10^{43}), long enough for any practical calculation. A few years ago the prevailing opinion was that a a unique set of the two starting seeds such that $0 <$ `ixx` ≤ 31328 and $0 <$ `jxx` ≤ 30081 would produce an *independent* random sequence. In view of the work by Tezuka, l'Ecuyer and Couture this information must be regarded with some suspicion. In fact, now provides 100 seedings for their version of the "subtract-with-borrow" algorithm with a guaranteed subsequence length of at least 2×10^9 (before running into the sequence from another seeding).

At least for this course, existing RNG's will suffice. For the purpose of large-scale, multi-dimensional, massively parallel applications, it appears that there is still fundamental work to be done.

Bibliography

[Bie86] A. F. Bielajew. RNG64 - A 64-bit random number generator for use on a VAX computer. *National Research Council of Canada Report PIRS-0049*, 1986.

[Cl96] R. Couture and P. l'Ecuyer. Orbits and lattices for linear random number generators with composite moduli. *Mathematics of Computation*, 65:189 – 201, 1996.

[Ehr81] J. R. Ehrman. The care and feeding of random numbers. *SLAC VM Notebook, Module 18, SLAC Computing Services*, 1981.

[Jam88] F. James. A review of pseudorandom number generators. *CERN-Data Handling Division, Report DD/88/22*, 1988.

[Jam94] F. James. RANLUX: A FORTRAN implementation of the high-quality pseudorandom number generator of Lüscher. *Computer Physics Communications*, 79:111 – 114, 1994.

[Knu81] D. E. Knuth. *Seminumerical algorithms*, volume II of *The art of computer programming*. Addison Wesley, Reading Mass., 1981.

[Knu97] D. E. Knuth. *Seminumerical algorithms*, volume II of *The art of computer programming*. Addison Wesley, Reading Mass., 1997.

[L9̈4] M. Lüscher. A portable high-quality random number generator for lattice field theory simulations. *Computer Physics Communications*, 79:100 – 110, 1994.

[Mar68] G. Marsaglia. Random numbers fall mainly in the planes. *Nat. Acad. Sci.*, 61:25 – 28, 1968.

[MZ91] G. Marsaglia and A. Zaman. A new class of random number generators. *Annals of Applied Probability*, 1:462 – 480, 1991.

[MZT90] G. Marsaglia, A. Zaman, and W. W. Tsang. Toward a universal random number generator. *Statistics and Probability Letters*, 8:35 – 39, 1990.

[Tez94] S. Tezuka. A unified view of long-period random number generators. *Journal of the Operations Research Society of Japan*, 37:211 – 227, 1994.

[TlC93] S. Tezuka, P. l'Ecuyer, and R. Couture. On the lattice structure of the add-with-carry and subtract-with-borrow random number generators. *ACM Transactions on Modeling and Computer Simulation*, 3:315 – 331, 1993.

Problems

Using a computer, operating system, computing language and compiler of your choice (tell me what you used), as long as it supports 32-bit 2's complement integer arithmetic:

1. Assign or otherwise make an integer adopt the values 0, 1, 2, $2^{31}-2$, $2^{31}-1$, 2^{31}, -($2^{31}-2$), -($2^{31}-1$), -(2^{31}) and have the computer output the integer values and the floating-point representations after conversion $r = 0.5 + i * 0.23283064e - 09$.

2. Write a computer code that simply performs the following:

   ```
   :Initialisation:    i = 987654321

   :Iteration:         i = i * 663608941
   ```

   Verify that the sequence length is 2^{30} by seeing that `i` returns to its original seed. How much CPU time does it take to go through the whole sequence?

Chapter 4

Sampling Theory

Sir,
In your otherwise beautiful poem (The Vision of Sin) there is a verse which reads

> *"Every moment dies a man,*
> *every moment one is born."*

Obviously, this cannot be true and I suggest that in the next edition you have it read

> *"Every moment dies a man,*
> *every moment* $1\frac{1}{16}$ *is born."*

Even this value is slightly in error but should be sufficiently accurate for poetry.

...Charles Babbage (in a letter to Lord Tennyson)

Now that we have tackled the essentials of elementary probability theory and random number generation, it is now time to connect the two and demonstrate how random numbers may be employed to sample from probability distributions.

We will consider three kinds of sampling techniques, the direct approach, the rejection technique and the mixed method that combines the two. Then, we we go through a small catalogue of examples.

4.1 Invertible cumulative distribution functions (direct method)

A typical probability distribution function is shown in Figure 4.1. It is defined over the range $[a, b]$ where neither a nor b are necessarily finite. A probability distribution function *must* have the properties that it is integrable (so that one can normalise it by integrating it over its entire range) and that it is non-negative. (Negative probability distributions are difficult to interpret.)

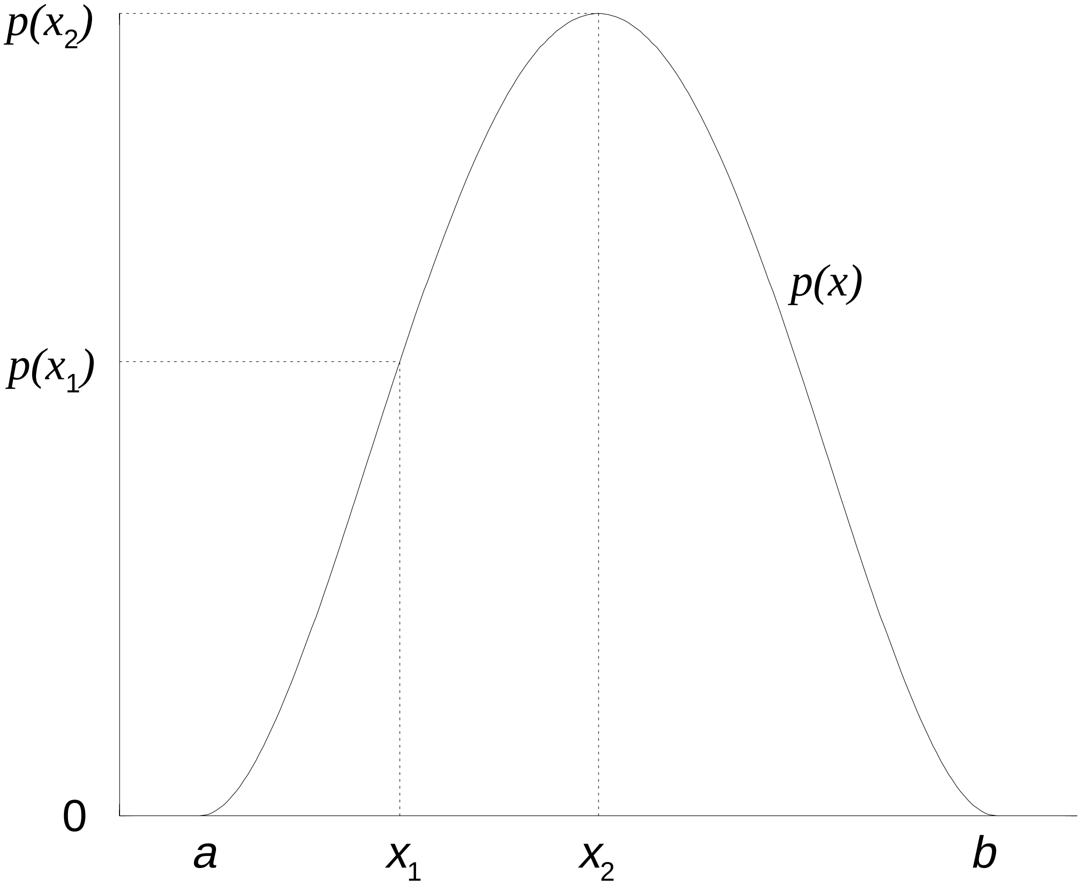

Figure 4.1: A typical probability distribution.

We now construct its cumulative probability distribution function:

$$c(x) = \int_a^x \mathrm{d}x'\; p(x') \tag{4.1}$$

and assume that it is properly normalised, *i.e.* $c(b) = 1$. The corresponding cumulative probability distribution function for our example is shown in Figure 4.2.

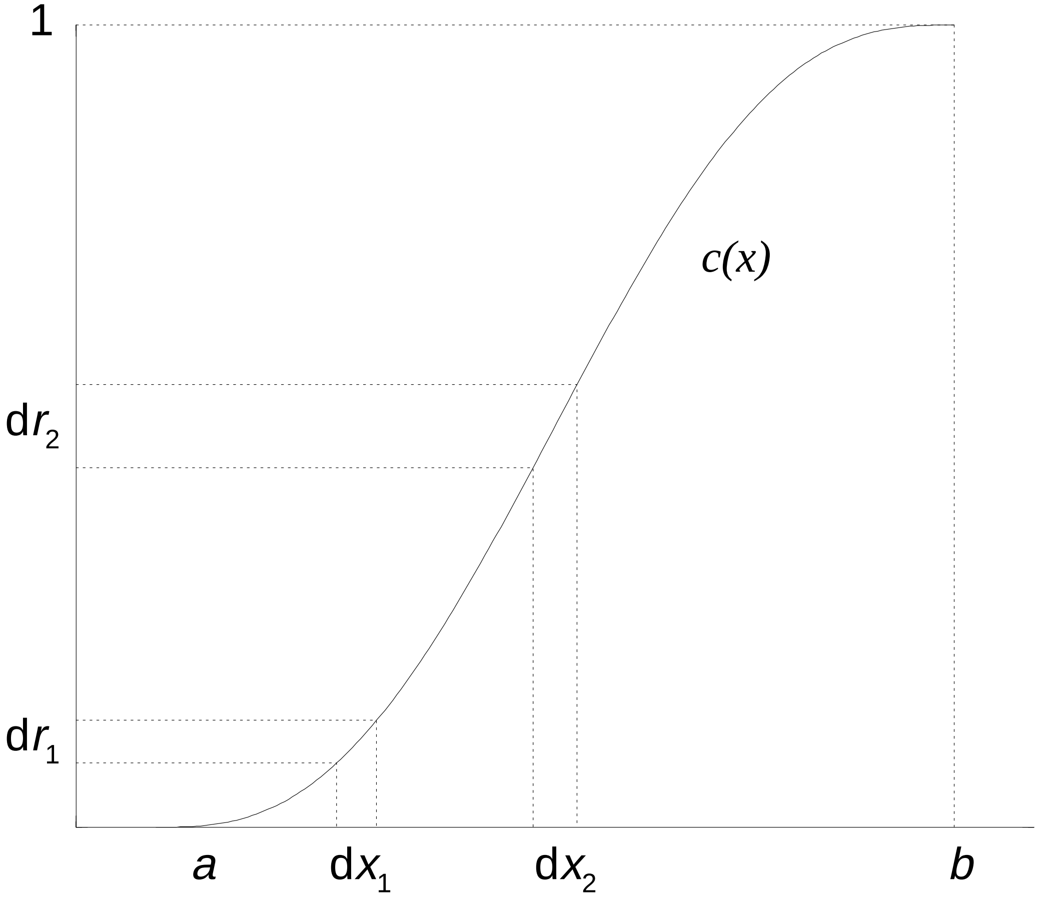

Figure 4.2: The cumulative probability distribution obtained by integrating the probability distribution function in Figure 4.1.

By its definition, we can map the cumulative probability distribution function onto the range of random variables, r, where $0 \le r \le 1$ and r is distributed uniformly. That is, $r = c(x)$.

Now consider two equally spaced intervals $\mathrm{d}x_1$ and $\mathrm{d}x_2$, differential elements in x in the vicinity of x_1 and x_2. Using some elementary calculus we see that:

$$\frac{\mathrm{d}r_1}{\mathrm{d}r_2} = \frac{(\mathrm{d}/\mathrm{d}x)c(x)|_{x=x_1}}{(\mathrm{d}/\mathrm{d}x)c(x)|_{x=x_2}} = \frac{p(x_1)}{p(x_2)} \ . \tag{4.2}$$

We can interpret this as meaning that, if we select many random variables in the range [0,1], then the number that fall within $\mathrm{d}r_1$ divided by the number that fall within $\mathrm{d}r_1$ is equal to the ratio of the probability distribution at x_1 to x_2. (Recall the interpretation of the probability distribution as given in Chapter 2.)

Having mapped the random numbers onto the cumulative probability distribution function, we may invert the equation to give:

$$x = c^{-1}(r) \ . \tag{4.3}$$

All cumulative probability distribution functions that arise from properly defined probability distribution functions are invertible, numerically if not analytically[1].

Then, by choosing r's randomly over a uniform distribution and substituting them in the above equation, we generate x's according to the proper probability distribution function.

Example:

As we will discuss in Chapter 8, the distance, z, to an interaction is governed by the well-known probability distribution function:

$$p(z)\mathrm{d}z = \mu e^{-\mu z}\mathrm{d}z \ , \tag{4.4}$$

where μ is the interaction coefficient. The valid range of z is $0 \leq z < \infty$ and this probability distribution function is already properly normalised. The corresponding cumulative probability distribution function and its random number map is given by:

$$r = c(z) = 1 - e^{-\mu z} \ . \tag{4.5}$$

Inverting gives:

$$z = -\frac{1}{\mu}\log(1-r) \ . \tag{4.6}$$

If r is uniformly distributed over $[0,1]$ then so is $1-r$. An equivalent form of the above equation (that saves one floating point operation) is:

$$z = -\frac{1}{\mu}\log(r) \ . \tag{4.7}$$

[1] However, there are subtleties. Sampling the probability distribution function is a differentiation process. Thus, if a cumulative probability distribution function is constructed numerically, differentiation leads to minor difficulties. For example, if a cumulative probability distribution function is represented by a set of linear splines, differentiation will lead to a step-wise continuous probability distribution function.

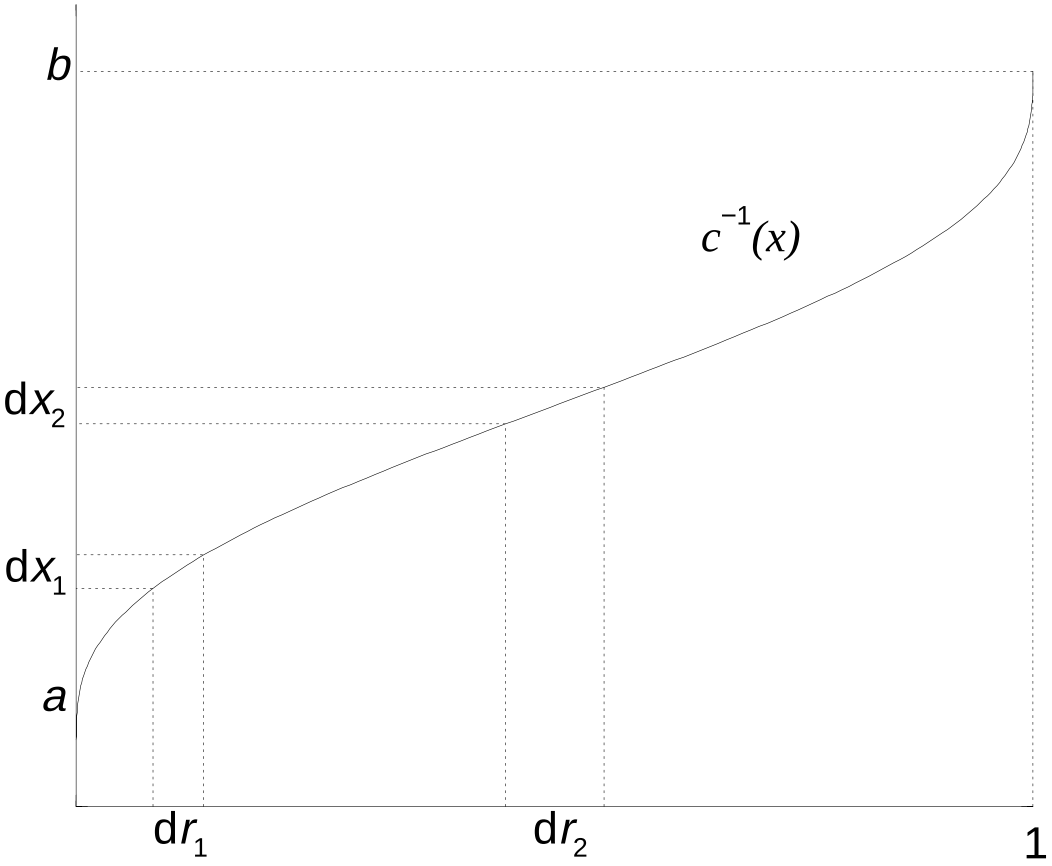

Figure 4.3: The inverse cumulative probability distribution obtained by inverting the cumulative probability distribution function in Figure 4.2.

This is exactly the form used to calculate a particle's distance to an interaction in all Monte Carlo codes. Recall, however, that if the random number generator provides an exact zero as a possibility, the sampling implied by Equation 4.7 will cause a floating-point error. (It is best to have a random number generator that does not provide exact zero's unless you really need them for something specific.)

4.2 Rejection method

While the invertible cumulative probability distribution function method is always possible, at least in principle, it is often impractical to calculate $c()^{-1}$ because it may be exceedingly complicated mathematically or contain mathematical structure that is difficult to control. Another approach is to use the rejection method.

In recipe form, the procedure is this:

1. Scale the probability distribution function by its maximum value obtaining a new distribution function, $f(x) = p(x)/p(x_{\rm max})$, which has a maximum value of 1 which occurs at $x = x_{\rm max}$ (see Figures 4.4 and 4.5). Clearly, this method works only if the probability distribution function is not infinite anywhere and if it is not prohibitively difficult to determine the location of the maximum value. If it is not possible to determine the maximum easily, then overestimating it will work as well, but less efficiently.

2. Choose a random number, r_1, uniform in the range $[0, 1]$ and use it to obtain an x which is uniform in the probability distribution function's range [a,b]. (To do this, calculate $x = a + (b - a)r_1$.) (Note: This method is restricted to finite values of a and b. However, if either a or b are infinite a suitable transformation may be found to allow one to work with a finite range. *e.g.* $x \in [a, \infty)$ may be mapped into $y \in [0, 1)$ via transformation $x = a[1 - \log(1 - y)]$.)

3. Choose a second random number r_2. If $r_2 < p(x)/p(x_{\rm max})$ (region under $p(x)/p(x_{\rm max})$ in Figure 4.5) then accept x, else, reject it (shaded region above $p(x)/p(x_{\rm max})$ in Figure 4.5) and go back to step 2.

The efficiency of the rejection technique is defined as:

$$\epsilon = \frac{1}{p(x_{\rm max})} \int_a^b {\rm d}x \; p(x) \; . \tag{4.8}$$

This is the ratio of the expected number of random numbers pairs that are accepted to the total number of pairs employed.

Remarks:

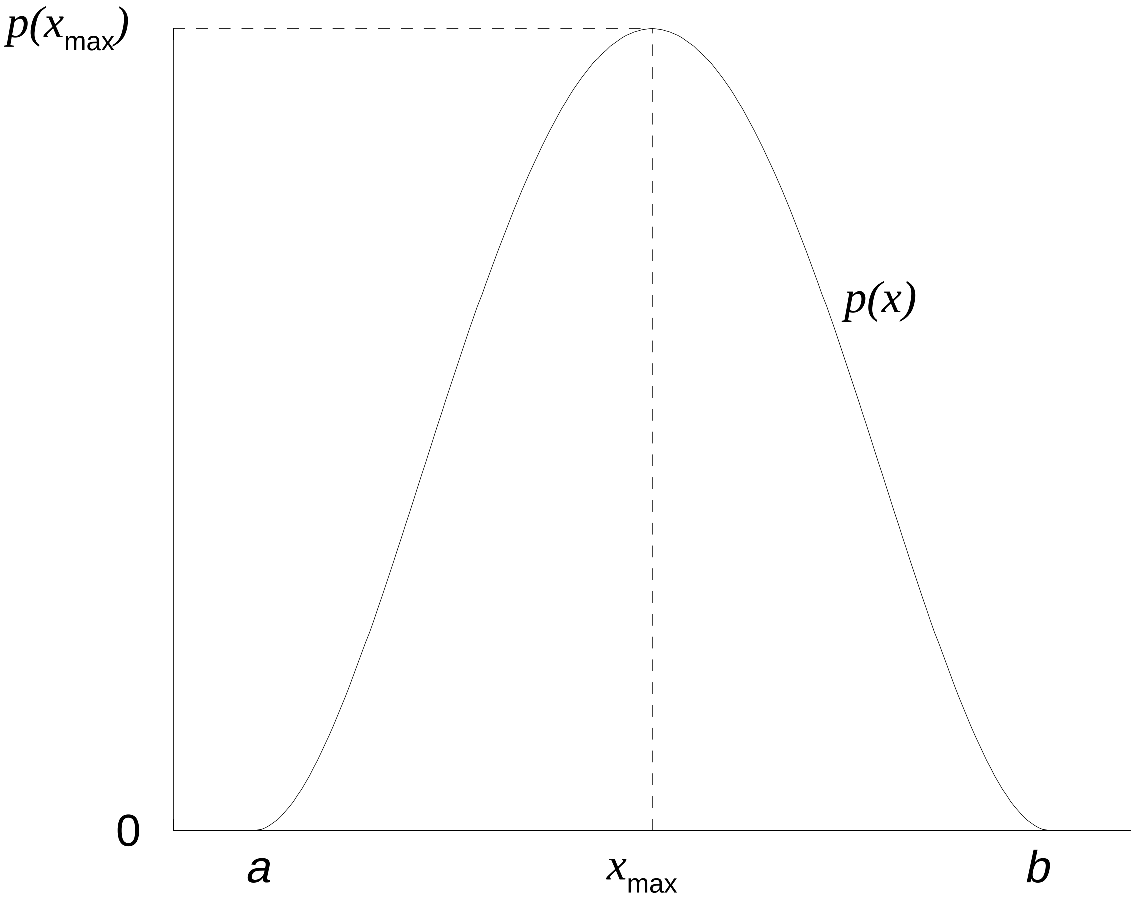

Figure 4.4: A typical probability distribution.

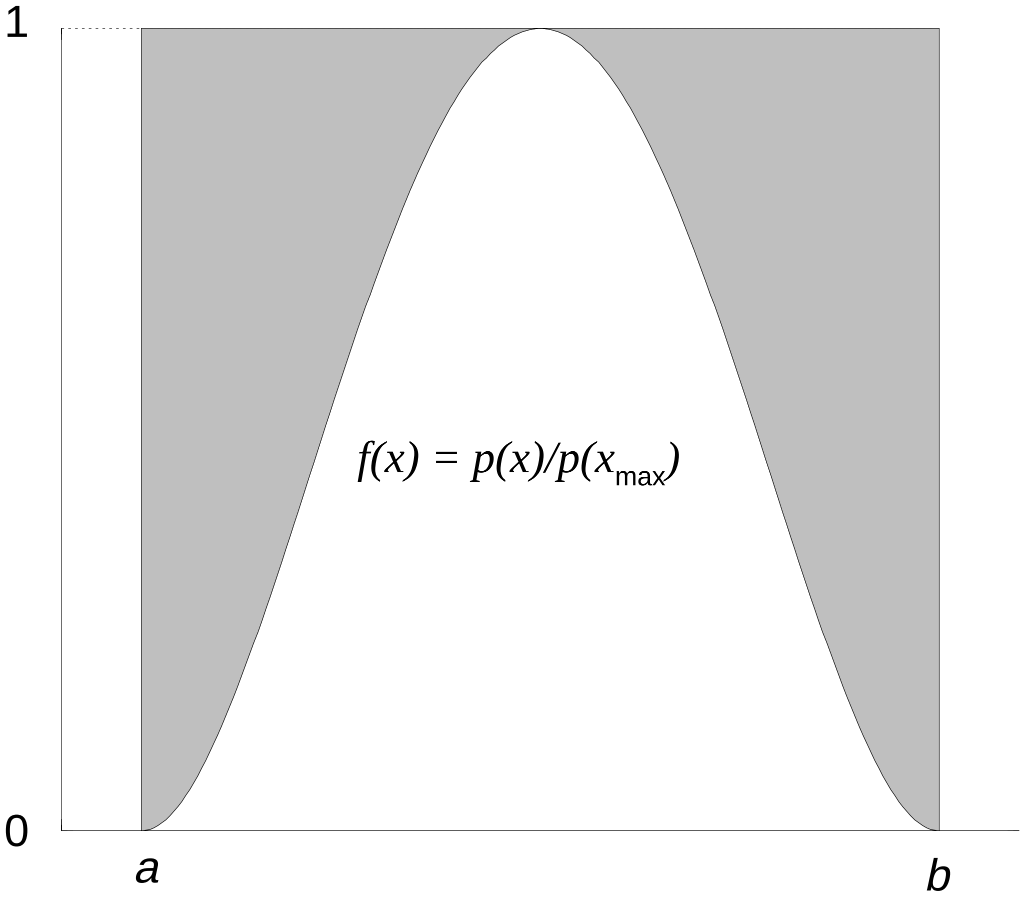

Figure 4.5: The probability distribution of Figure 4.4 scaled for the rejection technique.

This method will result in x being selected according to the probability distribution function. Some consider this method "crude" because random numbers are "wasted" unlike the invertible cumulative probability distribution function method. It is particularly wasteful for "spiky" probability distribution functions. However, it can save computing time if the $c()^{-1}$ is very complicated. One has to "waste" many random numbers to use as much computing time as in the evaluation of a transcendental function!

4.3 Mixed methods

As a final topic in elementary sampling theory we consider the "mixed method", a combination of the previous two methods.

Imagine that the probability distribution function is too difficult to integrate and invert, ruling out the direct approach without a great deal of numerical analysis, and that it is "spiky", rendering the rejection method inefficient. (Many probability distributions have this objectionable character.) However, imagine that the probability distribution function can be factored as follows:

$$p(x) = f(x)g(x) \tag{4.9}$$

where $f(x)$ is an invertible function that contains most of the "spikiness", and $g(x)$ is relatively flat but contains most of the mathematical complexity. The recipe is as follows:

1. Normalise $f(x)$ producing $\tilde{f}(x)$ such that $\int_a^b \mathrm{d}x\ \tilde{f}(x) = 1$.

2. Normalise $g(x)$ producing $\tilde{g}(x)$ such that $\tilde{g}(x) \leq 1\ \forall\ x \in [a, b]$.

3. Using the direct method described previously, choose an x using $\tilde{f}(x)$ as the probability distribution function.

4. *Using this* x, apply the rejection technique using $\tilde{g}(x)$. That is, choose a random number, r, uniformly in the range $[0, 1]$. If $\tilde{g}(x) \leq r$, accept x, otherwise go back to step 3.

Remarks:

With some effort, any mathematically complex, spiky function can be factored in this manner. The art boils down to the appropriate choice of $\tilde{f}(x)$ that leaves a $\tilde{g}(x)$ that is nearly flat. For two recent examples of this method as applied to a production-level code, see References [BR86] and [BMC89].

The mixed method is also tantamount to a change in variables. Let

$$p(x)\mathrm{d}x = f(x)g(x)\mathrm{d}x = (\tilde{f}(x)\mathrm{d}x)\left(\int_a^b \mathrm{d}x\ f(x)\right) g(x)\ , \tag{4.10}$$

where $\tilde{f}(x)$ is now a properly normalized probability distribution function. Employing $\tilde{f}(x)$ as the function for the direct part, we let

$$u = c(x) = \int_a^x \tilde{f}(x')\mathrm{d}x' \ , \tag{4.11}$$

be a transformation between x and u. Note the limits of u, $0 \leq u \leq \int_a^b \tilde{f}(x')\mathrm{d}x' = 1$. By definition, the inverse exists so that $x = c^{-1}(u)$. As well $\mathrm{d}u = f(x)\mathrm{d}x$. Thus, we can rewrite Equation 4.10 as:

$$p(x)\mathrm{d}x = \left(\int_a^b \mathrm{d}x \ f(x)\right) g(u)\mathrm{d}u \ , \tag{4.12}$$

which eliminates $f(x)$ through a change in variables. Thus, one can sample $g(u)$ using rejection (or some other technique) and relate the selected u to x through the inverse relation $x = c^{-1}(u)$.

If the rejection technique is employed for $g(x)$, then the efficiency of is calculated in the same way as in Equation 4.8.

4.4 Examples of sampling techniques

4.4.1 Circularly collimated parallel beam

The normalised probability distribution in this case is:

$$p(\rho, \phi) \ \mathrm{d}\rho \ \mathrm{d}\phi = \frac{1}{\pi\rho_0^2}\rho \ \mathrm{d}\rho \ \mathrm{d}\phi \qquad 0 \leq \rho \leq \rho_0 \quad 0 \leq \phi \leq 2\pi \tag{4.13}$$

where ρ is the cylindrical radius, ρ_0 is the collimation radius and ϕ is the azimuthal angle. $\rho\mathrm{d}\rho\mathrm{d}\phi$ is a differential surface element in cylindrical coordinates. This is a separable probability distribution of the form:

$$p(\rho, \phi) \ \mathrm{d}\rho \ \mathrm{d}\phi = \mathrm{d}p_1(\rho) \ \mathrm{d}p_2(\phi) \tag{4.14}$$

where:

$$p_1(\rho) \ \mathrm{d}\rho = \frac{2}{\rho_0^2}\rho \ \mathrm{d}\rho \qquad 0 \leq \rho \leq \rho_0 \tag{4.15}$$

and

$$p_2(\phi) \ \mathrm{d}\phi = \frac{1}{2\pi} \ \mathrm{d}\phi \qquad 0 \leq \phi \leq 2\pi \tag{4.16}$$

Direct method

The cumulative probability distribution functions in this case are:

$$c_1(\rho) = \frac{2}{\rho_0^2}\int_0^\rho \mathrm{d}\rho'\rho' = \frac{\rho^2}{\rho_0^2} \tag{4.17}$$

$$c_2(\phi) = c_2(\phi) = \frac{1}{2\pi}\int_0^\phi \mathrm{d}\phi' = \frac{\phi}{2\pi} \tag{4.18}$$

Inverting gives:

$$\rho = \rho_0\sqrt{r_1} \tag{4.19}$$

$$\phi = 2\pi r_2 \tag{4.20}$$

where the r_i are random numbers on the range $[0, 1]$.

The code segment that would produce accomplish this looks like:

```
      rho = rho_0  * sqrt(rng())
      phi = 2e0 * pi * rng()
      x = rho * cos(phi)
      y = rho * sin(phi)
```

where `rng()` is a function that return a random number uniformly on the range $[0, 1]$ [or $(0, 1]$ or $[0, 1)$ or $(0, 1)$].

Rejection method

In this technique, a point is chosen randomly within the square $-1 \leq x \leq 1; -1 \leq y \leq 1$. If this point lies within a circle with unit radius the point is accepted and the x and y values scaled by the collimation radius, ρ_0. The code segment that would accomplish this looks like:

```
1         x = 2e0 * rng() - 1e0
          y = 2e0 * rng() - 1e0
          IF (x**2 + y**2 .gt. 1e0) goto 1
      x = rho_0 * x
      y = rho_0 * y
```

Which is better?

Actually, both methods are equivalent mathematically. However, one or the other may have advantages in execution speed depending on other factors in the application. If the geometry is not cylindrically symmetric or all the scoring that is done does not make use of the inherent cylindrical symmetry, then the rejection method is about twice as fast as the direct method because the trigonometric functions are not employed in the rejection method.

If the geometry is cylindrically symmetric and the scoring takes advantage of this symmetry, then the direct method is about 2–3 times faster because symmetry reduces the calculation to:

```
x = rho_0  * sqrt(rng())
y = 0
```

Many computers now have hardware square root capabilities. With this capability the direct method may be advantageous, whether or not one makes use of the cylindrical symmetry.

4.4.2 Point source collimated to a planar circle

The normalised probability distribution in this case is:

$$p(\theta,\phi)\mathrm{d}\theta\mathrm{d}\phi = \frac{\mathrm{d}\phi}{2\pi}\frac{\sin\theta\ \mathrm{d}\theta}{1-\cos\theta_0} \qquad 0 \le \theta \le \theta_0 \quad 0 \le \phi \le 2\pi \tag{4.21}$$

where θ is the polar angle and ϕ is the azimuthal angle. $\sin\theta\ \mathrm{d}\theta\ \mathrm{d}\phi$ is a differential solid angle element in spherical coordinates. θ_0 is the collimation angle. In terms of the distance to the collimation plane z_0 and the diameter of the collimation circle on this plane ρ_0, $\cos\theta_0 = z_0/\sqrt{z_0^2+\rho_0^2}$.

This is a separable probability distribution of the form:

$$p(\theta,\phi)\mathrm{d}\theta\mathrm{d}\phi = p_1(\theta)\mathrm{d}\theta\ p_2(\phi)\mathrm{d}\phi \tag{4.22}$$

where:

$$p_1(\theta)\mathrm{d}\theta = \frac{\sin\theta\mathrm{d}\theta}{1-\cos\theta_0} \qquad 0 \le \theta \le \theta_0 \tag{4.23}$$

and

$$p_2(\phi)\mathrm{d}\phi = \frac{1}{2\pi}\mathrm{d}\phi \qquad 0 \le \phi \le 2\pi \tag{4.24}$$

The cumulative probability distribution functions in this case are:

$$c_1(\theta) = \frac{1}{1-\cos\theta_0}\int_0^\theta \sin\theta'\mathrm{d}\theta' = \frac{1-\cos\theta}{1-\cos\theta_0} \tag{4.25}$$

$$c_2(\phi) = c_2(\phi) = \frac{1}{2\pi}\int_0^{\phi} \mathrm{d}\phi' = \frac{\phi}{2\pi} \tag{4.26}$$

Inverting gives:

$$\cos\theta = 1 - r_1[1 - \cos\theta_0] \tag{4.27}$$

$$\phi = 2\pi r_2 \tag{4.28}$$

where the r_i are random numbers on the range $[0, 1]$.

The code segment that would accomplish this looks like:

```
cos_theta = 1e0 - rng() * (1e0 - cos_theta_0)
theta     = acos(theta)
sin_theta = sin(theta)

phi = 2e0 * pi * rng()

u = sin_theta * cos(phi) ! u is sin(theta)*cos(phi),
                         ! the x-axis direction cosine
v = sin_theta * sin(phi) ! v is sin(theta)*sin(phi),
                         ! the y-axis direction cosine
w = cos_theta            ! w is cos(theta),
                         ! the z-axis direction cosine

x = z_0 * u/w            ! x = z_0 * tan(theta)*cos(phi)
y = z_0 * v/w            ! y = z_0 * tan(theta)*sin(phi)
```

In terms of the cylindrical coordinates on the collimation plane, Equation 4.27 becomes:

$$\frac{z_0}{\sqrt{\rho^2 + z_0^2}} = 1 - r_1\left[1 - \frac{z_0}{\sqrt{\rho_0^2 + z_0^2}}\right] \tag{4.29}$$

which yields a value for ρ on the collimation plane.

In the small angle limit, $\theta_0 \longrightarrow 0$, the circularly collimated parallel beam result should be recovered. If one employs the small angle approximation, $\rho \ll z_0$ and $\rho_0 \ll z_0$, Equation 4.29 obtains the result of Equation 4.19, *i.e.* $\rho = \rho_0\sqrt{r_1}$.

4.4.3 Mixed method example

Consider the probability function:

$$p(x)\mathrm{d}x = Ne^{-x^2}\frac{2x\mathrm{d}x}{(1+x^2)^2} \qquad 0 \le x < \infty\ , \tag{4.30}$$

where N is the normalization factor such that $\int_0^\infty p(x)\mathrm{d}x = 1$. Although $p(x)$ is integrable analytically[2], it can not be inverted analytically. Therefore, we consider the "spiky" part that we can integrate analytically:

$$f(x)\mathrm{d}x = \frac{2x\mathrm{d}x}{(1+x^2)^2}\mathrm{d}x \qquad 0 \le x < \infty \ , \tag{4.32}$$

which can be integrated directly,

$$r = c(x) = 1 - \frac{1}{1+x^2} \ , \tag{4.33}$$

and inverted,

$$x = \sqrt{\frac{r}{1-r}} \ . \tag{4.34}$$

This is equivalent to a the change of variables,

$$u = 1 - \frac{1}{1+x^2} \qquad x = \sqrt{\frac{u}{1-u}} \ , \tag{4.35}$$

and we must now sample,

$$g(x)\mathrm{d}x = \exp\left(-\frac{u}{1-u}\right)\mathrm{d}u \qquad 0 \le u \le 1 \ . \tag{4.36}$$

If we apply rejection to $g(x)$ directly, it can be shown that the "efficiency", $\epsilon = 0.404$.

Interestingly enough, we can choose to do this example the other way! We can choose as our direct function:

$$f(x)\mathrm{d}x = 2xe^{-x^2}\mathrm{d}x \qquad 0 \le x < \infty \ , \tag{4.37}$$

which can be integrated directly,

$$r = c(x) = 1 - e^{-x^2} \ , \tag{4.38}$$

and inverted,

$$x = \sqrt{-\log(1-r)} \ . \tag{4.39}$$

[2] The cumulative probability function can be written

$$c(x) = 1 - \frac{e^{-x^2} + e\left(1+x^2\right)Ei(-1-x^2)}{\left(1+x^2\right)\left(1+e\,Ei(-1)\right)} \tag{4.31}$$

where $Ei(z)$ is the exponential integral [AS64].

This is equivalent to a the change of variables,

$$u = 1 - e^{-x^2} \qquad x = \sqrt{-\log(1-u)}\ , \tag{4.40}$$

and we must now sample,

$$g(x)\mathrm{d}x = \frac{1}{[1-\log(1-u)]^2\mathrm{d}u} \qquad 0 \le u \le 1\ . \tag{4.41}$$

This approach has the same efficiency as the previous approach. However, it is more costly because is involves the use of more transcendental functions.

4.4.4 Multi-dimensional example

Consider the joint probability function:

$$p(x,y)\ \mathrm{d}x\ \mathrm{d}y = (x+y)\ \mathrm{d}x\ \mathrm{d}y \qquad 0 \le x, y \le 1\ . \tag{4.42}$$

The marginal probability in x is:

$$m(x) = \int_0^1 \mathrm{d}y\ (x+y) = x + \frac{1}{2}\ , \tag{4.43}$$

the conditional probability of y given x is:

$$p(y|x) = \frac{p(x,y)}{m(x)} = \frac{x+y}{x+\frac{1}{2}}\ , \tag{4.44}$$

so that

$$p(x,y) = m(x)p(y|x)\ . \tag{4.45}$$

First we sample the marginal probability distribution in x. The cumulative distribution function and its associated random number map is:

$$r_1 = c(x) = \int_0^x \mathrm{d}x' \left(x' + \frac{1}{2}\right) = \frac{x^2}{2} + \frac{x}{2}\ , \tag{4.46}$$

which is a quadratic relation that can be inverted to give:

$$x = \frac{-1+\sqrt{1+8r_1}}{2}\ . \tag{4.47}$$

The choice of the plus sign in the inversion of the quadratic relation was made based on the having $x = 0$ when $r_1 = 0$ and $x = 1$ when $r_1 = 1$.

Now that x is determined, we form the conditional cumulative probability distribution, $c(y|x)$, and its associated random number mapping:

$$r_2 = c(y|x) = \int_0^y \mathrm{d}y \; p(y|x) = \frac{y^2 + 2xy}{2x + 1} \; , \tag{4.48}$$

which itself can be inverted using quadratic inversion:

$$y = -x + \sqrt{x^2 + r_2(2x + 1)} \; , \tag{4.49}$$

which again involved a choice of sign based upon the expected limits, $y = 0$ when $r_2 = 0$ and $y = 1$ when $r_2 = 1$. For intermediate values of r_2, y depends upon the choice of x.

Bibliography

[AS64] M. Abramowitz and I. A. Stegun, editors. *Handbook of mathematical functions with formulae, graphs and mathematical tables.* Number 55 in Applied Mathematics. National Bureau of Standards, Washington, D.C., 1964.

[BMC89] A. F. Bielajew, R. Mohan, and C. S. Chui. Improved bremsstrahlung photon angular sampling in the EGS4 code system. *National Research Council of Canada Report PIRS-0203*, 1989.

[BR86] A. F. Bielajew and D. W. O. Rogers. Photoelectron angular distribution in the EGS4 code system. *National Research Council of Canada Report PIRS-0058*, 1986.

Problems

The following problems are to be solved on paper. Computer verification of the sampling methods developed is not necessary (but would be neat to try.)

1. Form the cumulative probability distribution for the Cauchy distribution:
$$p(x) = \frac{1}{\pi}\frac{1}{1+x^2} \qquad -\infty < x < \infty. \tag{4.50}$$
Invert it to indicate how x would be determined from a random number.

2. Form the cumulative probability distribution for the small angle form of the Rutherfordian distribution:
$$p(x) = \frac{2x}{(x^2+1)^2} \qquad 0 \le x < \infty, \tag{4.51}$$
Invert it to indicate how x would be determined from a random number.

3. Normalize
$$f(x,y) = \sin(x+y) \qquad 0 \le x < \frac{\pi}{2}, 0 \le y < \frac{\pi}{2} \tag{4.52}$$
converting it to a probability distribution. Develop a sampling technique by forming a marginal and a conditional probability distribution. *(Hint: You will probably have to use 4 random numbers to sample this probability distribution.)*

Chapter 5

Error estimation

In this chapter we consider the question of evaluating the results of one's Monte Carlo calculation. Without proper evaluation of the results, the numbers are *meaningless.* Without a doubt, the development of this evaluation process is one of the most immature in Monte Carlo and it is something that is often neglected by Monte Carlo practitioners, often to the chagrin of journal Referees!

Let us imagine that we are tallying[1] something called $T(x)$ over some range of x, say, $x_0 \leq \ x \leq X$. For concreteness, imagine that $T(x)\ \mathrm{d}x$ is the pathlength (or tracklength) distribution of particles in some volume region of space for a differential spread of $\mathrm{d}x$ centered around x. Pathlength (or tracklength) distributions are central to particle transport problems as one interpretation of fluence (and a very practical one from the standpoint of Monte Carlo calculation) is pathlength per volume [Chi78]. In reality, however, our computer tallies would have to be discretized. That is, we would have to set up a computer "mesh" or "grid" in x, say, $(x_0, x_1, x_2 \cdots x_{N-1}, x_N = X)$ with N tallying "bins" numbered $1, 2 \cdots N$ and $N+1$ tally "mesh-points" bounded by $N+1$ tally "mesh-points" $(x_0, x_1, x_2 \cdots x_{N-1}, x_N = X)$.

Note that the choice of the mesh points assumes that the Monte Carlo practitioner knows something about the nature of the tally, both its endpoints, x_0 and X and a general idea of the shape of $T(x)$. Ideally one would like to arrange that the tallying bins are populated in an equiprobable way! Generally, this knowledge is not known *a priori* and is determined in an iterative fashion.

Let us consider our photon pathlength distribution example in a model where there is no scattering, just a volume in space where we wish to tally the pathlength distribution from some source of photons. In this case, we could safely assume that $x_0 = 0$ corresponding

[1] "Tallying" is Monte Carlo jargon for "measuring" something, as one would do in an experiment. A tally is that which is measured. There are many similarities between the handling of measured and tallied data, except that the tallies in Monte Carlo can be unambiguous, not obfuscated by extraneous physical detail. Monte Carlo analysis also allows the deeper investigation into the statistical nature of the tally, something that experiment is often not able to accomplish.

to a grazing trajectory of a photon on the volume of interest and $x_N = x_{\rm max}$ where $x_{\rm max}$ corresponds to the maximum chord length in the volume. The details of the shape of $T(x)$ could be determined from the geometry and orientation of the volume with respect to the source. Indeed, it could be determined analytically and sometimes quite simply without resorting to Monte Carlo—depending on the complexity of the geometry.

Now let us consider a more interesting problem, one in which we have scattering within the volume of interest. There would be a finite probability that a photon could enter the volume and scatter many times producing very long x's compared to $x_{\rm max}$, the maximum chord length in the volume! The "safe" choice is $x_N = \infty$. How do we choose the mesh points in a situation where $x_N = \infty$? This is where the "trial-and-error" process enters. You would set up your mesh points $(0, x_1, x_2 \cdots x_{N-1}, \infty)$ and the last bin $x_{N-1} \longrightarrow \infty)$ would "catch" all the long pathlength contributions to the pathlength distribution. The subtlety here would then be choosing x_{N-1} large enough so that the preceding points $x_{N-1}, x_{N-2}, x_{N-3} \cdots$ are sufficient to characterize the asymptotic nature of the tally $T(x)$ in the large x regime. Why is this important? This is important for the accurate calculation of the moments of the distribution, assuming that the moments do, in fact, exist. We will discuss this more below.

Characterization of the asymptotic nature of the tally $T(x)$ in the large x regime would give indication whether or not the moments exist and will determine what sort of statistical analyses we perform.

Let us say that we are interested in the total pathlength distribution, T, in the volume per source particle. If we knew what the pathlength distribution function was, the total pathlength would correspond to:

$$T = \int_{x_0}^{x_{\rm max}} {\rm d}x \; T(x) \; . \tag{5.1}$$

However, this underlying distribution is unknown to us, so we do a simulation. We would execute a loop $i = 1, 2, 3 \cdots N_{\rm h}$ over $N_{\rm h}$ Monte Carlo histories and accumulate the pathlength in an accumulator called t_n^i. That is, if the particle's pathlength in the volume corresponding to the $i^{\rm th}$ history falls within the range x_{n-1} and x_n we accumulate x in t_n^i. Now we define

$$T_n = \frac{1}{N_{\rm h}} \sum_{i=1}^{N_{\rm h}} t_n^i \; , \tag{5.2}$$

which provides an estimate of the pathlength distribution as follows:

$$T_n \approx \int_{x_{n-1}}^{x_n} {\rm d}x \; T(x) \; . \tag{5.3}$$

Now we define

$$T^i = \sum_{n=1}^{N} t_n^i \; , \tag{5.4}$$

which is an estimate of the pathlength distribution from the $i^{\rm th}$ history, summed over all pathlength contributions. Now we can estimate the total pathlength distribution

$$T \approx \tilde{T} = \frac{1}{N_{\rm h}} \sum_{i=1}^{N_{\rm h}} T^i \ , \tag{5.5}$$

How good is our estimate for T? Fortunately, there are some fundamental mathematical proofs that come to the aid of our analysis.

First of all, we realize that our interpretation for $T(x)\ {\rm d}x$ as the pathlength distribution of particles in some volume region of space for a differential spread of ${\rm d}x$ centered around x can be rewritten as $T(x)\ {\rm d}x = xp(x)\ {\rm d}x$ where $p(x)$ is a probability distribution associated with the observation of the pathlength x. Moreover, the estimate expressed in Equation 5.5 is an estimate of the *mean* of the distribution associated with the probability distribution.

First of all, we assume that $p(x)$ is a properly defined probability distribution. It may not be in which case our estimate expressed in Equation 5.5 is *meaningless.* There are examples of a tallies with no mean, such as the fluence distribution in time of neutrons in a super-critical nuclear reactor but we will not consider them in this course.

Let us assume that the mean of the distribution does indeed exist. We give it a special name μ defined as follows:

$$\mu = \int_{x_0}^{x_{\rm max}} {\rm d}x\ T(x) = \int_{x_0}^{x_{\rm max}} {\rm d}x\ xp(x)\ . \tag{5.6}$$

The "law of large numbers" proven by Khintchine in 1929 [Khi29] states:

$$P\left\{ \left| \frac{T^1 + T^2 + \cdots + T^{N_{\rm h}}}{N_{\rm h}} \right| - \mu > \epsilon \right\} \longrightarrow 0\ , \tag{5.7}$$

in the limit of large $N_{\rm h}$, where ϵ is a vanishingly small. There is no assumption in Khintchine's proof about the existence of any higher moments other than the first-order one which yields the mean. It should be noted that this is a limiting theorem, one that does not guarantee that the mean is approached "nicely". It may fluctuate wildly for finite values of $N_{\rm h}$ and the practical use of Khintchine theorem, insofar as Monte Carlo applications are concerned is the verification of the intuitive concept, *"If you compute long enough, you should converge to the expected mean."*

Stronger and more meaningful statements about the rate of convergence can be made and they depend upon the existence of the variance which we may write as:

$$\sigma^2 = \left(\left[\int_{x_0}^{x_{\rm max}} {\rm d}x\ x^2 p(x) \right] - \mu \right)^2 = \left(\left[\int_{x_0}^{x_{\rm max}} {\rm d}x\ xT(x) \right] - \mu \right)^2\ . \tag{5.8}$$

If the variance exists, then the Central limit Theorem [Lin22, Fel67] states:

$$P\left(\frac{\tilde{T} - \mu}{\sigma/\sqrt{N_{\rm h}}} < \beta \right) \longrightarrow \Re(\beta) \tag{5.9}$$

in the limit of large N_h, where $\Re(\beta)$ is the normal distribution. That is, $\tilde{T}$ follows a normal distribution centered at μ with a Gaussian width $\sigma/\sqrt{N_\mathrm{h}}$ that narrows as *increases*! In other words, there is a 0.67 chance that $\tilde{T} \pm \sigma/\sqrt{N_\mathrm{h}}$ will include μ and a probability of 0.95 that $\tilde{T} \pm 2\sigma/\sqrt{N_\mathrm{h}}$ will include μ. Note that this makes no assumptions about the shape of the probability distribution $p(x)$, only that its second moment exists. Irrespective of the shape of $p(x)$, the estimate of its mean will be normally (Gaussian) distributed and that the width of this distribution narrows with increased sampling. It is truly one of the most remarkable results in mathematics!

We remark that it is an asymptotic theorem, one that only applies in the limit, $N_\mathrm{h} \to \infty$. How large N_h has to be depends on the details of the simulation and the characteristics of the underlying distribution function. However, Kalos and Whitlock [KW86] suggest that if one obtains the third moment of $p(x)$:

$$\langle x^3 \rangle = \int_{x_0}^{x_\mathrm{max}} \mathrm{d}x\ x^3 p(x)\ , \tag{5.10}$$

then the condition

$$N_\mathrm{h} \gg \sigma^6/\langle x^3 \rangle^2\ , \tag{5.11}$$

should be sufficient.

This section ends with a warning. The rest of this chapter continues with the standard cookbook recipes for estimating errors. Many Monte Carlo practitioners quote these numbers without investigating the characteristics of the underlying distributions. While this has become standard practice, be aware that these conclusions are based on the distributions having certain well-behaved characteristics. Sometimes ignoring this can lead you to faulty conclusions.

5.1 Direct error estimation

Assume that x is a quantity we calculate during the course of a Monte Carlo simulation, *i.e.* a scoring variable or simply a “score” or a “tally”. The output of a Monte Carlo calculation is usually useless unless we can ascribe a probable error to it.

The conventional approach to calculating the estimated error is as follows:

- Assume that the calculation calls for the simulation of N particle histories.
- Assign and accumulate the value x_i for the score associated with the i’th history, where $1 \le i \le N$. Assign as well the square of the score x_i^2 for the i’th history.
- Calculate the mean value of x:

$$\overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{5.12}$$

- Estimate the variance associated with the distribution of the x_i:

$$s_x^2 = \frac{1}{N-1}\sum_{i=1}^{n}(x_i - \overline{x})^2 = \frac{1}{N-1}\sum_{i=1}^{n}(x_i^2 - \overline{x}^2) \qquad (5.13)$$

- The estimated variance of $\overline{x}$ is the standard variance of the mean:

$$s_{\overline{x}}^2 = \frac{s_x^2}{N} \qquad (5.14)$$

  It is the error in $\overline{x}$ we are seeking, not the "spread" of the distribution of the x_i.

- Report the final result as $x = \overline{x} \pm s_{\overline{x}}$.

Remarks:

The true mean and variances are not available to us, however. We must estimate them. The estimated mean $\overline{x}$ calculated in eq. 5.12 is an estimate for the true mean μ and the estimated variance s_x^2 calculated in eq. 5.13 is an estimate for the true variance σ_x^2. The appearance of the $N-1$ in the denominator in the expression for the estimated variance for the mean implicit in eq. 5.14 is often introduced as a "degrees of freedom" arguments. However, it can be derived by considering the difference between "sample" variance and its relation to the true variance. A derivation is given at the end of this chapter.

5.2 Batch statistics error estimation

In many cases, the estimation of means and variances using the methods described in the previous section is not feasible. A score may be a complicated object or there may be many geometrical volume elements to consider. It may require a lot of effort to stop after each history and compute the x_i's and x_i^2's. The "direct error estimation" should be considered first and if it is not feasible, then an alternative that is nearly as good is "batch statistics error estimation". We present a "cook book" recipe as before.

- Split the N histories into n statistical batches of N/n histories each. Note that n must be "large enough" as discussed earlier. A standard choice is $n = 30$. The accumulated quantity for each of these batches is called $x_j = \sum_{i=1}^{n} x_i$ for the j'th statistical batch.

- Calculate the mean value of x:

$$\overline{x} = \frac{1}{N}\sum_{j=1}^{n} x_j \qquad (5.15)$$

- Estimate the variance associated with the distribution of the x_i:

$$s_x^2 = \frac{1}{n-1}\sum_{j=1}^{n}(x_j - \overline{x})^2 = \frac{1}{n-1}\sum_{j=1}^{n}(x_j^2 - \overline{x}^2) \qquad (5.16)$$

- The estimated variance of $\overline{x}$ is the standard variance of the mean:

$$s_{\overline{x}}^2 = \frac{s_x^2}{n} \tag{5.17}$$

- Report the final result as $x = \overline{x} \pm s_{\overline{x}}$.

Remarks:

We use eqs. 5.15–5.17 with n fixed (at say, 30) because it gives a reasonable estimate of the error in $\overline{x}$. Any large number will do, as long as we are within the range of applicability of the Central limit theorem. In reality, decisions based upon the error in $\overline{x}$ are usually subjective in nature. There is some evidence that the calculated statistic depends weakly on the choice of n. Therefore, it is important to report how your statistics were done when you publish your Monte Carlo results.

5.3 Combining errors of independent runs

For m independent Monte Carlo runs, it is easy to derive the following relation:

$$\overline{x} = \sum_{k=1}^{m} \left(\frac{N_k}{N}\right) \overline{x}_k \tag{5.18}$$

where $\overline{x}_k$ is the value of $\overline{x}$ for the k^{th} run and N_k is the number of histories in the k^{th} run. The total number of histories is given by:

$$N = \sum_{k=1}^{m} N_k \tag{5.19}$$

Then, assuming 1st-order propagation of independent errors, it is also easy to derive:

$$s_{\overline{x}}^2 = \sum_{k=1}^{m} \left(\frac{N_k}{N}\right)^2 s_{\overline{x}_k}^2 \tag{5.20}$$

where $s_{\overline{x}_k}^2$ is the estimated variance in $\overline{x}_k$.

Example: For m = 2:

$$\overline{x} = \left(\frac{N_1}{N}\right) \overline{x}_1 + \left(\frac{N_2}{N}\right) \overline{x}_2 \tag{5.21}$$

$$N = N_1 + N_2 \tag{5.22}$$

$$s_{\overline{x}} = \sqrt{\left(\frac{N_1}{N}\right)^2 s_{\overline{x}_1}^2 + \left(\frac{N_2}{N}\right)^2 s_{\overline{x}_2}^2} \tag{5.23}$$

Remarks:

This method of combining errors effectively increases the value of n, the number of statistical batches used in the calculation. In view of the fact that the calculated statistics are thought to depend weakly on n, it is preferable (but only marginally so for the sake of consistency) to combine the x_i's (the raw data) into the standard number of statistical batches. This is easy to do by initialising the data arrays to the results of the previous run before the start of a new run.

5.4 Error estimation for binary scoring

In the special case where the scoring quantity is binary (*e.g.* a particle enters a detector's sensitive region or not, *e.g.* a particle backscatters or not) there is a significant simplification. Consider N events where the scoring variable x_i is either 0 or 1. The mean is calculated according to eq. 5.12. Starting from eqs. 5.12–5.14 it is easy to show that:

$$s^2_{\overline{x}} = \frac{\overline{x}(1-\overline{x})}{N-1}. \tag{5.24}$$

Hence, knowing the mean value gives one knowledge of its estimated variance! One is not required to gather these events in statistical bins.

Combining the results of independent runs is particularly simple. The mean is calculated from eq. 5.12 but the use of the variance combination relation, eq. 5.20, is inaccurate unless the N_j are all "large" ($N_j \gg 1$). It is best to resort to eq. 5.24 which has no such restriction.

5.5 Relationships between S^2_x and s^2_x, $S^2_{\overline{x}}$ and $s^2_{\overline{x}}$

Consider N independent measurements of x, $x_i : i = 1, 2, \cdots N$. The x_i are distributed according to some parent distribution with a mean μ and a variance σ^2. The estimated mean $\overline{x}$ and the estimated variance s^2_x are approximations (lower-case letters) that are made from a finte data set for the true mean μ and the true variance σ^2. The "sample" mean, $\overline{X}$ and the "sample" variance are related to the estimate. What is the connection?

The sample mean and the estimated mean are the same

$$\overline{X} = \overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \approx \mu \ . \tag{5.25}$$

Since the x_i are independent, their errors combine in quadrature:

$$s^2_{\overline{x}} = \sum_{i=1}^{N}\left(\frac{\partial\overline{x}}{\partial x_i}\right)^2 \sigma^2 \approx \sum_{i=1}^{N}\left(\frac{\partial\overline{x}}{\partial x_i}\right)^2 s^2_x \ . \tag{5.26}$$

From Equation 5.25

$$\frac{\partial \overline{x}}{\partial x_j} = \frac{1}{N}\sum_{i=1}^{N}\frac{\partial x_i}{\partial x_j} = \frac{1}{N}\sum_{i=1}^{N}\delta_{ij} = 1 \ . \tag{5.27}$$

So,

$$s_{\overline{x}}^2 = \frac{1}{N^2}\sum_{i=1}^{N}\sigma^2 = \frac{1}{N}\sigma^2 \approx \frac{1}{N}s_x^2 \ , \tag{5.28}$$

which relates the estimated error of the mean to the estimated variance of the population.

Another interpretation for $s_{\overline{x}}^2$ can be obtained as follows. Suppose we have M different estimates of $\overline{x}$. Then

$$s_{\overline{x}}^2 = \frac{1}{M}\sum_{m=1}^{M}(\overline{x}_m - \mu)^2 \ . \tag{5.29}$$

In particular, for $M = 1$,

$$s_{\overline{x}}^2 = (\overline{x} - \mu)^2 \ . \tag{5.30}$$

What is the relationship between the "sample" variance S_x^2 and the estimate for σ^2, s_x^2?

The sample variance is:

$$S_x^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^2 = \frac{1}{N}(\sum_{i=1}^{N}x_i^2) - \overline{x}^2 \ , \tag{5.31}$$

while the estimated variance is:

$$s_x^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2 = \frac{1}{N}(\sum_{i=1}^{N}x_i^2) - 2\mu\overline{x} + \mu^2 \ . \tag{5.32}$$

Thus,

$$s_x^2 - S_x^2 = \overline{x}^2 - 2\mu\overline{x} + \mu^2 = (\overline{x} - \mu)^2 = s_{\overline{x}}^2 = \frac{s_x^2}{N} \ . \tag{5.33}$$

Or,

$$s_x^2 = \frac{N}{N-1}S_x^2 \ . \tag{5.34}$$

Hence, the estimate variance, s_x^2, is obtained from the sample variance variance, S_x^2, by a $N/(N-1)$ factor.

Thus the estimated variance is:

$$s_x^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2 \ , \tag{5.35}$$

and the estimated variance of the mean is:

$$s_{\overline{x}}^2 = \frac{1}{N(N-1)}\sum_{i=1}^{N}(x_i - \overline{x})^2 \ . \tag{5.36}$$

Bibliography

[Chi78] A. B. Chilton. A note on the fluence concept. *Health Physics*, 34:715 – 716, 1978.

[Fel67] W. Feller. *An introduction to probability theory and its applications, Volume I, 3rd Edition.* Wiley, New York, 1967.

[Khi29] A. Khintchine. In *Comptes rendus de 'lAcadè mie des Sciences*, volume 189, pages 477 – 479. L'académie des Sciences, Paris, 1929.

[KW86] M. H. Kalos and P. A. Whitlock. *Monte Carlo methods, Volume I: Basics.* John Wiley and Sons, New York, 1986.

[Lin22] J. W. Lindeberg. Eine neue Herleitung des Exponentialgesetzes in der Wahrscheinlichkeitrechnung. *Mathematische Zeitschrift*, 15:211 – 225, 1922.

Problems

Using a computer, operating system, computing language and compiler of your choice (tell me what you used):

1. Write a code to sample the Cauchy probability distribution:

$$p(x) = \frac{1}{\pi}\frac{1}{1+x^2} \qquad -\infty < x < \infty. \tag{5.37}$$

   As a function of the number of histories, $N_{\rm h}$, over the range $1 \le N_{\rm h} \le 10,000$, plot $\overline{x}$ and $s^2_{\overline{x}}$. Discuss.

2. Write a code to sample the small angle form of the Rutherfordian probability distribution:

$$p(x) = \frac{2x}{(x^2+1)^2} \qquad 0 \le x < \infty, \tag{5.38}$$

   As a function of the number of histories, $N_{\rm h}$, over the range $1 \le N_{\rm h} \le 10,000$, plot $\overline{x}$ and $s^2_{\overline{x}}$. Discuss.

3. Write a code to sample the probability distribution:

$$p(x) = \frac{4x}{(x^2+1)^3} \qquad 0 \leq x < \infty, \tag{5.39}$$

As a function of the number of histories, N_{h}, over the range $1 \leq N_{\mathrm{h}} \leq 10,000$, plot $\overline{x}$ and $s^2_{\overline{x}}$. Discuss.

4. Write a code to sample the probability distribution:

$$p(x) = e^{-x} \qquad 0 \leq x < \infty, \tag{5.40}$$

As a function of the number of histories, N_{h}, over the range $1 \leq N_{\mathrm{h}} \leq 10,000$, plot $\overline{x}$ and $s^2_{\overline{x}}$. Discuss.

Chapter 6

Oddities: Random number and precision problems

Now that we understand about random number generators, sampling and error estimation, it is time for a brief respite to consider some of the oddities one might encounter during Monte Carlo calculations. These oddities are related to artefacts associated with random number generation and machine precision.

6.1 Random number artefacts

Consider the determination of the value of π one obtains by throwing random "darts" at a circle inscribed within a square. This is depicted in Figure 6.1. The ratio of the number of darts within the circle to the total number of darts within the square should be $\pi/4$.

For a small number of iterations, the result converges as expected. This is shown in Figure 6.2 where the ratio $4N_{\rm in}/(N\pi)$ is plotted up to 10^4 cycles along with the 1-σ error bars predicted by the "binary statistics" method. The estimated mean goes over and under the theoretical prediction, takes an excursion in the overprediction direction and eventually begins to settle down.

However, some difficulties are evident for large cycles as shown in Figures 6.3. and 6.4

There are some classic signals indicated in Figure 6.3 that the random number generator is cycling. The first piece of evidence is that the result exhibits a periodic structure. The random number generator employed in this study is a multiplicative congruential random number generator (MCRNG) with a sequence length of $2^{30} = 1,073,741,824$. Since two random numbers are employed, the periodic structure occurs over a period of $2^{29} = 536,870,912$. Another curious anomaly is that the result is close to unity (actually to within a few parts

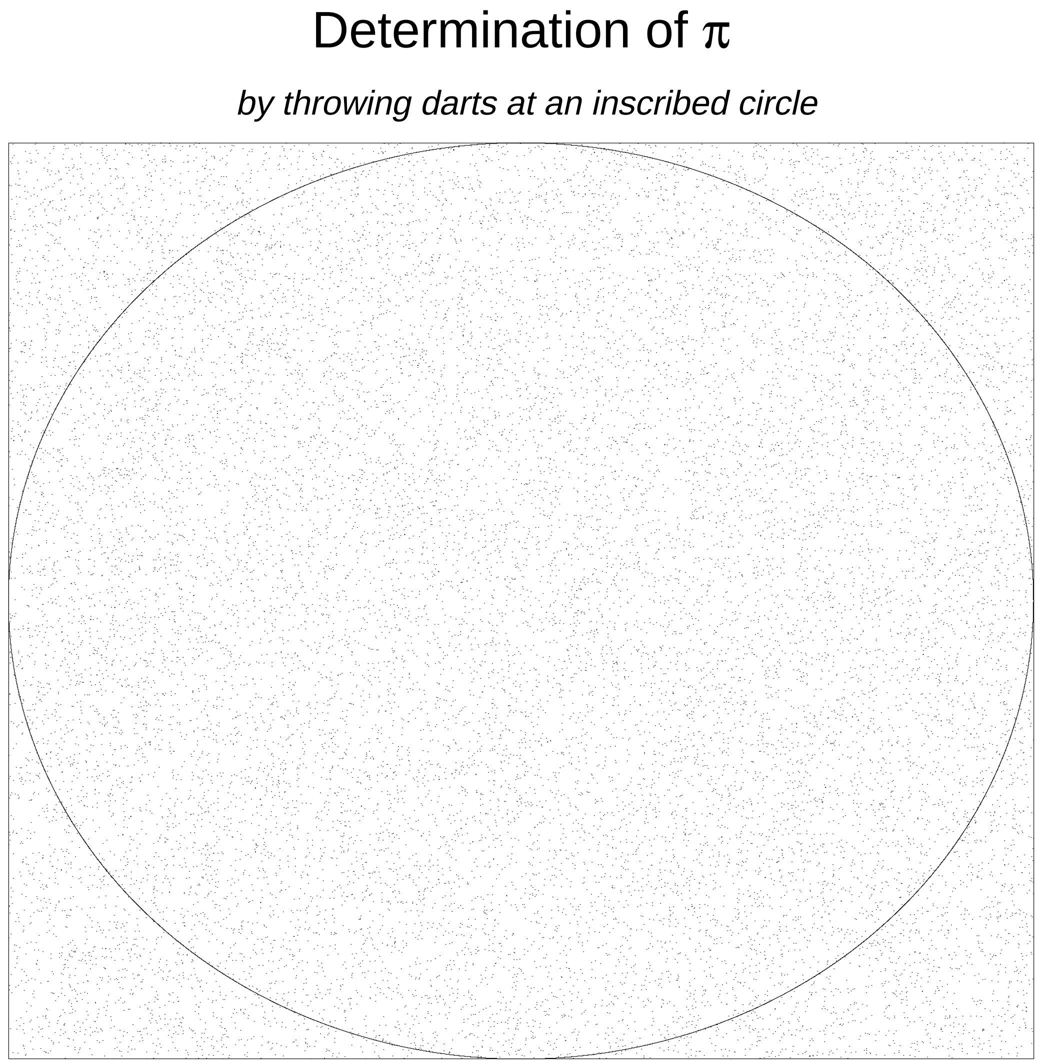

Figure 6.1: Random darts are thrown at a square with an inscribed circle. The ratio of the number of darts within the circle to the total number of darts within the square should be $\pi/4$.

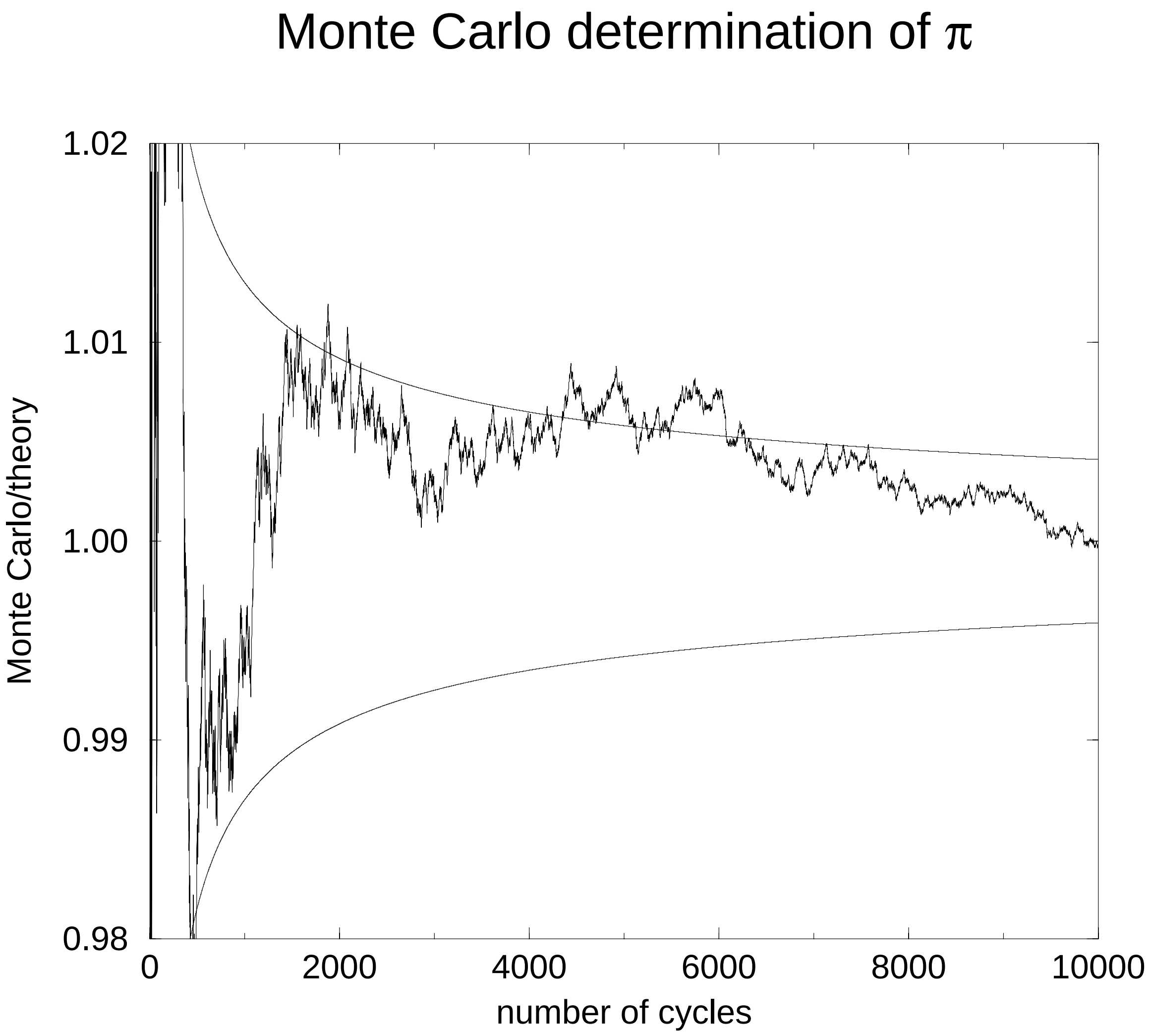

Figure 6.2: Random darts are thrown at a square with an inscribed circle. The Monte Carlo prediction divided by the theoretical prediction with the associated $1 \pm \sigma$ prediction.

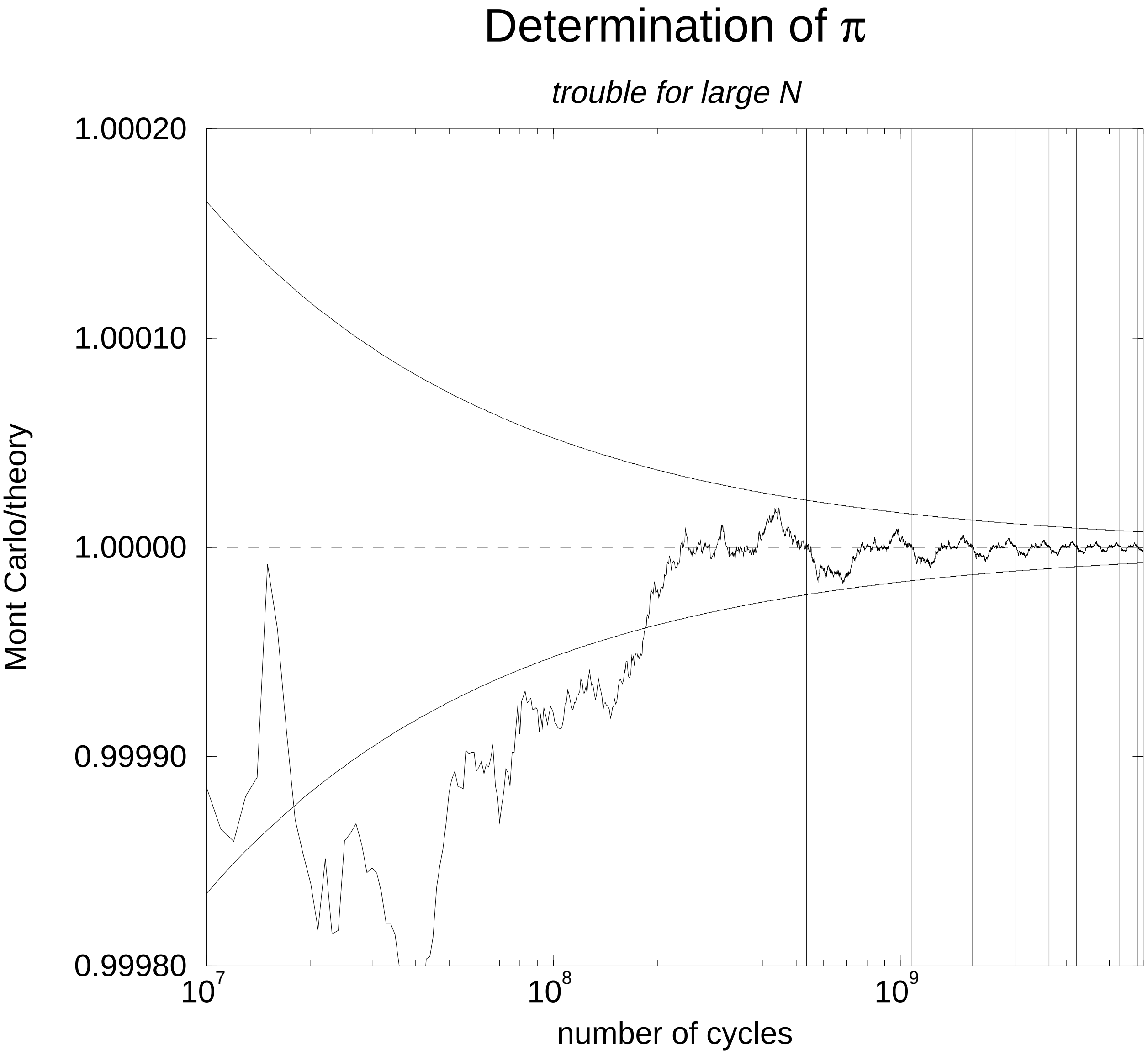

Figure 6.3: Random darts are thrown at a square with an inscribed circle. Large cycle behaviour of the Monte Carlo π experiment. The vertical lines are drawn where the random number generator begins a new cycle.

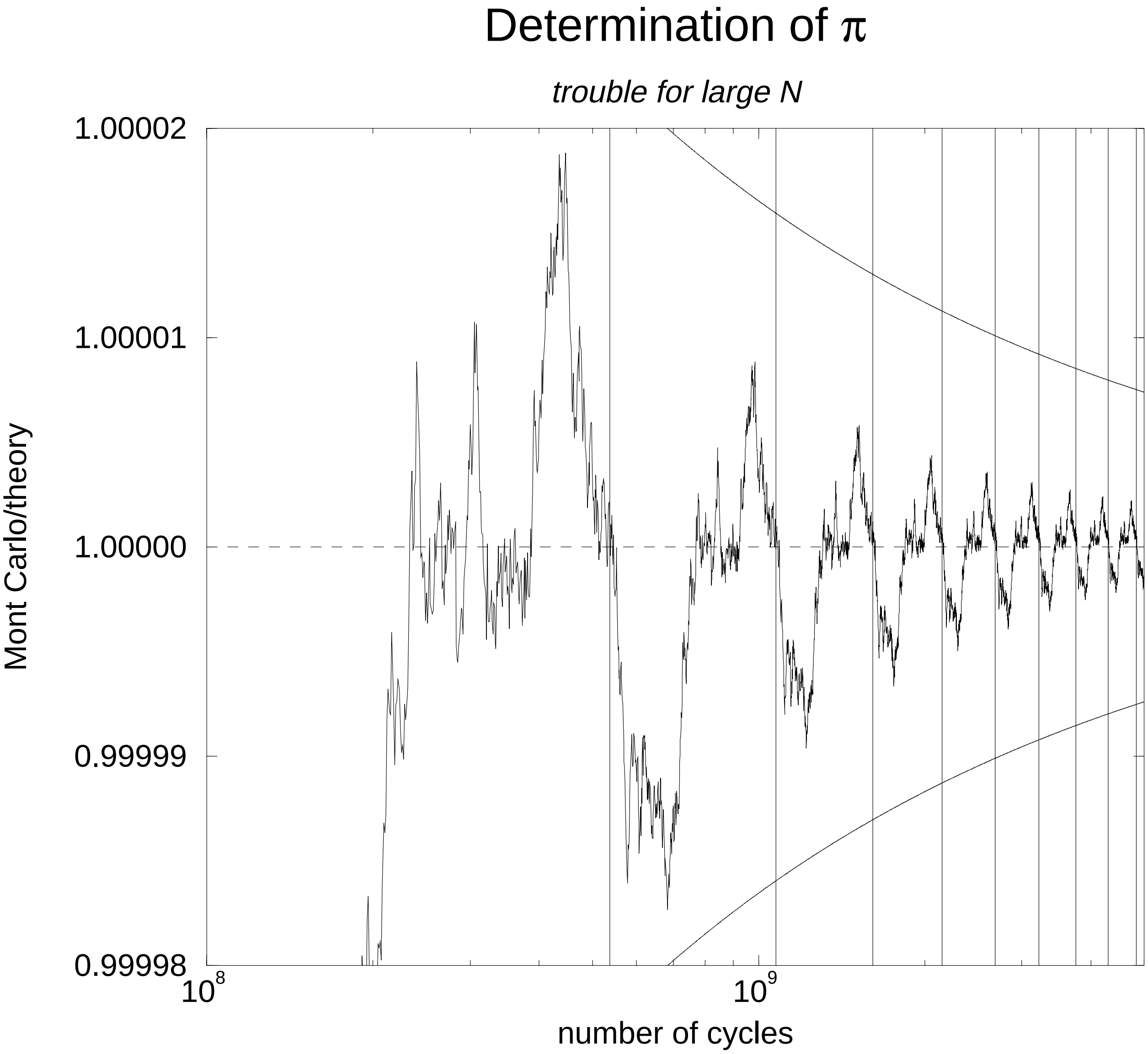

Figure 6.4: A zoom-in on the large cycle behaviour of previous figure.

in 10^7) at the point where the cycle restarts[1]. As a result, the calculated value appears to be well below the 1σ bounds predicted by the Central Limit theorem in the range shown. These are all strong signals that the random number has been "looped". It is never wise to use more than a fraction, say 1/10th of the sequence. Note that the latter half of the sequence anti-correlates with the first half. This could lead to spurious results if a sequence is exhausted.

It is also false to conclude: *"Despite the periodic structure, the result converges to the correct answer."* **Wrong!** We just happened to be lucky in this case! The result converged to about $1 + 5 \times 10^{-7}$ after one complete cycle, nearly the correct answer but not **the** correct answer. The "error term" after one cycle just happens to be very small for this application. If we ran this application for about 12×10^9 cycles, we would note a "false convergence" to $1+5\times10^{-7}$ whereas the 1σ bounds would be smaller and converging on unity.

An example of "false convergence" is given in Figure 6.5 which is the same example except that a large number of random numbers were thrown away after each sample of π, as if to simulate many random numbers being employed in a different aspect of a calculation. Although the example is somewhat extreme, it depicts clearly an anomalous result that will never converge to the correct answer.

The object of this lesson is to warn against using random number generators beyond a fraction of their sequence length.

The signals that you have cycled the random number generator are:

- The tally exhibits a period structure.
- The tally converges in a way that is contrary to Central Limit predictions, assuming that the second moment of the tally exists.
- The presence of false convergence, which may be very difficult to detect.

[1] This is due to 2D space being nearly uniformly filled by this MCRNG.

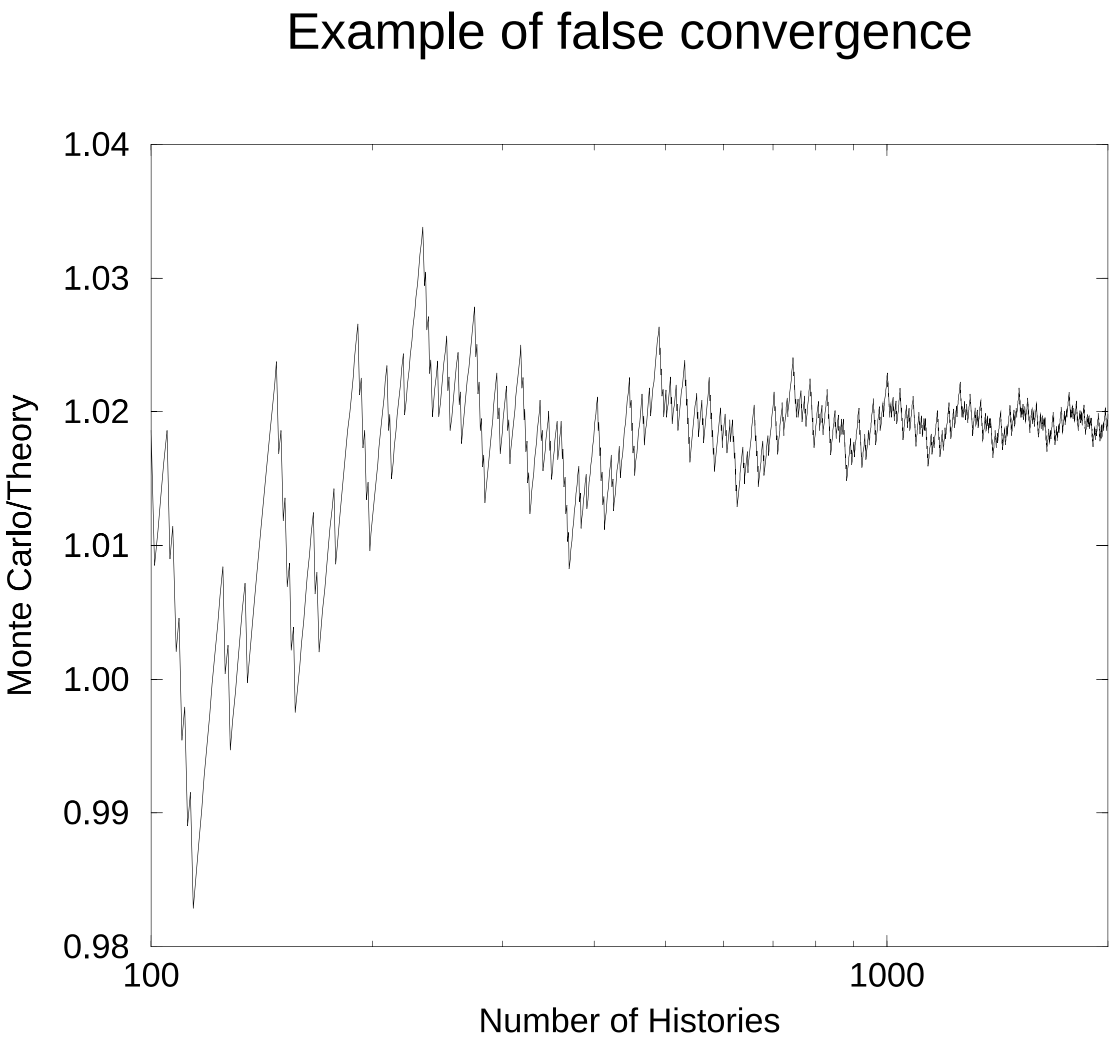

Figure 6.5: An example of false convergence.

6.2 Accumulation errors

Consider the summation:

$$s = \sum_{i=1}^{N} (1/N) \; . \tag{6.1}$$

Of course, mathematically the result is $s \equiv 1$. Numerically, however, it is a different story. The result of s *vs.* N is give in Figure 6.6.

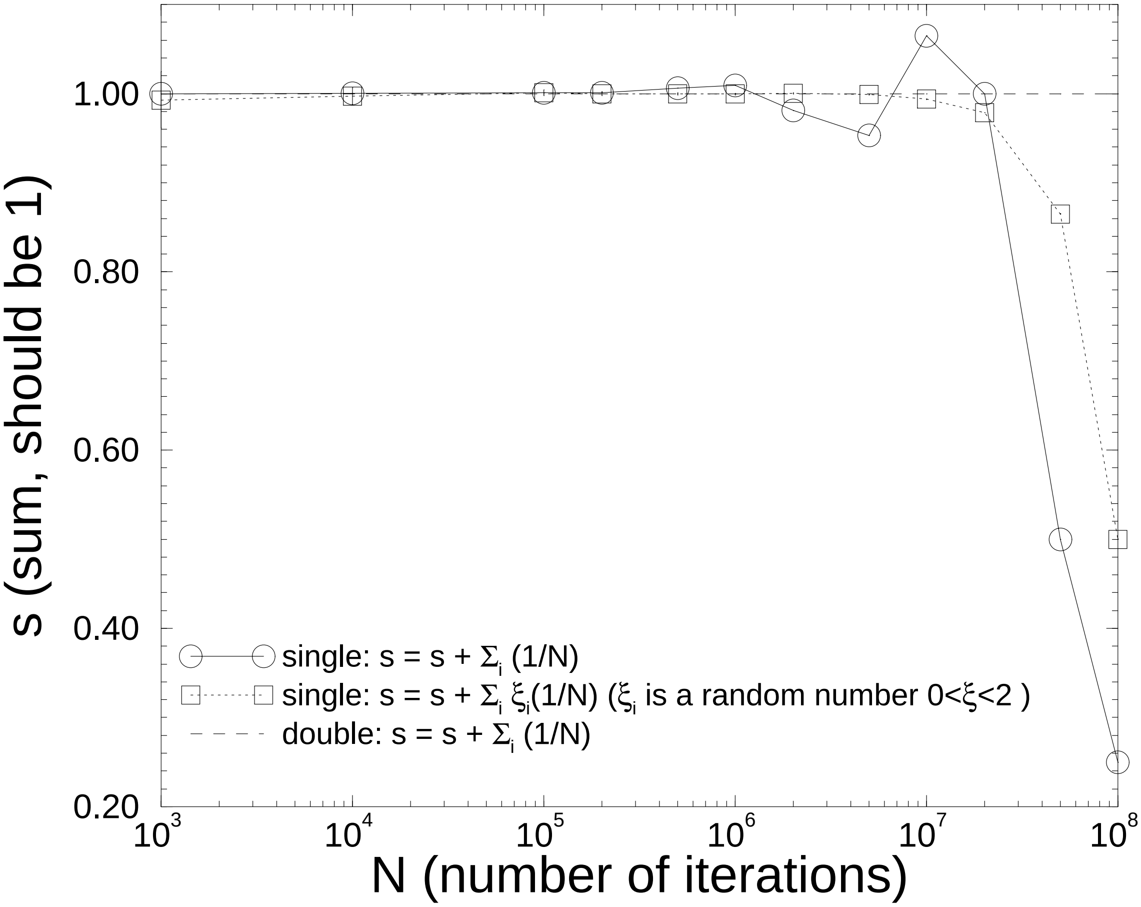

Figure 6.6: An example of constant and random accumulation errors in single-precision arithmetic.

We note that an accumumulation error is seen starting from about 10^6 iterations when the accumulation is done using single precision Fortran, a 32-bit representation of floating-point numbers. The shape of this curve is the result of constant accumulated round-off error, can be positive or negative, but eventually underestimates due to truncation error. The precise shape of the artefact is probably machine dependent. The reason for the underestimate at large value of N is because $1 + 10^{-8} \equiv 1$ in single precision arithmetic.

Another expression of the similar thing is:

$$s = \sum_{i=1}^{N} (2r/N) \ . \tag{6.2}$$

where r is a random number uniformly distributed on $[0, 1]$. Since $\langle r \rangle = 1/2$, $s \equiv 1$ mathematically as well. However, a numerical evaulation exhibits some accumulation error starting from about 10^7 iterations. This is also seen in Figure 6.6. The shape of this curve is the result of random accumulation error and is probably common to all single-precision architectures. A double precision accumulation is shown as well. For double precision, $1 + 10^{-8} = 1.00000001$ and no accumulation error is evident. Double precision errors would start at about 10^{15} to 10^{16} iterations, a realm where no application has dared to go (yet).

The obvious solution to this problem is: Use double precision! However, there are good reasons for using single precison numbers. On some architectures, single precision arithmetic is faster than double precision. (There are counter examples to this as well!) Double precision numbers also take more computer storage. Fetching and storing them can take longer than for single precision numbers. A good rule is to develop your application in double precision. Then, if fast exacution or computer storage become critical to your application, consider single precision for some, if not all of your calculation. However, you must be aware of the shortcomings (pun intended!) of single precision variables and use them with caution.

Bibliography

Chapter 7

Ray tracing and rotations

In this chapter, we begin to enter into a specialization of Monte Carlo methods for particle transport. We discuss how particles are "ray-traced" or "displaced" in space. We also discuss coordinate transformations and rotations. Ray-tracing is related to moving a particle around a medium and/or a geometry. Rotations occur at interaction sites where particles are deflected by scattering angles. These scattering angles are usually expressed in terms of "laboratory coordinates" where the particle was assumed to be travelling along the positive z-axis. Coordinate transformations are useful for specifying geometrical elements in some easy-to-describe coordinate system. Coordinate transformations maybe applied to situate these geometrical elements in the position appropriate to the application.

First we introduce the concept of a particle's phase space. For the meantime we will assume it takes the following form:

$$\{\vec{x}, \vec{u}\} \ . \tag{7.1}$$

The particle's phase space is simply a collection of dynamic variables that describe the particle's absolute location in space referred back to the origin of some laboratory coordinate system, $\vec{x}$, and its direction, $\vec{u}$, referred back to a fixed set of axes in the same laboratory coordinate system. The laboratory coordinate system is usually the one where the fixed components of the experiment resides. $\vec{x}$ represents the 3-vector $\vec{x} = (x, y, z)$ and $\vec{u}$ represents the 3-vector $\vec{u} = (u, v, w)$ the direction cosines. The direction cosines may also be expressed in terms of the angles $\vec{u} = (u, v, w) = (\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta)$, where θ and ϕ are the polar and azimuthal angles respectively.

The phase space can be much more descriptive. It can include the particle's energy, its particle type (in a mixed particle simulation), an identifier that indicates which geometrical element it is in, its spin, the time, or anything that the application may demand. We may even include information that physically would not be available such as, for example, the number of Compton interactions that a photon has undergone. For now, however, we keep it as simple as possible.

7.1 Displacements

Transport is carried out using some very simple geometrical constructs. Given that a particle has a position $\vec{x}_0$ and direction $\vec{u}_0$ and distance to travel s, the new position, $\vec{x}$, is given by:

$$\vec{x} = \vec{x}_0 + \vec{u}_0 s, \tag{7.2}$$

or

$$\begin{aligned} x &= x_0 + u_0 s \\ y &= y_0 + v_0 s \\ z &= z_0 + w_0 s \end{aligned} \tag{7.3}$$

where $\vec{x_0} = (x_0, y_0, z_0)$, $\vec{x} = (x, y, z)$. That is, transport effects the following change in the particle's phase space:

$$\mathcal{T}(\{\vec{x}_0, \vec{u}_0\}, s) = \{\vec{x}_0 + \vec{u}_0 s, \vec{u}_0\} = \{\vec{x}, \vec{u}_0\} \ , \tag{7.4}$$

in operator notation.

The displacement s in this development is a geometric measure of $|\vec{x} - \vec{x}_0|$. There may be other measures such as $v\Delta t$ where v is the velocity (assumed to be constant) and Δt is the time over which the transport takes place. Sometimes, s may be more difficult to obtain. For example, a charged particle in an electric or magnetic field will be deflected as it moves and the determination of s may be quite involved. For the present, we keep it simple.

7.2 Rotation of coordinate systems

Let us consider the problem of the representation of a 3-vector in a coordinate system where the axes are rotated. Figure 7.1 shows the connection between the 3-vector as measured in the laboratory system $\vec{x}$ with respect to another system $\vec{x}'$ where there has been a rotation by an angle ϕ in the positive sense (right hand rule) about positive z axis. By inspection,

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} \tag{7.5}$$

Another rotation of the coordinates in the $\vec{x}'$ system by an angle θ about the y' axis as shown in Figure 7.2 gives:

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} x'' \\ y'' \\ z'' \end{pmatrix} \tag{7.6}$$

Combining the rotations by matrix multiplication gives the coordinates of the 3-vector as measured in the laboratory system $\vec{x}$ with respect to the rotated system system $\vec{x}''$.

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \cos\theta\cos\phi & -\sin\phi & \sin\theta\cos\phi \\ \cos\theta\sin\phi & \cos\phi & \sin\theta\sin\phi \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} x'' \\ y'' \\ z'' \end{pmatrix} \tag{7.7}$$

We define the rotation matrix

$$\Re(\theta, \phi) = \begin{pmatrix} \cos\theta\cos\phi & -\sin\phi & \sin\theta\cos\phi \\ \cos\theta\sin\phi & \cos\phi & \sin\theta\sin\phi \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \tag{7.8}$$

that transforms 3-vectors from their description in a rotated coordinate systems to that of the laboratory system.

Rotation matrix has some special properties. The determinant of the rotation matrix is unity, *i.e.*

$$\|\Re(\theta, \phi)\| = 1 \ . \tag{7.9}$$

The inverse of the rotation matrix, $\Re^{-1}(\theta, \phi)$ is its transpose, $\Re^{\mathrm{T}}(\theta, \phi)$, that is

$$\Re^{-1}(\theta, \phi) = \Re^{\mathrm{T}}(\theta, \phi) \ , \tag{7.10}$$

or

$$\Re^{-1}_{ij}(\theta, \phi) = \Re_{ji}(\theta, \phi) \ . \tag{7.11}$$

The inverse matrices are important! They express vectors in local coordinate systems when they are expressed originally in laboratory coordinate systems. We will make use of this in the following section.

7.3 Changes of direction

Returning now to the deflection of particles, imagine that a particle with direction $\vec{u}_0 = (u_0, v_0, w_0) = (\sin\theta_0\cos\phi_0, \sin\theta_0\sin\phi_0, \cos\theta_0)$ is scattered by angles Θ and Φ with respect to the particle's $\vec{u}_0$ direction. The new direction is given by:

$$\vec{u} = \Re(\theta_0, \phi_0)\Re(\Theta, \Phi)\Re^{-1}(\theta_0, \phi_0)\vec{u}_0 \ . \tag{7.12}$$

The explanation is as follows:

- $\Re^{-1}(\theta_0, \phi_0)\vec{u}_0$ rotates the vector $\vec{u}_0$ in laboratory coordinates to a local coordinate system where the particle is going along the z axis. In fact, $\Re^{-1}(\theta_0, \phi_0)\vec{u}_0 = \hat{z} = (0, 0, 1)$.

- $\Re(\Theta,\Phi)\hat{z}$ effects a rotation of the coordinate system such that the new direction cosines are given by the angles of the scatter. The result of $\Re(\Theta,\Phi)\hat{z}$ is familiar, $\Re(\Theta,\Phi)\hat{z} = (\sin\Theta\cos\Phi, \sin\Theta\sin\Phi, \cos\Theta)$.
- Finally, $\Re(\theta,\phi)\Re(\Theta,\Phi)\hat{z}$ rotates the coordinate system from the particle's local coordinates system in which had it travelling along the positive z axis before the scattering to the laboratory system where is was travelling in the direction $\vec{u}_0$ before the scattering event.

The whole process can be written out explicitly by performing the matrix multiplications:

$$\begin{aligned} u = \sin\theta\cos\phi &= u_0\cos\Theta + \sin\Theta(w_0\cos\Phi\cos\phi_0 - \sin\Phi\sin\phi_0) \\ v = \sin\theta\sin\phi &= v_0\cos\Theta + \sin\Theta(w_0\cos\Phi\sin\phi_0 + \sin\Phi\cos\phi_0) \\ w = \cos\theta &= w_0\cos\Theta - \sin\Theta\sin\theta_0\cos\Phi \end{aligned} \tag{7.13}$$

Returning now to our operator and phase-space notation, rotation effects the following change in the particle's phase space:

$$\mathcal{R}(\{\vec{x}_0,\vec{u_0}\},\Theta,\Phi) = \{\vec{x}_0, \Re(\theta_0,\phi_0)\Re(\Theta,\Phi)\Re^{-1}(\theta_0,\phi_0)\vec{u}_0\} = \{\vec{x}_0,\vec{u}\}\ . \tag{7.14}$$

7.4 Putting it all together

Imagine that a simulation provides us with a set of pathlengths, $(s_1, s_2, s_3\cdots)$ and an associated set of deflections, $([\Theta_1,\Phi_1],[\Theta_2,\Phi_2],[\Theta_3,\Phi_3]\cdots)$ we have now built up the complete mathematical mechanism for transporting particles. Translation is a well defined mathematical (and computational process) accomplished in Equations 7.2 and 7.3. Rotation is also a well-defined mathematical and computational process described by Equations 7.12 and 7.13. The mechanism by which the set of s's or scattering angles are determined is left to later chapters. For the moment, let us assume that they are given.

The other important realization is that $\mathcal{T}$ and $\mathcal{R}$ *do not commute.* For example.

$$\mathcal{R}(\{\mathcal{T}(\{0,\hat{z}\},1)\},\pi/2,0) = \{\hat{z},\hat{x}\}\ , \tag{7.15}$$

whereas

$$\mathcal{T}(\{\mathcal{R}(\{0,\hat{z}\},\pi/2,0)\},1) = \{\hat{x},\hat{x}\}\ . \tag{7.16}$$

So, the ordering of the operations of transport and scattering are important.

The process of repeated transport and scattering can be written mathematically as:

$$\begin{array}{rcl}
\{\vec{x}_1, \vec{u}_0\} & = & \mathcal{T}(\{\vec{x}_0, \vec{u}_0\}, s_1) \\
\{\vec{x}_1, \vec{u}_1\} & = & \mathcal{R}(\{\vec{x}_1, \vec{u}_0\}, \Theta_1, \Phi_1) \\
\{\vec{x}_2, \vec{u}_1\} & = & \mathcal{T}(\{\vec{x}_1, \vec{u}_1\}, s_2) \\
\{\vec{x}_2, \vec{u}_2\} & = & \mathcal{R}(\{\vec{x}_2, \vec{u}_1\}, \Theta_2, \Phi_2) \\
\{\vec{x}_3, \vec{u}_2\} & = & \mathcal{T}(\{\vec{x}_2, \vec{u}_2\}, s_3) \\
\{\vec{x}_3, \vec{u}_3\} & = & \mathcal{R}(\{\vec{x}_3, \vec{u}_2\}, \Theta_3, \Phi_3) \\
\vdots & &
\end{array} \tag{7.17}$$

A picture of this process is given in Figure 7.3.

We can now invent a scattering scattering scheme and a way of generating the pathlengths s and begin doing Monte Carlo calculations in infinite media!

For example, we can specify as isotropic scattering law:

$$p(\Theta, \Phi) = \frac{\sin(\Theta)}{4\pi} \ . \tag{7.18}$$

and a fixed pathlength, say $s = 1$ in some units and make a realistic simulation of a random walk or Brownian motion in the absence of gravity.

Bibliography

Problems

Using a computer, operating system, computing language and compiler of your choice (tell me what you used):

1.

Chapter 8

Transport in media, interaction models

In the previous chapter it was stated that a simulation provides us with a set of pathlengths, $(s_1, s_2, s_3 \cdots)$ and an associated set of deflections, $([\Theta_1, \Phi_1], [\Theta_2, \Phi_2], [\Theta_3, \Phi_3] \cdots)$ that enter into the transport and deflection machinery developed there. In this chapter we discuss how this data is produced and give some indication as to its origins. We also present some basic interaction models that are facsimiles of the real thing, just so that we can have something to model without a great deal of computational effort.

8.1 Interaction probability in an infinite medium

Imagine that a particle starts at position $\vec{x}_0$ in an arbitrary infinite scattering medium and that the particle is directed along an axis with unit direction vector $\vec{\mu}$. The medium can have a density or a composition that varies with position. We call the probability that a particle exists at position $\vec{x}$ relative to $\vec{x}_0$ *without having collided*, $p_\mathrm{s}(\vec{\mu}\cdot(\vec{x}-\vec{x}_0))$, the *survival probability*. The *change* in probability due to interaction is characterized by an interaction coefficient (in units L^{-1}) which we call $\mu(\vec{\mu}\cdot(\vec{x}-\vec{x}_0))$. This interaction coefficient can change along the particle flight path. The change in survival probability is expressed mathematically by the following equation:

$$\mathrm{d}p_\mathrm{s}(\vec{\mu}\cdot(\vec{x}-\vec{x}_0)) = -p_\mathrm{s}(\vec{\mu}\cdot(\vec{x}-\vec{x}_0))\mu(\vec{\mu}\cdot(\vec{x}-\vec{x}_0))\mathrm{d}(\vec{\mu}\cdot(\vec{x}-\vec{x}_0)) \ . \tag{8.1}$$

The minus sign on the right-hand-side indicates that the probability decreases with pathlength and the proportionality with $p_\mathrm{s}(\vec{\mu}\cdot(\vec{x}-\vec{x}_0))$ on the right-hand-side means the probability of loss is proportional to the probability of the particle *existing* at the location where the interaction takes place.

To simplify the notation we translate and rotate so that $\vec{x}_0 = (0, 0, 0)$ and $\vec{\mu} = (0, 0, 1)$. (We can always rotate and translate back later if we want to.) However, the more complicated notation reinforces the notion that the starting position and direction can be arbitrary and that the interaction coefficient can change *along the path* of the particle's direction. In the simpler notation with a small rearrangement we have:

$$\frac{\mathrm{d}p_{\mathrm{s}}(z)}{p_{\mathrm{s}}(z)} = -\mu(z)\mathrm{d}z \ . \tag{8.2}$$

We can integrate this from $z = 0$ where we assume that $p(0) = 1$ to z and obtain:

$$p_{\mathrm{s}}(z) = \exp\left(-\int_0^z \mathrm{d}z'\mu(z')\right) \ . \tag{8.3}$$

The probability that a particle has interacted *within* a distance z is simply $1 - p_{\mathrm{s}}(z)$. This is also called the *cumulative probability:*

$$c(z) = 1 - p_{\mathrm{s}}(z) = 1 - \exp\left(-\int_0^z \mathrm{d}z'\mu(z')\right) \ . \tag{8.4}$$

Since the medium is infinite, $c(\infty) = 1$ and $p_{\mathrm{s}}(\infty) = 0$.

The differential (per unit length) probability for interaction *at* z can be obtained from the derivative of the cumulative probability $c(z)$:

$$p(z) = \frac{\mathrm{d}}{\mathrm{d}z}\left[1 - \exp\left(-\int_0^z \mathrm{d}z'\mu(z')\right)\right] = \mu(z)\exp\left(-\int_0^z \mathrm{d}z'\mu(z')\right) = \mu(z)p_{\mathrm{s}}(z) \ . \tag{8.5}$$

There is a critical assumption in the above expression! It is assumed that the existence of a particle at z means that it has not interacted in any way. $p_{\mathrm{s}}(z)$ is after all, the survival probability. Why is this important? The interaction coefficient $\mu(z)$ may represent a number of possibilities. Choosing which interaction channel the particle takes is *independent* of whatever happened along the way so long as it has survived and what happens is only dependent on the local $\mu(z)$. There will be more discussion of this subtle point later.

8.1.1 Uniform, infinite, homogeneous media

This is the usual case where it is assumed that μ is a constant, independent of position within the medium. That is, the scattering characteristic of the medium is uniform. (The interaction can depend on the energy ofthe incoming particle, however.) In this case we have the simplification:

$$p_{\mathrm{s}}(z) = e^{-\mu z} \ , \tag{8.6}$$

which is the well-known exponential attenuation law for the survival probability of particles as a function of depth. The cumulative probability is:

$$c(z) = 1 - p_{\mathrm{s}}(z) = 1 - e^{-\mu z} \ , \tag{8.7}$$

and the differential probability for interaction z is:

$$p(z) = \frac{\mathrm{d}}{\mathrm{d}z}[1 - e^{-\mu z}] = \mu e^{-\mu z} = \mu p_{\mathrm{s}}(z) \ . \tag{8.8}$$

The use of random numbers to sample this probability distribution was given in Chapter 4.

8.2 Finite media

In the case where the medium is finite, our notion of probability distribution seems to break down since:

$$c(\infty) = 1 - p_{\mathrm{s}}(\infty) = 1 - \exp\left(-\int_0^\infty \mathrm{d}z'\mu(z')\right) < 1 \ . \tag{8.9}$$

That is, the cumulative probability at infinity is less than one because either the particle escapes a finite geometry or $\mu(z)$ is zero beyond some limit. (These are two ways of saying the same thing!)

We can recover our notion of probability theory by drawing a boundary at $z = z_{\mathrm{b}}$ at the end of the geometry and rewrite the cumulative probability as:

$$c(z) = 1 - p_{\mathrm{s}}(z) = 1 - \exp\left(-\int_0^z \mathrm{d}z'\mu(z')\right) + \exp\left(-\int_0^{z_{\mathrm{b}}} \mathrm{d}z'\mu(z')\right)\theta(z - z_{\mathrm{b}}) \ , \tag{8.10}$$

where we assume that $\mu(z) = 0$ for $z \geq z_{\mathrm{b}}$. The probability distribution becomes:

$$p(z) = \mu(z) \exp\left(-\int_0^z \mathrm{d}z'\mu(z')\right) + p_{\mathrm{s}}(z_{\mathrm{b}})\delta(z - z_{\mathrm{b}}) \ . \tag{8.11}$$

The interpretation is that once the particle reaches the boundary, it reaches it with a cumulative probability equal to one minus its survival probability. Probability **is** conserved. In fact, this is *exactly* what is done in a Monte Carlo simulation. If a particle reaches the bounding box of the simulation it is absorbed at that boundary and transport discontinues. If this were not the case, an attempt at particle transport simulation would put the logic into an infinite loop with the particle being transported elsewhere looking for more medium to interact in. In a sense, this is exactly what happens in nature! Particles escaping the earth go off into outer space until they interact with another medium or else they transport to the end of the universe. It is simply a question of conservation of probability!

8.3 Regions of different scattering characteristics

Let us imagine that our "universe" contains regions of different scattering characteristics such that it is convenient to write:

$$\mu(z) \quad = \quad \mu_1(z)\theta(z)\theta(b_1 - z)$$

$$\begin{aligned}
\mu(z) &= \mu_2(z)\theta(z-b_1)\theta(b_2-z) \\
\mu(z) &= \mu_3(z)\theta(z-b_2)\theta(b_3-z) \\
&\vdots
\end{aligned} \tag{8.12}$$

Typically, the μ's may be constant in their own domains. However, the analysis about to be presented is much more general. Note also that there is no fundamental restriction to layered media. Since we have already introduced arbitrary translations and rotations, the notation just means that along the flight path the interaction changes in some way at certain distances that we wish to assign a particular status to.

The interaction probability for this arrangement is:

$$\begin{aligned}
p(z) &= \theta(z)\theta(b_1-z)\mu_1(z)e^{-\int_0^z \mathrm{d}z'\mu_1(z')} \\
&+ \theta(z-b_1)\theta(b_2-z)\mu_2(z)e^{-\int_0^{b_1} \mathrm{d}z'\mu_1(z')}e^{-\int_{b_1}^z \mathrm{d}z'\mu_2(z')} \\
&+ \theta(z-b_2)\theta(b_3-z)\mu_3(z)e^{-\int_0^{b_1} \mathrm{d}z'\mu_1(z')}e^{-\int_{b_1}^{b_2} \mathrm{d}z'\mu_2(z')}e^{-\int_{b_2}^z \mathrm{d}z'\mu_3(z')} \\
&\vdots
\end{aligned} \tag{8.13}$$

Another way of writing this is:

$$\begin{aligned}
p(z) &= \theta(z)\theta(b_1-z)\mu_1(z)e^{-\int_0^z \mathrm{d}z'\mu_1(z')} \\
&+ p_{\mathrm{s}}(b_1)\theta(z-b_1)\theta(b_2-z)\mu_2(z)e^{-\int_{b_1}^z \mathrm{d}z'\mu_2(z')} \\
&+ p_{\mathrm{s}}(b_2)\theta(z-b_2)\theta(b_3-z)\mu_3(z)e^{-\int_{b_2}^z \mathrm{d}z'\mu_3(z')} \\
&\vdots
\end{aligned} \tag{8.14}$$

Now consider a change of variables $b_{i-1} \le z_i \le b_i$ and introduce the conditional survival probability $p_{\mathrm{s}}(b_i|b_{i-1})$ which is the probability that a particle does not interact in the region $b_{i-1} \le z_i \le b_i$ given that it has not interacted in a previous region either. By conservation of probability:

$$p_{\mathrm{s}}(b_i) = p_{\mathrm{s}}(b_{i-1})p_{\mathrm{s}}(b_i|b_{i-1}) \; . \tag{8.15}$$

Then Equation 8.14 can be rewritten:

$$p(z) = p(z_1, z_2, z_3\cdots) = p(z_1) + p_{\mathrm{s}}(b_1)[p(z_2) + p_{\mathrm{s}}(b_2|b_1)[p(z_3) + p_{\mathrm{s}}(b_3|b_2)[\cdots \tag{8.16}$$

What this means is that the variables $z_1, z_2, z_3\cdots$ can be treated as independent. If we consider the interactions over z_1 as independent, then from Equation 8.11 we have:

$$p(z_1) = \mu_1(z_1)\exp\left(-\int_0^{z_1}\mathrm{d}z'\mu_1(z')\right) + p_{\mathrm{s}}(b_1)\delta(z-b_1) \; . \tag{8.17}$$

If the particle makes it to $z = b_1$ we consider z_2 as an independent variable and sample from:

$$p(z_2) = \mu_2(z_2)\exp\left(-\int_0^{z_2} \mathrm{d}z'\mu_2(z')\right) + p_\mathrm{s}(b_2|b_1)\delta(z - b_2) \ . \tag{8.18}$$

If the particle makes it to $z = b_2$ we consider z_3 as an independent variable and sample from:

$$p(z_3) = \mu_3(z_3)\exp\left(-\int_0^{z_3} \mathrm{d}z'\mu_3(z')\right) + p_\mathrm{s}(b_3|b_2)\delta(z - b_3) \ , \tag{8.19}$$

and so on.

A simple example serves to illustrate the process. Consider only two regions of space on either side of $z = b$ with different interaction coefficients. That is:

$$\mu(z) = \mu_1\theta(b - z) + \mu_2\theta(z - b) \ . \tag{8.20}$$

The interaction probability for this example is:

$$p(z) = \theta(b - z)\mu_1 e^{-\mu_1 z} + \theta(z - b)\mu_2 e^{-\mu_1 b} e^{\mu_2(z-b)} \tag{8.21}$$

Now, treat z_1 as independent, that is

$$p(z_1) = \mu_1 e^{-\mu_1 z} + e^{-\mu_1 b}\delta(z - b) \tag{8.22}$$

If the particle makes it to $z = b$ we consider z_2 as an independent variable and sample from:

$$p(z_2) = \mu_2 e^{-\mu_2 z_2} \tag{8.23}$$

which is the identical probability distribution implied by Equation 8.21.

The importance of this proof is summarized as follows. The interaction of a particle is dependent only on the local scattering conditions. If space is divided up into regions of locally constant interaction coefficients, then we may sample the distance to an interaction by considering the space to be uniform in the local interaction coefficient. We sample the distance to an interaction and transport the particle. If a boundary demarcating a region of space with different scattering characteristics interrupts the particle transport, *we may stop at that boundary and resample using the new interaction coefficient of the region beyond the boundary.* The alternative approach would be to invert the cumulative probability distribution implied by Equation 8.4,

$$\int_0^z \mathrm{d}z'\mu(z') = -\log(1 - r) \ , \tag{8.24}$$

where r is a uniform random number between 0 and 1. The interaction distance z would be determined by summing the interaction coefficient until the equality in Equation 8.24 is satisfied. In some applications it may be efficient to do the sampling directly according to the above. In other applications it may be more efficient to resample every time the interaction coefficient changes. It is simply a trade-off between the between the time taken to index the look-up table for $\mu(z)$ and recalculating the logarithm in a resampling procedure.

8.4 Obtaining μ from microscopic cross sections

A microscopic cross section is the probability per unit pathlength that a particle interacts with one scattering center per unit volume. Thus,

$$\sigma = \frac{\mathrm{d}p}{n\,\mathrm{d}z} \ , \tag{8.25}$$

where dp is the differential probability of "some event" happening over pathlength dz and n is the number density of scattering centers that "make the event happen". The units of cross section are L^2. A special quantity has been defined to measure cross sections, the *barn.* It is equal to 10^{-24} cm^2. This definition in terms of areas is essentially a generalization of the classical concept of "billiard ball" collisions.

Consider a uniform beam of particles with cross sectional area A_b impinging on a target of thickness dz as depicted in Figure 8.1. The scattering centers all have cross sectional area σ. The density of scattering centers is n so that there are $nA_\mathrm{b}\mathrm{d}z$ scattering centers in the beam. Thus, the probability per unit area that there will be an interaction is:

$$\mathrm{d}p = \frac{nA_\mathrm{b}\mathrm{d}z\sigma}{A_\mathrm{b}} \ , \tag{8.26}$$

which is just the fractional area occupied by the scattering centers. Hence,

$$\frac{\mathrm{d}p}{\mathrm{d}z} = n\sigma \ , \tag{8.27}$$

which is the same expression given in Equation 8.25. While the classical picture is useful for visualizing the scattering process it is a little problematic in dealing with cross sections that can be dependent on the incoming particle energy or that can include the notion that the scattering center may be much larger or smaller than its physical size. The electromagnetic scattering of charged particles (*e.g.* electrons) from unshielded charged centers (*e.g.* bare nuclei) is an example where the cross section is much larger than the physical size. Much smaller cross sections occur when the scattering centers are transparent to the incoming particles. An extreme example of this would be neutrino-nucleon scattering. On the other hand, neutron-nucleus scattering can be very "billiard ball " in nature. Under some circumstances contact has to be made before a scatter takes place.

Another problem arises when one can no longer treat the scattering centers as independent. Classically this would happen if the targets were squeezed so tightly together that their cross sectional areas overlapped. Another example would be where groups of targets would act together collectively to influence the scattering. These phenomena are treated with special techniques that will not be dealt with directly in this book.

The probability per unit length, d$p\,$dz is given a special symbol μ. That is,

$$\mu = \frac{\mathrm{d}p}{\mathrm{d}z} \ , \tag{8.28}$$

which we have been discussing in the previous sections.

The cross section, σ can also depend on the energy, E, of the incoming particle. In this case we may write the cross section differential in energy as $\sigma(E)$. The symbol σ is reserved for total cross section. If there is a functional dependence it is assumed to be differential in that quantity.

Once the particle interacts, the scattered particle can have a different energy, E' and direction, Θ, Φ relative to the initial z directions. The cross section can be differential in these parameters as well. Thus,

$$\sigma(E) = \int \mathrm{d}E' \, \sigma(E, E') = \int \mathrm{d}\Omega \, \sigma(E, \Theta, \Phi) = \int \mathrm{d}E' \int \mathrm{d}\Omega \, \sigma(E, E', \Theta, \Phi) \ , \tag{8.29}$$

relates the cross section to its differential counterparts. The interaction coefficient μ can have similar differential counterparts.

Now, consider a cross section differential both in scattered energy and angle. We may write this as:

$$\mu(E, E', \Theta, \Phi) = \mu(E) p(E, E', \Theta, \Phi) \ , \tag{8.30}$$

where $p(E, E', \Theta, \Phi)$ is a normalized probability distribution function, which describes the scattering of particle with energy E into energy E' and angles Θ, Φ. We are now ready to specify the final ingredients in particle transport.

Given a particle with energy E and cross section $\sigma(E, E', \Theta, \Phi)$:

- Form its interaction coefficient,

$$\mu(E) = n \int \mathrm{d}E' \int \mathrm{d}\Omega \, \sigma(E, E', \Theta, \Phi) \ . \tag{8.31}$$

- Usually the number density n is not tabulated but it can be determined from:

$$n = \frac{\rho}{A} N_{\mathrm{A}} \ . \tag{8.32}$$

  where ρ is the mass density, A is the atomic weight (g/mol) and N_{A} is Avogadro's number, the number of atoms per mole ($6.0221367(36) \times 10^{23}$ mol^{-1}) [C. 98][1].

- Assuming $\mu(E)$ does not depend on position, provide the distance to the interaction point employing (now familiar) sampling techniques:

$$s = -\frac{1}{\mu(E)} \log(1 - r) \ , \tag{8.33}$$

  where r is a uniformly distributed random number between 0 and 1. Provide this s and execute the transport step $\vec{x} = \vec{x_o} + \vec{\mu} s$. (Do not confuse the direction cosine vector $\vec{\mu}$ with the interaction coefficient μ!)

[1] The latest listing of physical constants and many more good things are available from the Particle Data Group's web page: `http://pdg.lbl.gov/pdg.html`

- Form the marginal probability distribution function in E':

$$m(E,E') = \int \mathrm{d}\Omega \, p(E,E',\Theta,\Phi) \ , \tag{8.34}$$

  and sample E'.

- Form the conditional probability distribution function in Θ, Φ:

$$p(E,\Theta,\Phi|E') = \frac{p(E,E',\Theta,\Phi)}{m(E,E')} \ , \tag{8.35}$$

  sample from it and provide $[\Theta, \Phi]$ to the rotation mechanism that will obtain the new direction after the scattering.

8.5 Compounds and mixtures

Often our scattering media are made up of compounds (*e.g.* water, H_2O) or homogeneous mixtures (*e.g.* air, made up of nitrogen, oxygen, water vapor and a little bit of Argon). How do we form interaction coefficients for these?

We can employ Equations 8.27 and 8.28 and extend them to include partial fractions, n_i of atoms each with its own cross section σ_i:

$$\mu(E) = \sum_i n_i \sigma_i \ . \tag{8.36}$$

Often, however, compounds are specified by fractional weights, w_i, the fraction of mass due to one atomic species compared to the total. From Equation 8.36 and 8.32:

$$\mu(E) = \sum_i \frac{\tilde{\rho}_i}{A_i} N_\mathrm{A} \sigma_i \ . = \sum_i \tilde{\rho}_i \left(\frac{\mu_i}{\rho_i} \right) = \rho \sum_i w_i \left(\frac{\mu_i}{\rho_i} \right) \ , \tag{8.37}$$

or

$$\frac{\mu(E)}{\rho} = \sum_i w_i \left(\frac{\mu_i}{\rho_i} \right) \ . \tag{8.38}$$

where ρ_i is the "normal" density of atomic component i (very often it is μ_i/ρ_i, called the mass attenuation coefficient, that is provided in numerical tabulations), $\tilde{\rho}_i$ is the "actual" mass density of the atomic species in the compound, ρ is the mass density of the compound and the fractional weight of atomic component i in the compound is $w_i = \tilde{\rho}_i/\rho$.

8.6 Branching ratios

Sometimes the interaction coefficient represents the compound process:

$$\mu(E) = \sum_i \mu_i(E) \ . \tag{8.39}$$

For example, a photon may interact through the photoelectric, coherent (Rayleigh), incoherent (Compton) or pair production processes. Thus, from Equation 8.5 we have:

$$p(z) = \sum_i p_i(z) = \sum_i \mu_i(E) e^{-z\mu(E)} = \sum_i \mu_i(E) p_{\mathrm{s}}(z) \ , \tag{8.40}$$

where we assume (a non-essential approximation) that the $\mu_i(E)$'s do not depend on position. Thus the fractional probability for interaction channel i is:

$$P_i = \frac{p_i(z)}{p(z)} = \frac{\mu_i(E)}{\mu(E)} \ . \tag{8.41}$$

In other words, the interaction channel is decided upon the relative attenuation coefficients at the location of the interaction, not on how the particle arrived at that point.

This is one of the few examples of discrete probability distributions and we emphasize this by writing an uppercase P for this probability. The sampling of the branching ratio takes the form of the following recipe:

1. Choose a random number r. If $r \leq \ \mu_1(E)/\mu(E)$, interact via interaction channel 1. Otherwise,
2. if $r \leq \ [\mu_1(E) + \mu_2(E)]/\mu(E)$, interact via interaction channel 2. Otherwise,
3. if $r \leq \ [\mu_1(E) + \mu_2(E) + \mu_3(E)]/\mu(E)$, interact via interaction channel 3. Otherwise...and so on.

The sampling must stop because $\sum_i \mu_i(E)/\mu(E)$ and a random number must always satisfy one of the conditions along the way.

Note that this technique can be applied directly to sampling from the individual atomic species in compounds and mixtures. The fractional probabilities can be calculated according to the discussion of the last section and the sampling procedure above adopted.

8.7 Other pathlength schemes

For complete generality it should be mentioned that there are other schemes for selecting pathlengths. We shall see this in the later chapter on electron transport. In many applications, electron transport steps are provided in a pre-prescribed, non-stochastic way, such as the pathlength for which a certain average energy loss would be obtained, or a pathlength that produces a given amount of angular deflection. Deeper discussion would invoke a long discussion that we will defer until later, after we have built up all the basic of the Monte Carlo technique that we will need to solve a broad class of problems.

8.8 Model interactions

In this section we will introduce several model interactions and employ them to study some basic characteristics of Monte Carlo results.

8.8.1 Isotropic scattering

The scattering of low to medium-energy neutrons (before resonances begin to appear) is nearly isotropic in the center-of-mass. Therefore the scattering of low to medium-energy neutrons from heavy nuclei is very nearly isotropic. Isotropic scattering is sampled from the distribution function:

$$p(\Theta, \Phi)\, \mathrm{d}\Theta\, \mathrm{d}\Phi = \frac{1}{4\pi} \sin\Theta\, \mathrm{d}\Theta\, \mathrm{d}\Phi \ . \tag{8.42}$$

The sampling procedure is:

$$\begin{aligned} \cos\Theta &= 1 - 2r_1 \\ \Phi &= 2\pi r_2 \end{aligned} \tag{8.43}$$

where the r_i are uniform random numbers between 0 and 1.

8.8.2 Semi-isotropic or P_1 scattering

The first order correction to isotropic scattering of low to medium-energy neutrons, a correction that becomes increasingly important for scattering from lighter nuclei is the semi-isotropic cross angular distribution:

$$p(\Theta, \Phi)\, \mathrm{d}\Theta\, \mathrm{d}\Phi = \frac{1}{4\pi}(1 + a\cos\Theta) \sin\Theta\, \mathrm{d}\Theta\, \mathrm{d}\Phi \qquad |a| \le 1 \ . \tag{8.44}$$

The sampling procedure is:

$$\begin{aligned} \cos\Theta &= \frac{2 - a - 4r_1}{1 + \sqrt{1 - a(2 - a - 4r_1)}} \\ \Phi &= 2\pi r_2 \end{aligned} \tag{8.45}$$

where the r_i are uniform random numbers between 0 and 1.

8.8.3 Rutherfordian scattering

A reasonable approximation to the elastic scattering of charged particles is the Rutherfordian distribution:

$$p(\Theta, \Phi)\, \mathrm{d}\Theta\, \mathrm{d}\Phi = \frac{a(2+a)}{4\pi} \frac{\sin\Theta\, \mathrm{d}\Theta\, \mathrm{d}\Phi}{(1-\cos\Theta+a)^2} \qquad 0 \le a < \infty \ . \tag{8.46}$$

The sampling procedure is:

$$\begin{aligned} \cos\Theta &= 1 - 2a\frac{1-r_1}{a+2r_1} \\ \Phi &= 2\pi r_2 \end{aligned} \tag{8.47}$$

where the r_i are uniform random numbers between 0 and 1. An interesting feature f the Rutherfordian distribution is that it becomes the isotropic distribution in the limit $a \longrightarrow \infty$ and the "no-scattering" distribution $\delta(1-\cos\Theta)$ in the limit $a \longrightarrow 0$.

8.8.4 Rutherfordian scattering—small angle form

One approximation that is made to the Rutherfordian distribution in the limit of small a is to make the small-angle approximation to Θ as well. The distribution then takes the form:

$$p(\Theta, \Phi)\, \mathrm{d}\Theta\, \mathrm{d}\Phi = \frac{2a}{\pi} \frac{\Theta\, \mathrm{d}\Theta\, \mathrm{d}\Phi}{(\Theta^2+2a)^2} \qquad 0 \le a < \infty\ , 0 \le \Theta < \infty \ . \tag{8.48}$$

The sampling procedure is:

$$\begin{aligned} \Theta &= \sqrt{\frac{2ar}{1-r}} \\ \Phi &= 2\pi r_2 \end{aligned} \tag{8.49}$$

If $\Theta > \pi$ is sampled in the first step, it is rejected and the distribution in Θ is resampled. One also has to take care that the denominator in the $\sqrt{\ }$ does not become numerically zero through some quirk of the random number generator.

Bibliography

[C. 98] C. Caso *et al.* The 1998 Review of Particle Physics. *European Physical Journal*, C3:1, 1998.

Problems

The following are paper exercises, not computer exercises.

1. Give a derivation of the sampling procedure in Equation 8.45 for the semi-isotropic probability distribution given in Equation 8.44 is the isotropic distribution.
2. Prove that the $a \longrightarrow \infty$ limit of the Rutherfordian distribution given in Equation 8.46 is the isotropic distribution.
3. Prove that the $a \longrightarrow 0$ limit of the Rutherfordian distribution is the δ-function $\delta(1 - \cos\Theta)$.
4. Consider the small-angle form of the Rutherfordian distribution:

   $$p(\Theta, \Phi)\, \mathrm{d}\Theta\, \mathrm{d}\Phi = N \frac{\Theta\, \mathrm{d}\Theta\, \mathrm{d}\Phi}{(\Theta^2 + 2a)^2} \qquad 0 \le a < \infty\ , 0 \le \Theta < \pi\ , \tag{8.50}$$

   where we now exclude the non-physical regime, $\Theta > \pi$. What is the normalization factor, N? Develop the sampling procedure for this distribution.

Chapter 9

Lewis theory

This chapter contains a discussion of Lewis theory [Lew50], which provides us with a very powerful way to evaluate the quality of our Monte Carlo algorithms. Lewis' theory provides us with scattering-model-independent *exact* predictions of spatial moments (*e.g.* $\langle z \rangle$, $\langle x^2 + y^2 \rangle \cdots$), angular moments (*e.g.* $\langle \cos\theta \rangle$, $\langle \sin\theta^2 \cos\phi^2 \cdots$), and their correlations (*e.g.* $\langle z\cos\theta \rangle$, $\langle x \sin\theta \cos\phi \rangle \cdots$). It is one of the most remarkable (and often ignored) developments in transport theory.

Particle transport of the form we consider in this course is described by the linear Boltzmann transport equation:

$$\left[\frac{\partial}{\partial s} + \vec{\mu}\cdot\vec{\nabla} + \Sigma(E)\right]\psi(\vec{x},\vec{\mu},s) = \int_{4\pi} \mathrm{d}\mu'\,\Sigma(\vec{\mu}\cdot\vec{\mu}',E)\psi(\vec{x},\vec{\mu}',s)\ . \tag{9.1}$$

where $\vec{x}$ is the position, $\vec{\mu}$ is a unit vector indicating the direction of the electron, E is the energy of the electron and s is the pathlength. $\Sigma_{\mathrm{s}}(\vec{\mu}\cdot\vec{\mu}',E)$ is the macroscopic differential scattering cross section,

$$\Sigma(E) = \int_{4\pi} \mathrm{d}\mu'\,\Sigma(\vec{\mu}\cdot\vec{\mu}',E) \tag{9.2}$$

is the total macroscopic cross section[1] (probability per unit length) and $\psi(\vec{x},\vec{\mu},E,s)\,\mathrm{d}\vec{x}\,\mathrm{d}\mu\,\mathrm{d}E$ is the probability of there being an electron in $\mathrm{d}\vec{x}$ about $\vec{x}$, in $\mathrm{d}\mu$ about $\vec{\mu}$ and in $\mathrm{d}E$ about E at after having been transported a pathlength s. The boundary condition to be applied is:

$$\psi(\vec{x},\vec{\mu},E,0) = \delta(\vec{x})\delta(\hat{z}-\vec{\mu})\delta(E_0 - E)\ , \tag{9.3}$$

where we have assumed that at the start of transport the particle is at the origin and pointed in the z-direction. ($\hat{z}$ is a unit vector pointing along the z-axis.) The energy of the particle at the start of the transport process is E_0.

[1] There is a slight departure in notation from the previous chapter. The macroscopic cross section was called μ there. In this chapter the appearence of μ and $\vec{\mu}$ together in the same equation seem too awkward to be employed.

This formulation is a very powerful and comprehensive description of particle scattering. It can encompass transport models where the drift between collision points happens with no loss of energy. In this case:

$$s = vt \ , \tag{9.4}$$

where v is the velocity and t is the time, or it can allow for any non-stochastic functional connection between pathlength and energy. One important energy-loss model is the continuous slowing down (CSD) approximation (ir CSDA) which connects pathlength and energy through the relation:

$$s = vt = \int_E^{E_0} \frac{\mathrm{d}E'}{L(E')} \ , \tag{9.5}$$

where $L(E)$ is called the "stopping power" and describes, for example, how electrons slow down in media. The cross section still depends on E which may be calculated from Equation 9.5.

Except under very special circumstances (such as no scattering in simple geometries), Equation 9.5 can not be solved analytically by any means and so the Monte Carlo method is adopted as a practical approach. However, Lewis [Lew50] has presented a "formal" solution to Equation9.1 from which all the spatial-angular moments can be derived. We sketch below and derive the major results.

9.1 The formal solution

Assume that ψ can be written as an expansion in spherical harmonics,

$$\psi(\vec{x}, \vec{\mu}, s) = \sum_{lm} \psi_{lm}(\vec{x}, s) Y_{lm}(\vec{\mu}) \ , \tag{9.6}$$

one finds that

$$\left[\frac{\partial}{\partial s} + \kappa_l\right] \psi_{lm}(\vec{x}, s) = -\sum_{\lambda\mu} \vec{\nabla}\psi_{\lambda\mu}(\vec{x}, s) \cdot \vec{Q}_{lm}^{\lambda\mu} \ , \tag{9.7}$$

where

$$\kappa_l(E) = \int_0^{2\pi} \mathrm{d}\Phi \int_0^{\pi} \mathrm{d}\Theta \, \sin\Theta \, \Sigma(\cos\Theta, E)[1 - P_l(\cos\Theta)] \ , \tag{9.8}$$

and

$$\vec{Q}_{lm}^{\lambda\mu} = \int_{4\pi} \mathrm{d}\mu \, Y_{lm}^*(\vec{\mu}) \, \vec{\mu} \, Y_{\lambda\mu}(\vec{\mu}) \ . \tag{9.9}$$

It has been assumed that the scattering law depends only on Θ and not Φ, that is, it is azimuthally symmetric. The boundary condition in the parametrization is:

$$\psi_{lm}(\vec{x}, 0) = \delta_{m0}\delta(\vec{x})Y_{l0}(0) = \sqrt{\frac{2l+1}{4\pi}}\delta_{m0}\delta(\vec{x}) \ . \tag{9.10}$$

If one considers angular distributions only, then one may integrate over all $\vec{x}$ in Equation 9.7 giving:

$$\left[\frac{\partial}{\partial s} + \kappa_l\right] \psi_l(s) = 0 \ , \tag{9.11}$$

resulting in the solution derived by Goudsmit and Saunderson [GS40a, GS40b]:

$$\psi(\vec{\mu}, s) = \frac{1}{4\pi} \sum_l (2l+1) P_l(\cos\theta) k_l(s) \ , \tag{9.12}$$

where

$$k_l(s) = \exp\left(-\int_0^s \mathrm{d}s' \, \kappa_l(E)\right) \ . \tag{9.13}$$

Equation 9.7 represents a complete formal solution of the linear Boltzmann transport equation but it has never been solved exactly in closed form. However, Equation 9.7 may be employed to extract important information regarding the moments of the distributions.

A long and very boring proof should go here!

Lewis [Lew50] has shown the moments $\langle z \rangle$, $\langle z \cos\Theta \rangle$, and $\langle x^2 + y^2 \rangle$ to be:

$$\langle z \rangle = \int_0^s \mathrm{d}s' \, k_1(s') \ , \tag{9.14}$$

$$\langle z \cos\theta \rangle = \frac{k_1(s)}{3} \int_0^s \mathrm{d}s' \, \frac{1 + 2k_2(s')}{k_1(s')} \ , \tag{9.15}$$

and

$$\langle x^2 + y^2 \rangle = \frac{4}{3} \int_0^s \mathrm{d}s' \, k_1(s') \int_0^{s'} \mathrm{d}s'' \, \frac{1 - k_2(s'')}{k_1(s'')} \ . \tag{9.16}$$

It can also be shown using Lewis's methods that

$$\langle x \sin\theta \cos\phi + y \sin\theta \sin\phi \rangle = \frac{2k_1(s)}{3} \int_0^s \mathrm{d}s' \, \frac{1 - k_2(s')}{k_1(s')} \ , \tag{9.17}$$

$$\langle \vec{x} \cdot \vec{\mu} \rangle = k_1(s) \int_0^s \mathrm{d}s' \, \frac{1}{k_1(s')} \ , \tag{9.18}$$

$$\langle z^2 \rangle = \frac{2}{3} \int_0^s \mathrm{d}s' \, k_1(s') \int_0^{s'} \mathrm{d}s'' \, \frac{1 + 2k_2(s'')}{k_1(s'')} \ , \tag{9.19}$$

and

$$\langle x^2 + y^2 + z^2 \rangle = 2 \int_0^s \mathrm{d}s' \, k_1(s') \int_0^{s'} \mathrm{d}s'' \, \frac{1}{k_1(s'')} \ , \tag{9.20}$$

which gives the radial coordinate after the total transport distance, s. Note that there was an error[2] in Lewis's paper where the factor 1/3 was missing from his version of $\langle z \cos\Theta \rangle$. In

[2] The correction of Lewis's Equation 26 is:

$$H_{l1} = \sqrt{\frac{1}{4\pi(2l+1)}} k_l(s) \int_0^s \mathrm{d}s' \, \frac{l k_{l-1}(s') + (l+1) k_{l+1}(s')}{k_1(s')}$$

the limit that $s \longrightarrow 0$, one recovers from Eqs. 9.15 and 9.19 the results $\lim_{s\longrightarrow 0}\langle z\cos\Theta\rangle = s$ and $\lim_{s\longrightarrow 0}\langle z^2\rangle = s^2$ which are not obtained without correcting the error as described in the footnote.

It warrants repeating that these equations are all "exact" and are independent of the form of the scattering cross section.

9.2 Isotropic scattering from uniform atomic targets

We assume that the cross section has the form:

$$\sigma(\Theta,\Phi)\mathrm{d}\Theta\,\mathrm{d}\Phi = \frac{\sigma_0}{4\pi}\sin\Theta\mathrm{d}\Theta\,\mathrm{d}\Phi\ . \tag{9.21}$$

Using the results of the previous chapter,

$$\Sigma(\Theta,\Phi)\mathrm{d}\Theta\,\mathrm{d}\Phi = \rho\frac{N_{\mathrm{A}}}{A}\frac{\sigma_0}{4\pi}\sin\Theta\mathrm{d}\Theta\,\mathrm{d}\Phi = \frac{\Sigma_0}{4\pi}\sin\Theta\mathrm{d}\Theta\,\mathrm{d}\Phi\ . \tag{9.22}$$

From Equation 9.8

$$\kappa_l(E) = \int_0^{2\pi}\mathrm{d}\Phi\int_0^{\pi}\mathrm{d}\Theta\,\sin\Theta\,\Sigma(\cos\Theta,E)[1-P_l(\cos\Theta)] = \Sigma_0(1-\delta_{l0})\ , \tag{9.23}$$

and from Equation 9.13

$$k_l(s) = \exp\left(-\int_0^s \mathrm{d}s'\,\kappa_l(E)\right) = \delta_{l0} + (1-\delta_{l0})e^{-\Sigma_0 s} = \delta_{l0} + (1-\delta_{l0})e^{-\lambda}\ , \tag{9.24}$$

where λ is a natural measure of distance in terms of the number of "mean-free-paths" (MFP's). If we substitute this into Equations 9.14–9.20, we obtain:

$$\Sigma_0\langle z\rangle = 1-e^{-\lambda}\ , \tag{9.25}$$

$$\Sigma_0\langle z\cos\theta\rangle = \frac{1-(1-2\lambda)e^{-\lambda}}{3}\ , \tag{9.26}$$

$$\Sigma_0\langle x\sin\theta\cos\phi\rangle = \Sigma_0\langle y\sin\theta\sin\phi\rangle = \frac{1-(1+\lambda)e^{-\lambda}}{3}\ , \tag{9.27}$$

$$\Sigma_0\langle\vec{x}\cdot\vec{\mu}\rangle = 1-e^{-\lambda}\ , \tag{9.28}$$

$$\Sigma_0^2\langle z^2\rangle = \frac{2[1+\lambda-(1+2\lambda)e^{-\lambda}]}{3}\ , \tag{9.29}$$

$$\Sigma_0^2\langle x^2\rangle = \Sigma_0^2\langle y^2\rangle = \frac{2[\lambda-2-(2+\lambda)e^{-\lambda}]}{3}\ , \tag{9.30}$$

The reader should consult Lewis's paper [Lew50] for the definition of the H-functions.

$$\Sigma_0^2\langle x^2+y^2+z^2\rangle = 2[\lambda - 1 + e^{-\lambda}] \ . \tag{9.31}$$

The leading order small pathlength ($\lambda \longrightarrow 0$) behavior is:

$$\Sigma_0\langle z\rangle \longrightarrow \lambda \ , \tag{9.32}$$

$$\Sigma_0\langle z\cos\theta\rangle \longrightarrow \lambda \ , \tag{9.33}$$

$$\Sigma_0\langle x\sin\theta\cos\phi\rangle = \Sigma_0\langle y\sin\theta\sin\phi\rangle \longrightarrow \lambda^2/6 \ , \tag{9.34}$$

$$\Sigma_0\langle\vec{x}\cdot\vec{\mu}\rangle \longrightarrow \lambda \ , \tag{9.35}$$

$$\Sigma_0^2\langle z^2\rangle \longrightarrow \lambda^2 \ , \tag{9.36}$$

$$\Sigma_0^2\langle x^2\rangle = \Sigma_0^2\langle y^2\rangle \longrightarrow \lambda^3/9 \ , \tag{9.37}$$

$$\Sigma_0^2\langle x^2+y^2+z^2\rangle \longrightarrow \lambda^2 \ , \tag{9.38}$$

which is to be expected since the particle has not had much chance to scatter away form the z axis.

The large pathlength ($\lambda \longrightarrow \infty$) behavior is:

$$\Sigma_0\langle z\rangle \longrightarrow 1 \ , \tag{9.39}$$

$$\Sigma_0\langle z\cos\theta\rangle \longrightarrow 1/3 \ , \tag{9.40}$$

$$\Sigma_0\langle x\sin\theta\cos\phi\rangle = \Sigma_0\langle y\sin\theta\sin\phi\rangle \longrightarrow 1/3 \ , \tag{9.41}$$

$$\Sigma_0\langle\vec{x}\cdot\vec{\mu}\rangle \longrightarrow 1 \ , \tag{9.42}$$

$$\Sigma_0^2\langle z^2\rangle \longrightarrow 2(\lambda+1)/3 \ , \tag{9.43}$$

$$\Sigma_0^2\langle x^2\rangle = \Sigma_0^2\langle y^2\rangle \longrightarrow 2(\lambda-2)/3 \ , \tag{9.44}$$

$$\Sigma_0^2\langle x^2+y^2+z^2\rangle \longrightarrow 2(\lambda-1) \ , \tag{9.45}$$

Several interesting properties may be derived. The variance associated with the distribution in z is:

$$\Sigma_0^2\langle z^2\rangle - (\Sigma_0\langle z\rangle)^2 = \frac{2[1+\lambda-(1+2\lambda)e^{-\lambda}]}{3} - (1-e^{-\lambda})^2 \tag{9.46}$$

The small pathlength limit is $4\lambda^3/9$ and the large pathlength limit is $(2\lambda-1)/3$. Similarly, the radial variance is:

$$\Sigma_0^2\langle\vec{x}\cdot\vec{x}\rangle - \Sigma_0^2\langle\vec{x}\rangle\cdot\langle\vec{x}\rangle = 2[\lambda-1+e^{-\lambda}] - (1-e^{-\lambda})^2 \ . \tag{9.47}$$

The small pathlength limit is $2\lambda^3/3$ and the large pathlength limit is $(2\lambda-1)$.

These results and other similarly derived ones may be employed to verify, in a scattering-model independent way, the operation of a Monte Carlo code.

Bibliography

[GS40a] S. A. Goudsmit and J. L. Saunderson. Multiple scattering of electrons. *Phys. Rev.*, 57:24 – 29, 1940.

[GS40b] S. A. Goudsmit and J. L. Saunderson. Multiple scattering of electrons. II. *Phys. Rev.*, 58:36 – 42, 1940.

[Lew50] H. W. Lewis. Multiple scattering in an infinite medium. *Phys. Rev.*, 78:526 – 529, 1950.

Problems

1. 10,000 particles start out in an infinite, unbounded medium with initial position vector $\vec{x}_0 = 0$ and initial direction vector $\vec{u}_0 = \hat{z}$. The interaction probability is:

$$p(s_i)\ \mathrm{d}s_i = \Sigma \exp(-\Sigma s_i)\ \mathrm{d}s_i$$

where s_i is a measure of the pathlength of the particle along its current direction of motion starting from its current position to the i'th interaction point. Having interacted, the particle scatters <u>isotropically</u> and can scatter again, repeatedly, according to the same scattering law. This repeated scattering and transport results in a probability distribution function of the form

$$F(\vec{x}, \vec{u}, s)\ \mathrm{d}\vec{x}\ \mathrm{d}\vec{u}\ \mathrm{d}s$$

which you will develop stochastically through a Monte Carlo program. The variable s is a measure of <u>total</u> pathlength travelled, *i.e.* $s = \sum_i^{N(s)} s_i$. Note that the number of interactions $N(s)$ is a probabilistic quantity.

(a) Consider the following moments:

- $\langle x \rangle \pm \sigma_{\langle x \rangle}$ (*i.e.* average x)

- $\langle y \rangle \pm \sigma_{\langle y \rangle}$ (*i.e.* average y)
- $\langle z \rangle \pm \sigma_{\langle z \rangle}$ (*i.e.* average z)
- $\langle x\vec{u} \cdot \hat{x} \rangle \pm \sigma_{\langle x\vec{u} \cdot \hat{x} \rangle}$ (*i.e.* average correlation of x with the x-axis direction cosine)
- $\langle y\vec{u} \cdot \hat{y} \rangle \pm \sigma_{\langle y\vec{u} \cdot \hat{y} \rangle}$ (*i.e.* average correlation of y with the y-axis direction cosine)
- $\langle z\vec{u} \cdot \hat{z} \rangle \pm \sigma_{\langle z\vec{u} \cdot \hat{z} \rangle}$ (*i.e.* average correlation of z with the z-axis direction cosine)
- $\langle xy \rangle \pm \sigma_{\langle xy \rangle}$ (*i.e.* average correlation of x and y)
- $\langle xz \rangle \pm \sigma_{\langle xz \rangle}$ (*i.e.* average correlation of x and z)
- $\langle yz \rangle \pm \sigma_{\langle yz \rangle}$ (*i.e.* average correlation of y and z)
- $\langle x^2 \rangle \pm \sigma_{\langle x^2 \rangle}$ (*i.e.* average x^2)
- $\langle y^2 \rangle \pm \sigma_{\langle y^2 \rangle}$ (*i.e.* average y^2)
- $\langle z^2 \rangle \pm \sigma_{\langle z^2 \rangle}$ (*i.e.* average z^2)

(b) Which moments would you expect to be zero? Why?

(c) Which moments would you expect to be equal to each other? Why?

(d) For $\Sigma = 1$ and $s =$ 0.001, 0.002, 0.005 0.001, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100, 200, 500, 1000, tally the above moments and their estimated errors:

Plot the results divided by s^n as a function of $\log(s)$. n is some power of n such that the correct small and large pathlength asymptotic forms (according to Equations 9.32–9.45) divided by this s^n is a constant. (*e.g.* $\langle z^2 \rangle / s^2$ would be expected to go to the constant 1 for small s and $\langle z^2 \rangle / s$ would be expected to go to the constant 2/3 for large s.) If you expect the moment to be zero (on average), set $n = 0$. Verify that both the $s \longrightarrow 0$ and the $s \longrightarrow \infty$ limits discussed in this chapter are reached.

Chapter 10

Geometry

In particle transport or ray tracing for applications such as the Monte Carlo simulation of particles being transported through media or vacuum, the inclusion of boundaries permits the development of applications of particle transport in geometrical objects. These objects can be very simple, such as a planar interface between two semi-infinite media or a collection of geometrical objects that define a nuclear power vessel or a complex radiation detector.

The technique to address both of these problems is essentially the same. When a particle is about to be transported it knows that it is in a region of space identified by a region number (or some other unique characteristic), its position, its direction and its proposed flight distance (*e.g.* the distance to an interaction point). It also knows that its current location in space is bounded by a finite set of surfaces. So, it interrogates the geometry code associated with each of these to find out which surface it hits, if any, and the distance to that surface.

For most Monte Carlo applications the essential questions to be answered by the geometry modelling code are:

- If my particle starts off at position $\vec{x}_0$ with direction vector μ, and it "flies" in a straight line up to a maximum distance s, will it hit a given surface?
- If the answer to the above is "yes", what is the distance to the intersection?
- And is the answer to the first question above is "yes", what is the region number that lies on the other side?

In this chapter the general problem of solving for the distance to the intersection point of a directed straight line with an arbitrary plane or an arbitrary quadric surface is developed. Quadric surfaces include, for example, spheres, cylinders and cones as well as a few less familiar surfaces. A general strategy for boundary-crossing logic is presented which circumvents ambiguities associated with numerical precision and end-of-step directional uncertainties.

The specific examples of surfaces given are planes, circular cylinders, spheres and circular cones with arbitrary orientation and position. Care is taken to develop the mathematical equations so that they can be computed with numerical accuracy and a discussion on the influence of machine precision on the accuracy of results is given.

This simple design allows the Monte Carlo transport routine to concern itself only with the problem of transport of particles in infinite media and it only needs to know the composition of the medium that particles are being transported in. Thus, when the new region number is communicated to the Monte Carlo transport routine, it merely has to check in its look-up tables to see whether or not the medium has changed and then take appropriate action. This decoupling of transport physics and geometry is a very powerful generalizing feature and permits arbitrary flexibility in specifying the geometry in such a way that the underlying transport mechanisms are identical for all applications.

10.1 Boundary crossing

Let us imagine that a particle is being transported in vacuum, that its initial position is $\vec{x}_0$, its region number is N_0, it has direction $\vec{\mu}$ and that there is only one surface bounding its current region. The geometry code is then interrogated to see of the flight path will strike the surface.

If the geometry modelling code replies:

- Yes, I will hit a certain surface along my flight path,
- The distance to the intersection is s,
- The region number that lies on the other side is N,

particle transport is effected using the equation:

$$\vec{x} = \vec{x}_0 + \vec{\mu}s, \tag{10.1}$$

and one assumes naturally that mathematically (or logically) that the new position is exactly $\vec{x}$ and that any test of where the particle is will yield the answer N for its region number.

However, numerics and mathematics often differ.

In an "ideal" computer where floating point numbers could be specified to absolute accuracy, there would be no ambiguity. However, real computers represent real numbers to within a finite, non-zero precision.[1] Is the particle on a surface or not? Is it exactly on the surface or

[1] It is possible to recode geometry transport in integer arithmetic, thereby avoiding ambiguities. This "quantisation" of space approach has its drawbacks and further discussion would take us out of the scope of this discussion.

has truncation caused an “undershoot” or round-up caused an overshoot? The problem of the finite precision of floating point numbers in computers introduces three position-dependent possibilities that one must consider:

undershoot The transport distance does not quite reach the surface,

exact Numerically, the particle is exactly on the surface,

overshoot The transport distance is slightly overestimated so that the surface is actually crossed.

All of these possibilities occur with varying frequency during the course of a Monte Carlo calculation. In fact, it is correct to say that if you run a geometry code through enough examples with stochastic selection of input parameters, then everything that can happen *will* happen. Therefore, it is necessary to write geometry coding that is robust enough to handle all of these possibilities and also to be aware of the error handling that must be introduced. It has been suggested that all geometry be coded in double or extended precision to avoid these kinds of ambiguities. However, one must realise that double or extended precision does not mean absolute precision. Higher precision reduces the *size* of the undershoot or overshoot but does nothing to cure undershoot or overshoot ambiguities. A geometry code that survives using single precision arithmetic will work at higher precision providing that the coding does not make some intrinsic assumptions on precision or scale. The converse is not true. The routines developed for this course will work for both single and higher precision and the regions of validity for use with both single and double precision is investigated.

There is yet another complicating factor. Transport to a surface can cause the direction of a particle to change! If the boundary represents a reflecting surface, the direction will change according to the law of reflection (discussed in a later chapter). Many charged particle transport schemes apply multiple scattering angular deflection to electron/positron trajectories in media when the flight path is interrupted by a boundary. (This point is discussed further in a later chapter.) Or, with less probability is the possibility that an exact boundary intersection corresponds with the exact numerical point of interaction causing a deflection. The various possibilities are depicted in Figure 10.1.

Fortunately there is a general error handling strategy that resolves these ambiguities:

- Do not calculate flight distances to boundaries that the particle is assumed to be headed away from based on its known direction. The handles the undershoot and exact hit problem in the case the particle assumes that it is still directed away from the surface that it just crossed. However, the particle has to know in advance whether it is “inside” or “outside” of a surface to start with.

- If the flight distance returned is less than or equal to zero, set the transport step to zero and assume that the transport step causes a region change. This strategy resolves

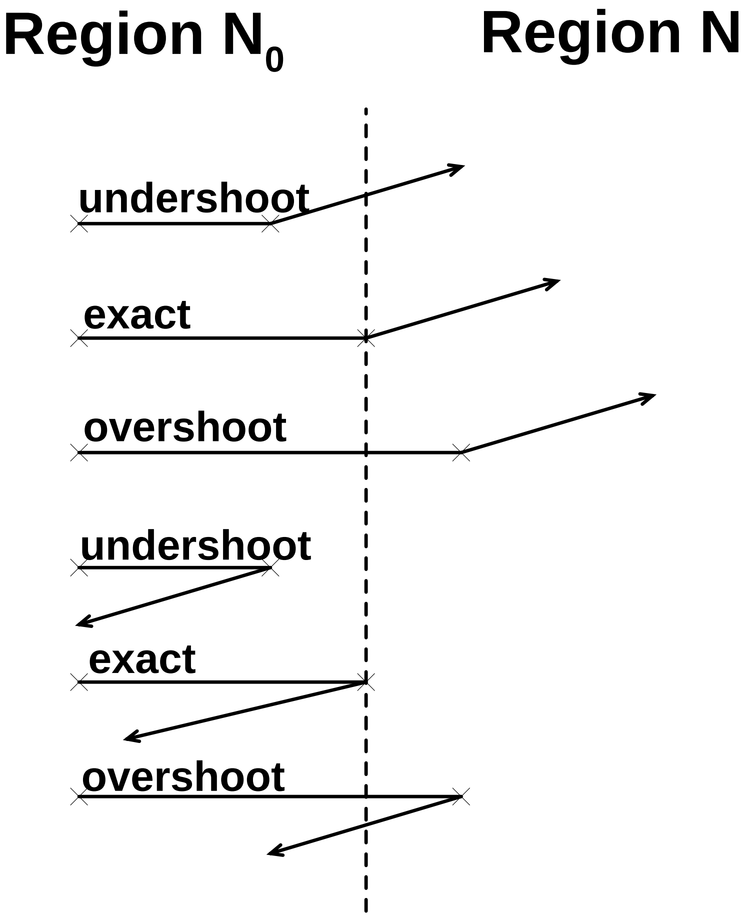

Figure 10.1: Boundary crossing of a particle across a boundary. There are six possibilities corresponding to undershoot, exact surface position, and overshoot with either forward scatter or backscatter.

ambiguities associated with undershoots and exact hits when the particle backscatters form the surface.

Remarkably, this simple logic resolves all boundary transport ambiguities. Its calculational efficiency is excellent, it is completely independent of the scale of the geometry with respect to the transport pathlengths and it heals precision errors "on the spot" not allowing them to accumulate.

Let us enumerate all the possibilities of boundary crossing and demonstrate how this boundary-crossing logic resolves all problems.

undershoot, forward scatter The transport distance does not quite reach the surface, but assumes logically that it is in the next region. The subsequent transport step ignores the surface because it is assumed to be heading away from it. Note that the positioning error is healed "on the spot" in the next transport step.

exact, forward scatter Numerically, the particle is exactly on the surface, but assumes that it is in the next region. The subsequent transport step ignores the surface because it is assumed to be heading away from it.

overshoot, forward scatter The transport distance is slightly overestimated so that the surface is actually crossed. There is no ambiguity in this case since the numerical position and the logical placement of the particle are in agreement.

undershoot, backscatter The transport distance does not quite reach the surface, but assumes logically that it is in the next region. The subsequent transport step checks this surface because it is assumed to be heading toward it. The calculation of the transport distance to the surface yields a small, negative distance. Set the transport distance to zero and let the next step transport the particle to this surface, doing nothing more than resetting the region number to the original. Note that the error is absorbed by the next transport step.

exact, backscatter The transport distance brings the particle exactly to the surface, and the logic assumes that it is in the next region. The subsequent transport step checks this surface because it is assumed to be heading toward it. The calculation of transport distance to the surface yields an exact zero. Employ this zero transport distance and let the next step transport the particle to this surface, doing nothing more than resetting the region number to the original.

overshoot, backscatter The transport distance is slightly overestimated so that the surface is actually crossed. There is no ambiguity in this case since the numerical position and the logical placement of the particle are in agreement. The next transport will be a small, positive step.

It is clear that if there is an overshoot, there is no ambiguity. It is tempting to think that by arranging for an overshoot (a small amount related to the relative or absolute floating point precision) in every case will resolve all ambiguities and be efficient as well. The problem with this approach is that is works well for normal incidence but not very well for grazing incidence. Certain implementations of this strategy introduce a scale dependence into the transport scheme, assume that errors in steps of a certain size can be ignored. Such strategies will either be ineffective if the geometries are very large or introduce positioning errors if the geometrical elements are very small.

10.2 Solutions for simple surfaces

10.2.1 Planes

The general equation for a plane of arbitrary orientation is:

$$\vec{n} \cdot (\vec{x} - \vec{P}) = 0 \tag{10.2}$$

where $\vec{n}$ is the unit normal of the plane and $\vec{P}$ is any point on the surface of the plane. Note the use of the inner product, $\vec{n} \cdot \vec{x} \equiv n_x x + n_y y + n_z z$.

Intersection distance

Inserting Equation 10.1 into Equation 10.2 and solving for s gives:

$$s = -\frac{\vec{n} \cdot (\vec{x}_0 - \vec{P})}{\vec{n} \cdot \vec{\mu}} \tag{10.3}$$

We remark that there is no solution ($s = \infty$) when the particle direction is perpendicular to the normal of the plan ($\vec{n} \cdot \vec{\mu} = 0$). This is the solution of a particle travelling parallel to a plane and never hitting it. Only positive solutions for s are acceptable and this depends upon whether or not the particle is travelling towards the plane.

Adopting the convention that a particle is considered to be outside the plane of it is on the side that the unit normal, $\vec{n}$, is pointing, we enumerate the possibilities:

Case I $\underline{\vec{n} \cdot \vec{\mu} = 0}$

Trajectory is parallel to the plane, no solution

Case II $\underline{\vec{n} \cdot (\vec{x}_0 - \vec{P}) \geq 0 \text{ and the particle is assumed to start from the outside}}$

1. If $\vec{n} \cdot \vec{\mu} < 0$, $s = -\vec{n} \cdot (\vec{x}_0 - \vec{P})/\vec{n} \cdot \vec{\mu}$

2. Elseif $\vec{n} \cdot \vec{\mu} > 0$, no solution

Case III $\vec{n} \cdot (\vec{x}_0 - \vec{P}) \leq 0$ and the particle is assumed to start from the inside

1. If $\vec{n} \cdot \vec{\mu} > 0$, $s = -\vec{n} \cdot (\vec{x}_0 - \vec{P})/\vec{n} \cdot \vec{\mu}$
2. Elseif $\vec{n} \cdot \vec{\mu} < 0$, no solution

Case IV $\vec{n} \cdot (\vec{x}_0 - \vec{P}) < 0$ but the particle is assumed to start from the outside

1. If $\vec{n} \cdot \vec{\mu} < 0$, $s = 0$ effecting a region change
2. Elseif $\vec{n} \cdot \vec{\mu} > 0$, no solution

Case V $\vec{n} \cdot (\vec{x}_0 - \vec{P}) > 0$ but the particle is assumed to start from the inside

1. If $\vec{n} \cdot \vec{\mu} > 0$, $s = 0$ effecting a region change
2. Elseif $\vec{n} \cdot \vec{\mu} < 0$, no solution

The case of a parallel trajectory is handled by **Case I**. The two "normal" conditions in **Case II** and **Case III** handle the eventuality where the particle is exactly on the plane, $\vec{n} \cdot (\vec{x}_0 - \vec{P}) = 0$. **Case IV** and **Case V** handle the anomalies. In the case of an undershoot and backscatter out of the region where the particle thinks it is, then a zero distance is returned so that the next transport step will switch to the correct region number. If the case of an undershoot and forward scatter, no solution is given allowing other surfaces in the geometry to determine the intersection. Note that no correction is made for the undershoot distance and this will be included in the next transport step. Therefore, numerical inaccuracies are not allowed to accumulate.

The FORTRAN codes which accomplishes this for a plane normal to the z axis is:

```
C23456789|123456789|123456789|123456789|123456789|123456789|123456789|12

      subroutine zplane(p0,z0,w,s,inside,hit)
      implicit none

C  Calculates the distance of a particle to a planar surface normal to the
C  z-axis

      logical
     *    inside ! Input:  inside = .true.  => particle thinks it is inside
     *    ,      ! Input:  inside = .false. => particle thinks it is outside
     *    hit    ! Output: hit    = .true.  => particle would hit  the surface
                 ! Output: hit    = .false. => particle would miss the surface
```

```
 real
*    p0,    ! Input:  Point at which the plane cuts the z-axis
*    z0,    ! Input:  z-coordinate of the particle
*    w ,    ! Input:  z-axis direction cosine of the particle
*    s      ! Output: Distance to the surface (if hit)

 if (
*    (inside.and.w.gt.0e0)      !headed towards the surface
*    .or.
*    (.not.inside.and.w.lt.0e0) !headed towards the surface
*   )
*then
     hit = .true.
     s   = max(0e0,(p0-z0)/w)
     return
 else
     hit = .false.
     return
 endif

 end
```

10.3 General solution for an arbitrary quadric

Borrowing from the notation of Olmsted [Olm47], an arbitrary quadric surface in 3(x,y,z)-space[2] can be represented by:

$$f(\vec{x}) = \sum_{i,j=0}^{3} a_{ij} x_i x_j = 0. \tag{10.4}$$

The a_{ij}'s are arbitrary constants and the 4-vector x_i has components $(1, x, y, z)$. The zeroth component is unity by definition allowing a very compact representation and a_{ij} is symmetric with respect to the interchange of i and j, that is $a_{ij} = a_{ji}$. Equation 10.4 is very general and encompasses a wide variety of possibilities including solitary planes (*e.g.* only a_{0i} non-zero), intersecting planes (*e.g.* only a_{11} and a_{22} non-zero), cylinders (circular, elliptical, parabolic and hyperbolic), spheres, spheroids and ellipsoids, cones (circular and elliptical), hyperboloids of one and two sheets and elliptic and hyperbolic paraboloids. These surfaces

[2] The only variance with Olmsted's notation is that the 4th component is labelled as the 0th component in this work.

can be combined to make geometrical objects of arbitrary complexity and are extremely useful in Monte Carlo modeling of physical objects.

Despite having apparently 10 independent constants, Equation 10.4 represents only 10 independent real surfaces (including the simple plane), unique after a translation and rotation to standard position. The three cross terms (a_{ij} for $i \neq j$ and $i, j \geq 1$) can be eliminated by rotation. The resultant equation then only involves terms like x_i^2 and x_i. In addition, providing that a given variable's quadratic constant is non-zero, the linear terms can be eliminated by a translation. The result is that there are only two generic forms:

$$f(\vec{x}) = \sum_{i=1}^{3} a_i x_i^2 + c = 0, \tag{10.5}$$

and

$$f(\vec{x}) = \sum_{i=1}^{2} a_i x_i^2 + b x_3 = 0. \tag{10.6}$$

Equations 10.5 and 10.6 describe only 10 distinct possibilities with real solutions.

1. **ellipsoids:** $a_1^2 x_1^2 + a_2^2 x_2^2 + a_3^2 x_3^2 - c^2 = 0.$
2. **cones:** $a_1^2 x_1^2 + a_2^2 x_2^2 - a_3^2 x_3^2 = 0.$
3. **cylinders:** $a_1^2 x_1^2 + a_2^2 x_2^2 - c^2 = 0.$
4. **hyperboloids of one sheet:** $a_1^2 x_1^2 + a_2^2 x_2^2 - a_3^2 x_3^2 - c^2 = 0.$
5. **hyperboloids of two sheets:** $a_1^2 x_1^2 + a_2^2 x_2^2 - a_3^2 x_3^2 + c^2 = 0.$
6. **elliptic paraboloids:** $a_1^2 x_1^2 + a_2^2 x_2^2 + a_3 x_3 = 0.$
7. **hyperbolic paraboloids:** $a_1^2 x_1^2 - a_2^2 x_2^2 + a_3 x_3 = 0.$
8. **hyperbolic cylinders:** $a_1^2 x_1^2 - a_2^2 x_2^2 + c^2 = 0.$
9. **parabolic cylinders:** $a_1^2 x_1^2 + a_3 x_3 = 0.$
10. **simple planes:** $a_3 x_3 + c = 0.$

The first nine of these are shown[3] in Figure 10.2. (The magnitude of the above constants were all chosen to be unity for the purposes of display. Consequently, the first six of these surfaces shown exhibit at least one axis of rotational symmetry.)

There are other imaginary surfaces (*e.g.* imaginary ellipsoids $a_1^2 x_1^2 + a_2^2 x_2^2 + a_3^2 x_3^2 + c^2 = 0$) that we will not consider nor will we consider quadrics that can be made up of two

[3] Thanks to Keath Borg (wherever you are!) for generating the data for these figures.

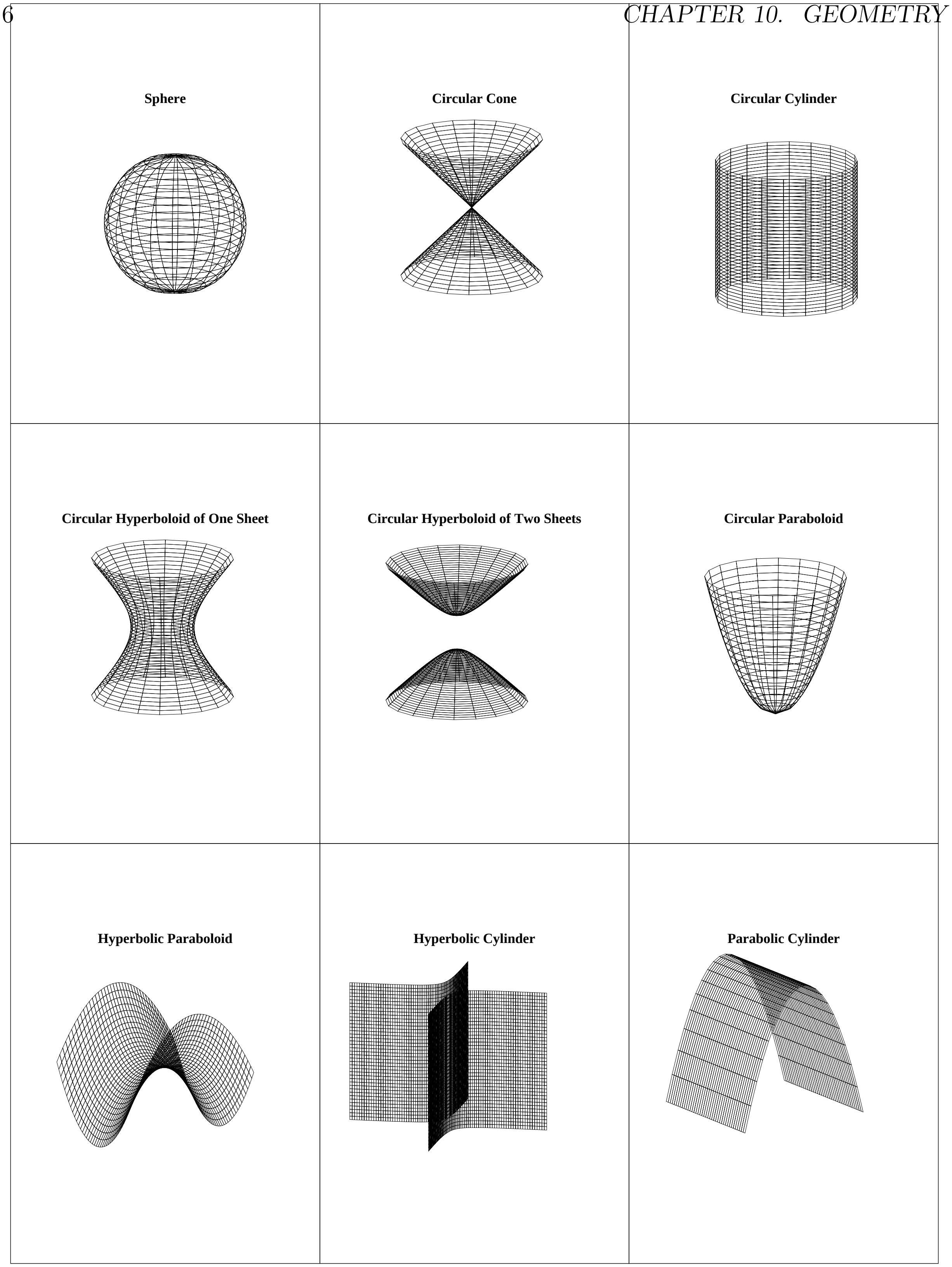

Figure 10.2: The nine real non-planar quadric surfaces.

independent planes in various orientations (*e.g.* intersection planes $a_1^2 x_1^2 - a_2^2 x_2^2 = 0$, parallel planes $a_1^2 x_1^2 - c^2 = 0$, and coincident planes $a_1^2 x_1^2 = 0$).

For more information on the reduction to canonical form, the reader is encouraged to read Olmsted's book [Olm47]. Olmsted also gives the classification of the surfaces and lists the entire set of 17 canonical quadric forms.

10.3.1 Intercept to an arbitrary quadric surface?

We now change notation slightly and write the particle trajectory by the parametric equation:

$$\vec{x} = \vec{p} + \vec{\mu} s \tag{10.7}$$

where the starting position of the particle is $\vec{p} = (p_x, p_y, p_z)$. A positive value of s expresses a distance along the direction that the particle is going (forward trajectory) and a negative value is associated with a distance that the particle came from (backward trajectory). Thus, negative solutions that are found for s below will be rejected.

In Monte Carlo particle transport calculations as well as ray-tracing algorithms a common problem is to find the distance a particle has to travel in order to intersect a surface. This is done by substituting for $\vec{x}$ from Equation 10.7 in Equation 10.4 to give:

$$s^2 \left(\sum_{i,j=0}^{3} a_{ij} \mu_i \mu_j \right) + 2s \left(\sum_{i,j=0}^{3} a_{ij} p_i \mu_j \right) + \left(\sum_{i,j=0}^{3} a_{ij} p_i p_j \right) = 0, \tag{10.8}$$

where we have adopted the convention that $\mu_0 = 0$ and $p_0 = 1$. This is a quadratic equation in s of the form $A(\vec{\mu})s^2 + 2B(\vec{\mu},\vec{p})s + C(\vec{p}) = 0$ where $A(\vec{\mu}) = \sum_{i,j=0}^{3} a_{ij}\mu_i\mu_j$, $B(\vec{\mu},\vec{p}) = \sum_{i,j=0}^{3} a_{ij} p_i \mu_j$ and $C(\vec{p}) = \sum_{i,j=0}^{3} a_{ij} p_i p_j$.

Interpretation of the quadratic constants

The constant $C(\vec{p})$ is identically zero when $\vec{p}$ is on the surface. When $\vec{p}$ is not on the surface, the sign of $C(\vec{p})$ can be interrogated to see if the particle is inside or outside. There is some arbitrariness in the definition of what is "inside" or "outside". A sphere with radius R, for example, has the form $\vec{p}^2 - R^2 = 0$ and in this case $C(\vec{p}) > 0$ when $|\vec{p}| > R$. So, for this example, $C(\vec{p})$ is positive when $\vec{p}$ is outside and negative when inside. However, the same sphere is defined by $R^2 - \vec{p}^2 = 0$ giving opposite interpretation for the signs of $C(\vec{p})$ for points inside and outside. It is best to adopt a constant interpretation and be aware that two points on opposite sides of the surface in the sense that a line joining them intersects the surface only once, have different signs.

For planes and most of the other surfaces, "inside" and "outside" are arbitrary since multiplying Equation 10.5 or Equation 10.6 by a minus sign leaves the surface intact. However, there is one natural interpretation provided by the calculation of the normal to the surface, $\nabla f(\vec{p})$, where $\vec{p}$ is a point on the surface, *i.e.* $C(\vec{p}) = 0$. In the way they were defined, $\nabla f(\vec{p})$ points to the "outside" region which can be defined as follows: If more than one line can be drawn through a point such that the surface is not intersected in either the forward or backward direction, then this point is on the outside. If at most only one such line exists, then the point is on the "inside". This defines the inside and outside in a unique and natural way (inside a sphere, for example). There are three exceptions to this rule, the simple plane, the hyperboloid of one sheet and the hyperbolic paraboloid. In their standard forms given below Equation 10.6, the outside or inside of a plane is completely arbitrary, the outside of a hyperbolic paraboloid contains the positive x_3-axis and the inside of the hyperboloid of one sheet contains the x_3-axis which also seems to be a "natural" choice.

The constant $B(\vec{\mu}, \vec{p})$ is related to the inner product of the particle's direction $\vec{\mu}$ with the normal to the surface at a point $\vec{p}$ when $\vec{p}$ is on the surface. Specifically, $B(\vec{\mu}, \vec{p}) = \sum_{i,j=0}^{3} a_{ij} p_i \mu_j = \frac{1}{2}\vec{\mu} \cdot \nabla f(\vec{p})$. When $\vec{p}$ is on the surface $\nabla f(\vec{p})$ is its normal there. This can be exploited to decide to which side of a surface a particle is going if it happens to be on the surface and is pointed in some direction. Imagine a particle on the surface at point $\vec{p}$ with some direction $\vec{\mu}$ and consider an infinitesimal step ϵ. The sign of $C(\vec{p} + \vec{\mu}\epsilon) = 2\epsilon B(\vec{\mu}, \vec{p}) + O(\epsilon^2)$ will have the sign of $B(\vec{\mu}, \vec{p})$. If $B(\vec{\mu}, \vec{p}) = 0$ for $\vec{p}$ on the surface, it means that the particle is moving in the tangent plane to the surface at that point.

When a particle is on the surface, the constant $A(\vec{\mu})$ can be related to the curvature of the surface. It can be shown[4] that the radius of curvature at the point $\vec{p}$ on the surface in the plane containing the normal to the surface there, $\nabla f(\vec{p})$ and the direction of the particle on the surface, $\vec{\mu}$, is given by $|\nabla f(\vec{p})|/|A(\vec{\mu})|$. There is one case among the surfaces we consider where both $|\nabla f(\vec{p})|$ and $|A(\vec{\mu})|$ vanish simultaneously and that is of a point on the vertex of a cone. In this anomalous case we can take the radius of curvature to be zero.

$A(\vec{\mu})$ vanishes when the particle is travelling parallel to a "ruled line" of the surface, whether on the surface or not. A ruled line is a line that lies entirely on the surface. Quadrics with one or more vanishing quadratic constants (one of the a_i's in Equation 10.5 or 10.6) always possess ruled lines, as do planes, cones, hyperboloids of one sheet and hyperbolic paraboloids. The constant $A(\vec{\mu})$ can also vanish for a particle having a trajectory that is parallel to an asymptote of a hyperboloid, or pointed at the "nose" of a paraboloid or in the plane perpendicular to it.

$A(\vec{\mu})$ can be used to decide where a particle is in relation to a surface in the case that $B(\vec{\mu}, \vec{p})$ and $C(\vec{p})$ vanish, that is when the particle is on the surface and in the plane tangent to it at that point. In this case an infinitesimal transport $C(\vec{p} + \vec{\mu}\epsilon) = A(\vec{\mu})\epsilon^2$ will have the sign of

[4] The way to do this is consider a particle at point $\vec{p}$ on the surface with an initial direction $\vec{\mu}$ tangent to the surface and moving in the plane defined by the normal orthogonal vectors $\vec{\mu}$ and $\nabla f(\vec{p})/|\nabla f(\vec{p})|$. The trajectory of the particle is then described by $f(\vec{p} + \vec{\mu}s_u + (\nabla f(\vec{p})/|\nabla f(\vec{p})|)s_n) = 0$ where s_u and s_n are projections of the particle's position vector on the $\vec{\mu}$ and $\nabla f(\vec{p})/|\nabla f(\vec{p})|$ axes, respectively. This yields the equation of a conic. The radius of curvature is then obtained by the standard equation for motion in a plane, $R_{\mathrm{c}} = \{[1 + (\mathrm{d}s_n/\mathrm{d}s_u)^2]^{3/2}\}/|\mathrm{d}^2 s_n/\mathrm{d}s_u^2|$.

$A(\vec{\mu})$. So, if ($A(\vec{\mu}) > 0$, $B(\vec{\mu}, \vec{p}) = 0$, $C(\vec{p}) = 0$) the particle is headed outside, if ($A(\vec{\mu}) < 0$, $B(\vec{\mu}, \vec{p}) = 0$, $C(\vec{p}) = 0$) the particle is headed inside, and if ($A(\vec{\mu}) = 0$, $B(\vec{\mu}, \vec{p}) = 0$, $C(\vec{p}) = 0$) the particle is on the surface and directed along a ruling and there is no intercept in this case.

For planar surfaces ($A(\vec{\mu}) = 0$ always in this case) there is always a solution for s unless the particle's trajectory is exactly parallel to the plane. If the solution for s is negative, it is rejected since it not a forward solution. If the solution for s is positive, then it represents a solution along the forward trajectory of the particle.

For the non-planar surfaces, the equation for s is quadratic. In general, when $B(\vec{\mu}, \vec{p})^2 - A(\vec{\mu})C(\vec{p}) < 0$, there are no solutions to the quadratic equation, which means that the particle's trajectory misses the surface. If the surface in question is one of the seven with the intuitive inside-outside interpretation, then one might guess that one does not have to test for the positiveness of $B(\vec{\mu}, \vec{p})^2 - A(\vec{\mu})C(\vec{p})$ when the particle is inside, thereby saving computer time. However, there are conditions where the limited numerical precision causes $B(\vec{\mu}, \vec{p})^2 - A(\vec{\mu})C(\vec{p})$ to be negative even when the particle is inside one of the natural surfaces. In this case it can be shown that the particle's trajectory is very close to that of being along a ruling and very close to the surface ($A(\vec{\mu}) \approx 0$, $B(\vec{\mu}, \vec{p}) \approx 0$, $C(\vec{p}) \approx 0$). It is consistent, therefore, to assume that there is no solution in this case even if the trajectory is not exactly on the surface or exactly tangent to it. In this case the particle is assumed to travel along the ruling until it hits another surface in the problem or an interaction effects a change of direction whereupon a decision can be made whether the particle is headed inside or outside the surface.

Employing the error recovery strategy of the previous section, a general algorithm for an arbitrary quadric surface may be sketched:

```
IF $\underline{B^2 - AC < 0}$ Particle does not intersect the surface.
ELSEIF The particle thinks it is outside
    IF $\underline{B \geq 0}$
        IF $\underline{A \geq 0}$ No solution.
        ELSE $s = -(B + \sqrt{B^2 - AC})/A$.
    ELSE $s = \max(0, C/[\sqrt{B^2 - AC} - B])$.
ELSE The particle thinks it is inside
    IF $\underline{B \leq 0}$
        IF $\underline{A > 0}$ $s = (\sqrt{B^2 - AC} - B)/A$.
        ELSE No solution.
    ELSE $s = \max(0, -C/[\sqrt{B^2 - AC} + B])$.
```

All quadric surfaces are special cases that can be solved by this algorithm. Only the constants A, B, C need to be specified for any case. Indeed, this algorithm will work for planes as well but the simplicity of planes motivates the construction of a more efficient algorithm specific to planes only.

The FORTRAN codes which solves the general solution of distance to intercept of a quadric surface is:

```
C23456789|123456789|123456789|123456789|123456789|123456789|123456789|12

      subroutine quadric(A,B,C,s,inside,hit)
      implicit none

C  Calculates the first positive distance to an arbitrary quadric surface

      logical
     *    inside ! Input:  inside = .true.  => particle thinks it is inside
     *    ,      ! Input:  inside = .false. => particle thinks it is outside
     *    hit    ! Output: hit    = .true.  => particle would hit  the surface
                 ! Output: hit    = .false. => particle would miss the surface

      real
     *     A,    ! Input: Quadratic coefficient A
     *     B,    ! Input: Quadratic coefficient B
     *     C,    ! Input: Quadratic coefficient C
     *     s     ! Output: Distance to the surface (if hit)

      real
     *     Q     ! Internal: quadratic coefficient


      Q = B**2 - A*C

      if (Q .lt. 0e0) then
          hit = .false.
          return
      end if

      if (inside) then              !inside the surface
          if (B .le. 0e0) then      !headed away from surface
              if (A .gt. 0e0) then  !but, surface curving up
                  hit = .true.      !always a hit in this case
                  s = (sqrt(Q) - B)/A
```

```
            return
        else                  !surface curving away and headed in
            hit = .false.     !never a hit in this case
            return
        end if
    else                      !headed toward the surface
        hit = .true.          !always a hit in this case
        s = max(0e0, -C/(sqrt(Q) + B))
        return
    end if
end if

!Must be outside the surface

if (B .ge. 0e0) then     !headed away
    if (A .ge. 0e0) then !surface curves away
        hit = .false.    !never a hit in this case
        return
    else                 !surface curves up
        hit = .true.     !always a hit in this case
        s = -(sqrt(Q) + B)/A
        return
    end if
else                     !headed toward the surface
    hit = .true.         !always a hit in this case
    s = max(0e0, C/(sqrt(Q) - B))
    return
end if

end
```

Given this algorithm, the only thing one needs to do for any quadric surface is to specify the constants A, B, C and present them to the quadratic solver. The following sections describe some examples of this.

10.3.2 Spheres

The general equation for a sphere is:

$$(\vec{x} - \vec{X})^2 - R^2 = (x - X)^2 + (y - Y)^2 + (z - Z)^2 - R^2 = 0 \qquad (10.9)$$

where $\vec{X} \equiv (X, Y, Z)$ is the location of the center of the sphere and R is its radius.

Intercept distance

Substituting the equation for particle trajectory, Equation 10.1, into the above yields a quadratic equation of the form $As^2 + 2Bs + C = 0$, where the quadratic constants, A, B and C, are:

$$\begin{aligned}
A &= 1 \\
B &= \vec{\mu} \cdot (\vec{x}_0 - \vec{X}) \\
&= u(x_0 - X) + v(y_0 - Y) + w(z_0 - Z) \\
C &= (\vec{x}_0 - \vec{X})^2 - R^2 \\
&= (x_0 - X)^2 + (y_0 - Y)^2 + (z_0 - Z)^2 - R^2 \qquad (10.10)
\end{aligned}$$

These constants may be employed in the general quadric surface forward-distance solver algorithm described previously.

The FORTRAN code that calculates the quadratic constants for the case when the sphere is centered at the origin is:

```
C23456789|123456789|123456789|123456789|123456789|123456789|123456789|12

      subroutine csphere(R,x0,y0,z0,u,v,w,s,inside,hit)
      implicit none

C  Calculates the distance to a sphere centered at (0,0,0)

      logical
     *    inside ! Input:  inside = .true.  => particle thinks it is inside
     *    ,      ! Input:  inside = .false. => particle thinks it is outside
     *    hit    ! Output: hit    = .true.  => particle would hit  the surface
                 ! Output: hit    = .false. => particle would miss the surface

      real
     *    R,     ! Input:  Radius of the cylinder
     *    x0,    ! Input:  x-coordinate of the particle
     *    y0,    ! Input:  y-coordinate of the particle
     *    z0,    ! Input:  z-coordinate of the particle
     *    u ,    ! Input:  x-axis direction cosine of the particle
     *    v ,    ! Input:  y-axis direction cosine of the particle
     *    w ,    ! Input:  w-axis direction cosine of the particle
     *    s      ! Output: Distance to the surface (if hit)

      real
```

```
*     A,    ! Internal: Quadratic coefficient A
*     B,    ! Internal: Quadratic coefficient B
*     C     ! Internal: Quadratic coefficient C

      A = 1e0    ! i.e. u**2 + v**2 + w**2 = 1
      B = u*x0  + v*y0  + w*z0
      C = x0**2 + y0**2 + z0**2 - R**2

      call quadric(A,B,C,s,inside,hit) ! Get the generic quadric solution

      return
      end
```

10.3.3 Circular Cylinders

The general equation for a circular cylinder is:

$$(\vec{x}-\vec{P})^2-[(\vec{x}-\vec{P})\cdot\vec{U}]^2-R^2=0 \qquad (10.11)$$

where $\vec{P}\equiv(P_x,P_y,P_z)$ is any fixed point on the axis of the cylinder, $\vec{U}$ is the direction vector of the axis of the cylinder, and R is its radius.

Intercept distance

Substituting the equation for particle trajectory, Equation 10.1, into the above yields a quadratic equation of the form $As^2+2Bs+C=0$, where the quadratic constants, A, B and C, are:

$$\begin{aligned} A &= 1-(\vec{\mu}\cdot\vec{U})^2 \\ B &= \vec{\mu}\cdot\{(\vec{p}-\vec{P})-\vec{U}[(\vec{p}-\vec{P})\cdot\vec{U}]\} \\ C &= (\vec{p}-\vec{P})^2-[(\vec{p}-\vec{P})\cdot\vec{U}]^2-R^2 \qquad (10.12) \end{aligned}$$

These constants may be employed in the general quadric surface forward-distance solver algorithm described previously.

The FORTRAN code that calculates the quadratic constants for the case when the cylinder is cylindrical, aligned and centered on the z axis is:

```
C23456789|123456789|123456789|123456789|123456789|123456789|123456789|12
```

```
      subroutine ccylz(R,x0,y0,u,v,s,inside,hit)
      implicit none

C  Calculates the distance to a circular cylinder centered and aligned along
C  the z-axis

      logical
     *    inside ! Input:  inside = .true.  => particle thinks it is inside
     *    ,      ! Input:  inside = .false. => particle thinks it is outside
     *    hit    ! Output: hit    = .true.  => particle would hit  the surface
                 ! Output: hit    = .false. => particle would miss the surface

      real
     *    R,     ! Input:  Radius of the cylinder
     *    x0,    ! Input:  x-coordinate of the particle
     *    y0,    ! Input:  y-coordinate of the particle
     *    u ,    ! Input:  x-axis direction cosine of the particle
     *    v ,    ! Input:  y-axis direction cosine of the particle
     *    s      ! Output: Distance to the surface (if hit)

      real
     *     A,     ! Internal: Quadratic coefficient A
     *     B,     ! Internal: Quadratic coefficient B
     *     C      ! Internal: Quadratic coefficient C

      A = u**2  + v**2
      B = u*x0  + v*y0
      C = x0**2 + y0**2 - R**2

      call quadric(A,B,C,s,inside,hit) ! Get the generic quadric solution

      return
      end
```

10.3.4 Circular Cones

In standard quadric form, the general equation for a cone is:

$$\cos^2\Theta\{(\vec{x}-\vec{P})-\vec{U}[(\vec{x}-\vec{P})\cdot\vec{U}]\}^2-\sin^2\Theta[(\vec{x}-\vec{P})\cdot\vec{U}]^2=0 \qquad (10.13)$$

where $\vec{P} \equiv (P_x, P_y, P_z)$ is the vertex point of the cone, $\vec{U}$ is the direction vector of the symmetry axis of the cone and Θ is its opening angle. This form, depicted in Figure 10.2, is actually two cones on the same axis situated point-to-point. To avoid ambiguities, we adopt the convention that $0 < \Theta < \pi/2$ and use $\vec{U}$ to orient the cone. (The special case, $\Theta = \pi/2$, corresponds to the quadric surface for coincident planes, while the special case, $\Theta = 0$ corresponds to a zero-radius cylinder.) Both cones are to be regarded as valid surfaces for which the intercept distance is to be calculated. If an application requires only one cone, then it will be assumed that the other "reflection" cone has been eliminated through the use of another surface that isolates only one of the cones.

Intercept distance

Substituting the equation for particle trajectory, eq. 10.1, into the above yields a quadratic equation of the form $As^2 + 2Bs + C = 0$, where the quadratic constants, A, B and C, are:

$$\begin{aligned} A &= \cos^2\Theta[\vec{\mu} - \vec{U}(\vec{\mu}\cdot\vec{U})]^2 - \sin^2\Theta(\vec{\mu}\cdot\vec{U})^2 \\ B &= \cos^2\Theta\vec{\mu}\cdot\{(\vec{p}-\vec{P}) - \vec{U}[(\vec{p}-\vec{P})\cdot\vec{U}] - \sin^2\Theta\vec{U}[(\vec{p}-\vec{P})\cdot\vec{U}]\} \\ C &= \cos^2\Theta\{(\vec{p}-\vec{P}) - \vec{U}[(\vec{p}-\vec{P})\cdot\vec{U}]\}^2 - \sin^2\Theta[(\vec{p}-\vec{P})\cdot\vec{U}]^2 \end{aligned} \tag{10.14}$$

These constants may be employed in the general quadric surface forward-distance solver algorithm described previously.

10.4 Using surfaces to make objects

Now that we know everything about surfaces, it is time to put them together to make objects. That is, we want to develop techniques to delineate regions of space and use them to define the geometrical elements that constitute some object in a Monte Carlo application.

10.4.1 Elemental volumes

We define first the concept of an *elemental* volume, a region of space that can be specified *uniquely* by a set of logical conditions related to being inside or outside the constituent surfaces.

Here are some examples:

A single plane $z = 0$

The gradient of $f(z) = z = 0$ is $\nabla f(z) = (\partial z/\partial z)\hat{z} = \hat{z}$. So, the normal to this surface points along the positive z axis. Therefore, using our definition, all points in space with

$z < 0$ are “inside” and all points in space with $z > 0$ are “outside”. (One immediately sees how arbitrary this definition is!).

What if you were asked to locate the point $(x_0, y_0, 0)$? Based on its position, this point is neither inside nor outside. To answer this question you would require more information, the particle direction, $\vec{\mu} = (u, v, w)$. You would base your decision on where the particle is going. If the particle is on the surface, form the product $\vec{\mu} \cdot \hat{n}$ where n is the normal to the plane. In this case $\vec{\mu} \cdot \hat{n} = w$. If $\vec{\mu} \cdot \hat{n} > 0$, it means that the particle is directed outside. If $\vec{\mu} \cdot \hat{n} < 0$, it means that the particle is directed inside. This would place the particle based on where it is going. What if $z = 0$ and $\vec{\mu} \cdot \hat{n} = 0$, or $w = 0$ in our case? This means that the particle is on the plane and has a trajectory that is parallel to the plane. In this case, the choice is arbitrary. Unless there is a special source of particles that specifically chooses this set of conditions (and would specify the logical location of the particle), the probability that a particle will transport and scatter into this condition is quite small. In this case, choose either “inside” or “outside”. Eventually the particle will scatter either inside or outside the plane and its position will be resolved at that point.

A single sphere of radius R centered at $(0, 0, 0)$

The equation of this surface is:

$$\vec{x}^2 - R^2 = x^2 + y^2 + z^2 - R^2 = 0 \; . \tag{10.15}$$

Given a particle at position $\vec{x}_0 = (x_0, y_0, z_0)$ and direction $\vec{\mu} = (u, v, w)$, the quadratic constants, A, B and C, are:

$$\begin{aligned} A &= 1 \\ B &= \vec{\mu} \cdot \vec{x}_0 \\ &= ux_0 + vy_0 + wz_0 \\ C &= \vec{x}_0^2 - R^2 \\ &= x_0^2 + y_0^2 + z_0^2 - R^2 \end{aligned} \tag{10.16}$$

The sphere delineates two elemental volumes, the interior of the sphere and the exterior of the sphere as shown in Figure 10.4.

If you were asked to locate the particle, first you would look at $C = x_0^2 + y_0^2 + z_0^2 - R^2$. If $C < 0$ the particle is inside the sphere and if the $C > 0$ the particle is outside the sphere. (In this example, the choice of “inside” and “outside” seems a little more natural.) If $C = 0$ the location of the particle is on the sphere and we would then look at the constant B. If $C = 0$ and $B < 0$, the particle is headed inside. If $C = 0$ and $B > 0$, the particle is headed outside. If $C = 0$ and $B = 0$ the particle is on the surface and has a trajectory that is tangent to the sphere. In this case we would appeal to the constant A. For a sphere is always has the same value, 1. It is positive, which means the surface is curving away from the particle trajectory and the particle is headed outside.

A single circular cylinder of radius R centered and directed along the z axis

The equation of this surface is:

$$x^2 + y^2 - R^2 = 0 \ . \tag{10.17}$$

Given a particle at position $\vec{x}_0 = (x_0, y_0, z_0)$ and direction $\vec{\mu} = (u, v, w)$, the quadratic constants, A, B and C, are:

$$\begin{aligned} A &= u^2 + v^2 \\ B &= ux_0 + vy_0 \\ C &= x_0^2 + y_0^2 - R^2 \end{aligned} \tag{10.18}$$

The cylinder delineates two elemental volumes, the interior of the cylinder and the exterior of the cylinder as shown in Figure 10.5.

If you were asked to locate the particle, first you would look at $C = x_0^2 + y_0^2 - R^2$. If $C < 0$ the particle is inside the cylinder and if the $C > 0$ the particle is outside the cylinder. (In this example, the choice of "inside" and "outside" also seems a little more natural.) If $C = 0$ the location of the particle is on the cylinder and we would then look at the constant B. If $C = 0$ and $B < 0$, the particle is headed inside. If $C = 0$ and $B > 0$, the particle is headed outside. If $C = 0$ and $B = 0$ the particle is on the surface and has a trajectory that is tangent to the cylinder. in this case we would appeal to the constant A. For a cylinder this can be zero if the particle's direction is identical to the axis of the cylinder. In this case, the choice is again arbitrary. Unless there is a special source of particles that specifically chooses this set of conditions (and would specify the logical location of the particle), the probability that a particle will transport and scatter into this condition is quite small. In this case, choose either "inside" or "outside". Eventually the particle will scatter either inside or outside the cylinder and its position will be resolved at that point.

An elemental volume made up of several surfaces

For this example, let us consider the planes $P_1 : z = 0$, $P_2 : z = 1$ and the cylinder $C_1 : x^2 + y^2 - R^2 = 0$. The geometry is depicted in Figure 10.6.

The first thing to notice (if you have not already) is that planes and cylinders are infinite and all of space is "carved up" by a set of surfaces. In fact this arrangement delineates 6 elemental volumes! With the notation S_n means outside of surface S_n, we can delineate the 6 elemental volumes as:

Region	Location	Description
1	$\overline{P}_1 \cap C_1 \cap P_2$	outside P_1, inside C_1, inside P_2
2	$P_1 \cap C_1$	inside P_1, inside C_1
3	$C_1 \cap \overline{P}_2$	inside C_1, outside P_2
4	$P_1 \cap \overline{C}_1$	inside P_1, outside C_1
5	$\overline{P}_1 \cap \overline{C}_1 \cap P_2$	outside P_1, outside C_1, inside P_2
6	$\overline{C}_1 \cap \overline{P}_2$	outside C_1, outside P_2

The location of a particle within one of these elemental volumes can be determined uniquely by finding which of these conditions is satisfied.

General considerations for elemental volumes

The first thing to notice that each elemental volume defines a *unique region of space* that can be specified by a *unique set of logical conditions.* Tracking particles through elemental volumes is extremely fast. First of all, the positioning errors mentioned earlier in this chapter cause no problem and are absorbed on subsequent transport steps. Correction schemes never have to be invoked. Particles that become "lost", that is, lose or never acquire a sense of which elemental volume they are in, can be found using the location techniques just described.[5] Finally, if a particle leaves an elemental volume by one of its surfaces and this surface is employed by another elemental volume, its next elemental volume is known by flipping the logical switch that orients the particle with respect to that surface.

10.5 Tracking in an elemental volume

We now consider the case where we are tracking a particle within an elemental volume. There are some nice consequences of tracking within elemental volumes. The first is that any given surface defining the elemental volume can only be intercepted once. Otherwise the "insideness" or the "outsidesness" would not be unique. The other consequence is that the intercept is defined as the shortest intercept to any surface bounding the elemental volume without consideration of location of the intercept. Recall that a surface extends to the exterior of any elemental volume because surfaces are infinite (except for ellipsoids) and volumes are (usually) finite. The consequence of this is that the shortest intercept with any surface is guaranteed to be in the elemental surface volume.

This is best demonstrated by example. Consider the interior of the right circular cylinder

[5] If each surface of each elemental volume joins uniquely onto other elemental volume, it can never become "lost" through tracking. Source particles may be "lost" before tracking starts and will have to acquire its location in an elemental volume by searching. Sometimes the surface of an elemental volumes join onto two or more other elemental volumes. A search algorithm may have to be initiated at this point to locate the particle uniquely in an elemental volume. Another difficulty occurs if "voids" are created by poorly constructed geometry code. Good coding defines all of space uniquely.

defined by the planes $P_1 : z = 0$, $P_2 : z = 1$ and the cylinder $C_1 : x^2 + y^2 - R^2 = 0$. The geometry is depicted in Figure 10.7 and some representative trajectories are given there.

Consider trajectory 1. It has intercepts with both C_1 and P_2. However, the distance to the intercept with C_1 is shorter and is the answer in this case. Contrast this with trajectory 2. It has intercepts with both C_1 and P_2. However, the distance to the intercept with P_2 is shorter and is the answer in this case.

Recall that the interior of the right circular cylinder is represented by the condition $\overline{P}_1 \cap C_1 \cap P_2$. Therefore, the distance to exit the elemental volume is given by the following coding:

```
TransportDistance = infinity ! (Some very large number)

! Check the distance to each bounding surface

! Check the leftmost plane
call zplane(0e0,z0,w,s,.false.,hit) ! 0e0 = zplane position, .false. => outside
if (hit .and. (s .le. TransportDistance)) TransportDistance = s

! Check the rightmost plane
call zplane(1e0,z0,w,s,.true.,hit) ! 1e0 = zplane position, .true. => inside
if (hit .and. (s .le. TransportDistance)) TransportDistance = s

! Check the cylinder
call ccylz(1e0,x0,y0,u,v,s,.true.,hit) ! 1e0 = radius, .true. => inside
if (hit .and. (s .le. TransportDistance)) TransportDistance = s
```

After this code segment is executed, the variable `TransportDistance` is the distance to exit the elemental volume by any surface irrespective of its direction, as long as it is logically placed within this elemental volume.

The equivalent coding for a general elemental volume would be:

```
TransportDistance = infinity ! (Some very large number)

! Check the distance to each bounding surface

! {...} are the parameters that define the particle’s position and direction
! [...] are the parameters that define the surface
! [.true.|.false.] true or false depending upon orientation of surface
```

```
! hit is always returned .true. or .false.
! If hit is returned as .true., s represents the distance to that surface

call Surface_1({...},[...],s,[.true.|.false.],hit)
if (hit .and. (s .le. TransportDistance)) TransportDistance = s

call Surface_2({...},[...],s,[.true.|.false.],hit)
if (hit .and. (s .le. TransportDistance)) TransportDistance = s

call Surface_3({...},[...],s,[.true.|.false.],hit)
if (hit .and. (s .le. TransportDistance)) TransportDistance = s

.
.
.

! Until all the bounding surfaces are exhausted
```

After this code segment is executed, the variable `TransportDistance` is the distance to exit the elemental volume by any surface irrespective of its direction, as long as it is logically placed within this elemental volume.

10.6 Using elemental volumes to make objects

10.6.1 Simply-connected elements

A simply-connected object is one in which one element connects to only one other element through a common surface. It is best to explain this by means of an example.

Consider the simple problem of creating the object depicted in Figure 10.8. This object is made up of the planes $P_1 : z = 0$, $P_2 : z = 1$ and the cylinder $C_1 : x^2 + y^2 - R^2 = 0$ and the sphere $S_1 : x^2 + y^2 + z^2 - R^2 = 0$. We will define three regions:

Region	Location	Description
1	$\overline{P}_1 \cap C_1 \cap P_2$	outside P_1, inside C_1, inside P_2
2	$P_1 \cap S_1$	inside P_1, inside S_1
0	All the rest	This is the “outside” of the object

We will adopt the convention that region “0” is the exterior of the object. By exterior we mean that if a particle is tracked from the interior to the exterior it is considered to “vanish” from the simulation.

Note that the way to define this object is not unique! Equivalently we may define it as follows:

Region	Location	Description
1	$\overline{S}_1 \cap C_1 \cap P_2$	outside S_1, inside C_1, inside P_2
2	S_1	inside S_1
0	All the rest	This is the "outside" of the object

However, we will employ the first definition in our example. We note that we do require two elemental volumes to define this object. The reason for this is that the interior space of the object is made up of regions that are both "inside" and "outside" the sphere. The division into two elemental surfaces can be done in a number of ways.

The logic to handle tracking in the object is to realize that if the particle is inside region 1 and tracked to either C_1 or P_2, it escapes on the next step. If the particle is inside region 2 and tracked to either S_1 or P_1, it escapes on the next step. If the particle is inside region 1 and tracked to P_1 it enters region 2 on the next step. Conversely, if the particle is inside region 2 and tracked to P_1 it enters region 1 on the next step.

We will now write the "pseudo-code" for this geometry. However, we must now extend the particle phase-space concept introduced in Chapter 7. In that chapter, in Equation 7.1, a particle's phase-space was defined as:

$$\{\vec{x}, \vec{u}\} \ , \tag{10.19}$$

where $\vec{x}$ is the particle's absolute location in space referred back to the origin of some laboratory coordinate system, and $\vec{u}$ is its direction referred back to a fixed set of axes in the same laboratory coordinate system. The phase-space is now expanded to:

$$\{\vec{x}, \vec{u}, N_{\text{elem}}\} \ , \tag{10.20}$$

where N_{elem} is the elemental volume number associated with its current position.

The pseudo-code for this geometry is:

```
IF $N_{\text{elem}} = 0$, stop transport, particle vanishes.

ELSEIF $N_{\text{elem}} = 1$

    IF particle hits $P_1$, $N_{\text{elem}} = 2$, next region is region 2
    ELSEIF particle hits $P_2$, $N_{\text{elem}} = 0$, next region is the outside
    ELSEIF particle hits $C_1$, $N_{\text{elem}} = 0$, next region is the outside
    ELSE No surface is hit, continue transport

ELSEIF $N_{\text{elem}} = 2$
```

```
    **IF** particle hits $P_1$, $N_{\text{elem}} = 1$, next region is region 1
    **ELSEIF** particle hits $S_1$, $N_{\text{elem}} = 0$, next region is the outside
    **ELSE** No surface is hit, continue transport
```

Now, let us write the Fortran code for this application:

```
.
.
.

! Determine the distance to a scattering point
TransportDistance = DistanceToScatteringPoint()

NextElement = Nelem ! Assume particle stays in the same region

if (Nelem .eq. 0) then
    ! Particle escapes
    ! Special coding or a logic transfer point must be specified to
    ! give us some particle phase-space to work with

elseif (Nelem eq.1) then

    ! In the first region

    ! Check the distance to each bounding surface of region 1

    ! Check plane 1 at z = 0
    call zplane(0e0,z0,w,s,.false.,hit) ! .false. => outside
    if (hit .and. (s .le. TransportDistance)) then
        TransportDistance = s
        NextElement = 2
    end if

    ! Check plane 2 at z = 1
    call zplane(1e0,z0,w,s,.true.,hit) ! .true. => inside
    if (hit .and. (s .le. TransportDistance)) then
        TransportDistance = s
        NextElement = 0
    end if
    ! Check the cylinder
    call ccylz(1e0,x0,y0,u,v,s,.true.,hit) ! 1e0 = radius, .true. => inside
    if (hit .and. (s .le. TransportDistance))
        TransportDistance = s
        NextElement = 0
    end if

elseif (Nelem .eq. 2) then

    ! In the second region

    ! Check the distance to each bounding surface of region 2

    ! Check plane 1 at z = 0
    call zplane(0e0,z0,w,s,.false.,hit) ! .true. => inside
    if (hit .and. (s .le. TransportDistance)) then
        TransportDistance = s
        NextElement = 1
    end if

    ! Check the sphere
    call csphere(1e0,x0,y0,z0,u,v,w,s,.true.,hit) ! 1e0 = radius, .true. => inside
    if (hit .and. (s .le. TransportDistance))
        TransportDistance = s
        NextElement = 0
    end if

endif

! Transport the particle

x0 = x0 + u*s
y0 = y0 + v*s
z0 = z0 + w*s

! Change region numbers

Nelem = NextRegion

.
.
.
```

Note that if the surface is hit, it must also shorten the transport distance in order to serve as a candidate for the surface through which a particle leaves the elemental volume.

The generalization to more complex objects is clear. With the knowledge of which elemental volume the particle is in cycle over all the bounding surfaces of the element. If the particle strikes this surface along its flight path and if this distance shortens the proposed transport distance, then the next element can be specified. It can also be superseded by a shorter intersection to another surface. *The shortest one is the one that matters!*

10.6.2 Multiply-connected elements

A multiply-connected object is one in which one element can connect to more than one element through one or more common surface. It is best to explain this by means of an example as well.

Consider the problem of creating the object depicted in Figure 10.9. This object is made up of two planes $P_1 : x = 0$, $P_2 : y = 0$. We will define three regions:

Region	Location	Description
1	P_1	inside P_1
2	$\overline{P}_1 \cap P_2$	outside P_1, inside P_2
3	$\overline{P}_1 \cap \overline{P}_2$	outside P_1, outside P_2

Regions 2 and 3 are simply connected to each other an region 1, but region 1 is multiply-connected to regions 2 and 3 through the common surface P_1.

In objects that are not too complex one can do a fake transport of the particle and see in which region its terminal position resides in. In this case we would test to see whether `y0 + v*s` os greater or less than zero to make a decision. Each case requires special coding and techniques to resolve. If one can turn a multiply-connected object into an simply-connected one, it often generates faster code at the expense of introducing more regions into the problem.

A general strategy for resolving this problem would take us beyond the scope of this course. A basic solution that is applied in some applications is first to do a search for position in all the candidate elements. If the search comes up empty, it means that the particle did not escape its current region by virtue of an undershoot that that surface. One can reset the region back to the original and try again. If the subsequent transport still does not resolve the problem the solution is to provide small boosts to the transport step starting with the smallest resolvable floating-point number and allow these boosts to grow geometrically until the surface is crosses, at least numerically.

It is ugly but it works and is fast. Unlike the simply-connected surfaces where some elegant logic solves the problem, the logic for multiply-connected surfaces gets quite involved and is usually not worth the effort in terms of code efficiency.

10.6.3 Combinatorial geometry

So far we have introduced elemental volumes and indicated how they may be combined to make objects. We have also introduced a simple yet powerful tracking algorithm for simply-connected surfaces and have given some idea as to the subtleties involved in multiply-connected elements.

The general discussion of this is called *combinatorial geometry* and would take us beyond the scope of this book. Before dismissing the topic, however, the following capabilities of combinatorial codes are either necessary or desirable:

1. The ability for a particle to locate itself unambiguously in a unique elemental volume based upon its position and direction.
2. A systematic numbering/identification scheme for the elemental volumes. For small applications this can be coded by the user. For involved applications, the combinatorial code should be able to do this on its own.
3. The ability to model reflecting, absorbing or partially absorbing surfaces.
4. The ability to "build" objects in standard position and then translate and rotate them into actual position.
5. The ability to re-use objects, once, twice, or in repetitive patterns.
6. The ability to model various sources of particles, either originating interior or exterior to the geometry.
7. The ability to see the results of geometry construction graphically.
8. A graphical tool to build geometries.

10.7 Law of reflection

The use of some of the inherent symmetry of the geometry can realize some real simplifications. We will consider the use of reflecting planes to mimic some of the inherent symmetry of a geometry.

For example, consider the geometry depicted in fig. 10.10. In this case, an infinite square lattice of cylinders is irradiated uniformly from the top. The cylinders are all uniform and aligned. How should one approach this problem? Clearly, one can not model an infinite array of cylinders. If one tried, one would have to pick a finite set and decide somehow that it was big enough. Instead, it is much more efficient to exploit the symmetry of the problem. It turns out that in this instance, one needs to transport particles in only 1/8'th

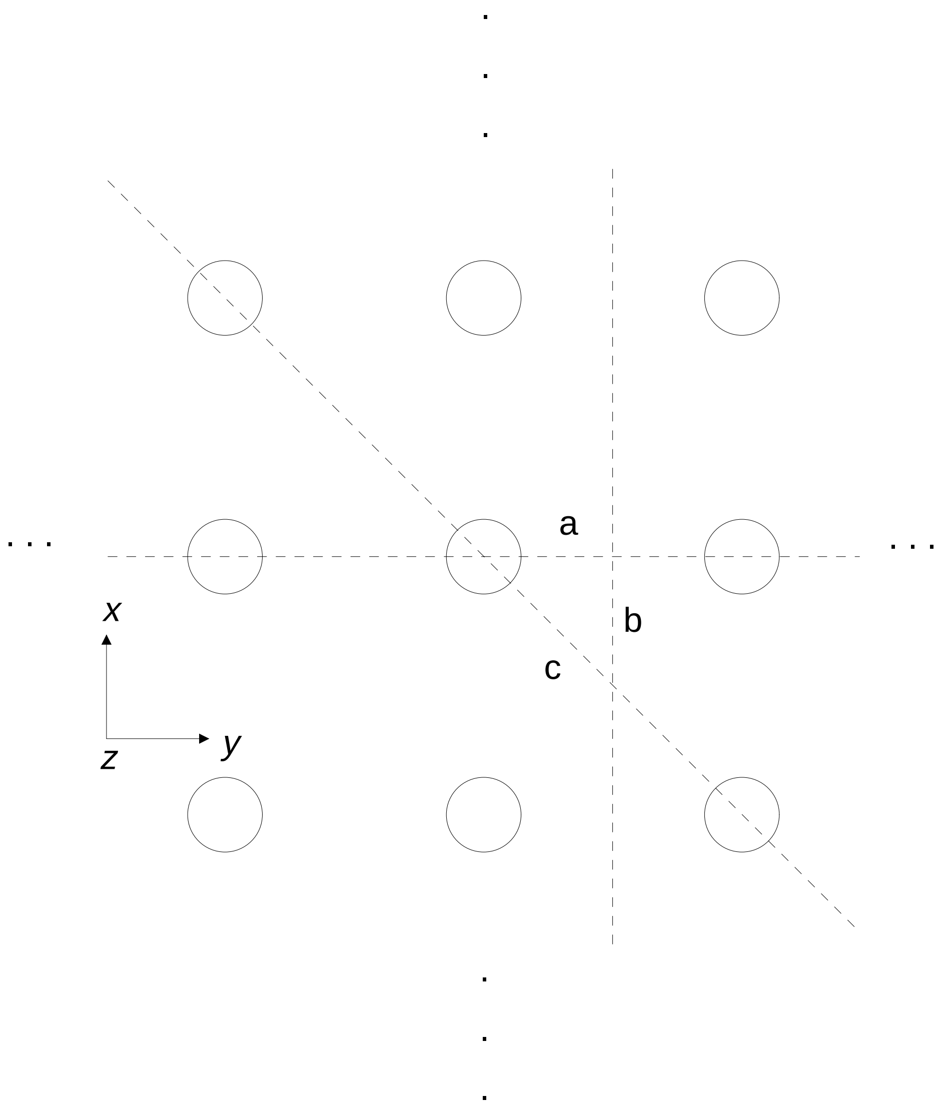

Figure 10.10: Top end view of an infinite square lattice of cylinders. Three planes of symmetry are drawn, **a**, **b**, and **c**. A complete simulation of the entire lattice may be performed by restricting the transport to the interior of the three planes. When a particle strikes a plane it is reflected back in, thereby mimicking the symmetry associated with this plane.

of a cylinder! To see this we find the symmetries in this problem. In fig. 10.10 we have drawn three planes of symmetry in the problem, planes **a**, **b**, and **c**[6]. There is reflection symmetry for each of these planes. Therefore, to mimic the infinite lattice, any particles that strike these reflecting planes should be reflected. One only needs to transport particles in the region bounded by the reflecting planes. Because of the highly symmetric nature of the problem, we only need to perform the simulation in a portion of the cylinder and the "response" functions for the rest of the lattice is found by reflection.

The rule for particle reflection about a plane of arbitrary orientation is easy to derive. Let $\vec{u}$ be the unit direction vector of a particle and $\hat{n}$ be the unit direction normal of the reflecting plane. Now divide the particle's direction vector into two portions, $\vec{u}_{\|}$, parallel to $\hat{n}$, and $\vec{u}_{\perp}$, perpendicular to $\vec{n}$. The parallel part gets reflected, $\vec{u}'_{\|} = -\vec{u}_{\|}$, and the perpendicular part remains unchanged, $\vec{u}'_{\perp} = \vec{u}_{\perp}$. That is, the new direction vector is $\vec{u}' = -\vec{u}_{\|} + \vec{u}_{\perp}$. Another way of writing this is,

$$\vec{u}' = \vec{u} - 2(\vec{u} \cdot \vec{n})\vec{n} \ . \tag{10.21}$$

Applying eq. 10.21 to the problem in fig. 10.10, we have: For reflection at plane **a**, $(u'_x, u'_y, u'_z) = (-u_x, u_y, u_z)$. For reflection at plane **b**, $(u'_x, u'_y, u'_z) = (u_x, -u_y, u_z)$. For reflection at plane **c**, $(u'_x, u'_y, u'_z) = (-u_y, -u_x, u_z)$. The use of this reflection technique can result in great gains in efficiency. Most practical problems will not enjoy such a great amount of symmetry but one is encouraged to make use of any available symmetry. The saving in computing time is well worth the extra care and coding.

[6] Note that this symmetry applies only to a square lattice, where the spacing is the same for the x and y-axes. For a rectangular symmetry, the planes of reflection would be somewhat different. There would be no plane **c** as for the square lattice in fig. 10.10.

Bibliography

[Olm47] J. M. H. Olmsted. Solid Analytic Geometry. *(Appleton-Century-Crofts Inc, New York)*, 1947.

Problems

1. Consider a right circular cylinder of radius R and length L where R and L are in units of, say, cm's. The interior of this object is defined by the planes $z = 0$, $z = L$ and the cylinder $x^2 + y^2 = R^2$.

   A beam of at least 10^4 particles is incident on the middle of the flat end of the geometry at $\vec{x} = 0$ with unit direction vector $\vec{u} = \hat{z}$. These particles exhibit discrete interactions with an attenuation factor of $\Sigma = 1$ cm^{-1}. Thus, the interaction probability distribution function at location z is $p(z)\mathrm{d}z = \Sigma \exp(-\Sigma z)\mathrm{d}z$. These particles scatter isotropically once they reach the scattering point.

   Write a Monte Carlo code to simulate this problem and answer the following questions:

   (a) The total pathlength/history is the sum of the distance from start to interaction (if it occurs) plus the drift distance to exit the geometry. For $R = 0$ and $L = \infty$, what is the average total pathlength/history and its associated estimated error, $\bar{t} \pm s_{\bar{t}}$?

   (b) For $L = 10$ cm, what is the average pathlength and its associated estimated error, $\bar{t} \pm s_{\bar{t}}$ when R is 0.1,0.2,0.5,1,2,5,10,20,50,100,200,500,1000 cm's? Plot the results and discuss the small-R and large-R limits.

   (c) For $R = 10$ cm, what is the average pathlength and its associated estimated error, $\bar{t} \pm s_{\bar{t}}$ when L is 0.1,0.2,0.5,1,2,5,10,20,50,100,200,500,1000 cm's? Plot the results and discuss the small-L and large-L limits.

2. Consider a right circular cylinder of radius R and length L. The interior of this object is defined by the planes $z = 0$, $z = L$ and the cylinder $x^2 + y^2 = R^2$. For this exercise, $R = L = 1$ cm. A beam of at least 10^4 particles is incident on the middle of the flat end of the geometry at $\vec{x} = 0$ with unit direction vector $\vec{u} = \hat{z}$. These particles

exhibit discrete interactions with an attenuation factor Σ which will be given later in units of cm^{-1}'s. Thus, the interaction probability distribution function at location z is $p(z)\ \mathrm{d}z = \Sigma \exp(-\Sigma z)\ \mathrm{d}z$. At the interaction point, the particle scatters into a polar, Θ, and an azimuthal, Φ, angle with probability distribution:

$$p(\Theta,\Phi)\,\mathrm{d}\Theta\ \mathrm{d}\Phi = \frac{\eta(2+\eta)}{4\pi}\frac{\sin\Theta\ \mathrm{d}\Theta\ \mathrm{d}\Phi}{(1-\cos\Theta+\eta)^2}\ ,$$

where η is a dimensionless constant that describes forward scattering for $\eta \longrightarrow 0$ and isotropic scattering for $\eta \longrightarrow \infty$. After the first interaction, second and subsequent interactions can occur with the same interaction probability until the particle escapes from the geometry, possibly having scattered repeatedly.

Write a Monte Carlo code to simulate this problem and answer the following associated questions:

(a) i. Show that in the limit, $\eta \longrightarrow 0$, $p(\Theta)\sin\Theta\ \mathrm{d}\Theta\ \mathrm{d}\Phi \longrightarrow \delta(1{-}\cos\Theta)\sin\Theta\ \mathrm{d}\Theta\ \mathrm{d}\Phi/2\pi$.

ii. Show that in the limit, $\eta \longrightarrow \infty$, $p(\Theta)\sin\Theta\ \mathrm{d}\Theta\ \mathrm{d}\Phi \longrightarrow \sin\Theta\ \mathrm{d}\Theta\ \mathrm{d}\Phi/4\pi$.

(b) For the set of Σ's, $\Sigma = 0.1$, 1, 10 cm^{-1}'s and both forward ($\eta \longrightarrow 0$) and isotropic ($\eta \longrightarrow \infty$) scattering, calculate the backscatter and transmission coefficients (number backscattered or transmitted divided by the number of histories), and the associated estimated statistical uncertainty.

(c) For $\Sigma = 10$ cm^{-1}, do simulations for a set of η's with enough points such that both asymptotic limits are nearly reached and the η behavior of the backscatter and transmission coefficients for intermediate η are fully described.

(d) Repeat 2(b) but with the back plane of the geometry being a reflective surface.

(e) Repeat 2(b) but with both the back plane and the cylindrical surface of the geometry reflective but with absorption included in the following way—at each interaction point, there is a 0.8 probability that the particle will scatter in the way previously described and a 0.2 probability that it will be absorbed.

(f) Repeat 2(b) but with grazing incidence of the starting point, *i.e.* the particles are incident on the middle of the flat end of the geometry at $\vec{x} = 0$ with unit direction vector $\vec{u} = \hat{y}$.

3. A cylindrical beam of particles with the following starting characteristics:

$$\psi(\vec{x},\vec{u}) = \frac{2}{R_{\mathrm{beam}}^2}\Theta(R_{\mathrm{beam}}^2 - x^2 - y^2)\delta(z)\delta(\hat{z}-\vec{u})$$

is incident on a geometry defined by the plane, $z = 0$ and the cylinder $x^2 + y^2 = R_{\mathrm{cyl}}^2$. Once inside the target (*i.e.* $z \geq 0$), the particle scatters isotropically with elastic scattering cross section, $\Sigma_s = 1$ and absorption cross section $\Sigma_a = 0.25$. Outside the target (*i.e.* $z < 0$), $\Sigma_s = \Sigma_a = 0$. *Note to take care that you inform the particles whether they are inside the cylinder or outside.*

(a) Let $R_{\mathrm{cyl}} = 1$. Tally and plot the average path-length/R^2_{cyl} (per particle) inside the cylinder and its estimated error as a function of $0.1 \leq R_{\mathrm{beam}} \leq 10$ using 10000 incident particle histories.

(b) Now let $R_{\mathrm{beam}} = 1$. Tally and plot the average path-length/R^2_{cyl} (per particle) inside the cylinder and its estimated error as a function of $0.1 \leq R_{\mathrm{cyl}} \leq 10$ using 10000 incident particle histories.

(c) Compare both the average values and the estimated errors of the above two examples. What conclusions do you make? Which is the more efficient way to solve the problem?

4. A geometry is defined by the infinite square lattice of cylinders

$$\sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} (x - di)^2 + (y - dj)^2 = R^2_{\mathrm{cyl}},$$

where i and j are integers and d is the lattice spacing constant. 10000 particles start out from the origin with an isotropically symmetric direction. Inside the cylinders there is only absorption, with an absorption constant Σ_a. Outside the cylinders there is only isotropic elastic scattering, with an interaction constant Σ_s. Starting with default values of $R_{\mathrm{cyl}} = 0.25$, $\Sigma_s = 1$, $\Sigma_a = 1$, tally and plot the average square cylindrical radius distance $\overline{x^2 + y^2}$ (and its estimated error) where the particle is absorbed as a function of the following geometrical variables:

(a) d, taking note that $d < 0.5$

(b) Σ_s

(c) Σ_a

5. The particle interaction scheme we will consider is that of either isotropic or forward scattering with a scattering constant, Σ_{scat}, and particle absorption with the constant, Σ_{abs}. In this example, $\Sigma_{\mathrm{scat}} = 1$ cm^{-1} and $\Sigma_{\mathrm{abs}} = 0.05$ cm^{-1}. The particles are incident normally on a planar geometry consisting of 21 planes normal to the z-axis separated by 1 cm. That is, $z = 0, 1, 2, ...20$ cm. Tally the average pathlength in each planar zone. Once within the geometry, if it hits the plane at $z = 0$ or $z = 20$, it escapes.

A working version of transport portion of the code is attached. The difficulty in doing this exercise is writing an efficient `subroutine geometry`. Use the `subroutine zplane()` from the code library to solve this problem. (It is possible to do it in about 15 lines of executable code.)

Plot the results for isotropic and forward scattering. Compare and explain the results. Hand in only your plot(s), your `subroutine geometry` and associated discussion.

```
      ! Starting position
      x = 0
      y = 0
      z = 0

      ! Starting direction
      u = 0
      v = 0
      w = 1

      iregion = 1 ! Starts incident on plane 1

1     continue ! Beginning of the transport loop

      t = -log(1 - rng())/(Sigma_scat + Sigma_abs)

      call geometry
     *(
     *    t,          ! On input, this is the proposed transport distance
     *                ! On output, if hit = .true., t is set to the
                      ! distance to the plane which it hits
     *    hit,        ! Set to .true. if the input distance t is greater
                      ! than the transport distance to a plane
     *    iregion,    ! Input: the current region number
     *    new_region  ! The new region number.
                      ! If new_region is set to 0 then is escape from either
                      ! the front or the back of the geometry
     *)

      ! Transport the particle
      x = x + u*t
      y = y + v*t
      z = z + w*t

      path(iregion)  = path(iregion)  + t    ! Tally the pathlength in region
                                             ! iregion
      path2(iregion) = path2(iregion) + t**2 ! Tally its square for statistics

      if (hit) then
          ! Particle moves out of the current region
          if (new_region .eq. 0) then
              ! It has escaped the geometry, do the next particle
```

```
        continue
    else
        iregion = new_region ! It has changed its region number
        goto 1 ! Continue transport in the new region
    end if
else
    if (rng() .lt. Sigma_scat/(Sigma_scat + Sigma_abs)) then
        ! Particle interacts isotropically
        ! For forward scattering, comment out the next two lines
        call isotropic(costhe)
        call rotate(u,v,w,costhe)
        goto 1 ! Continue transport in the same region
    else
        ! Particle is absorbed, do the next particle
        continue
    end if
end if
```

6. A geometry consists of a unit cube at the center of which is a sphere with diameter D, where $D \leq 1$. The planar surfaces are reflecting with an efficiency of $\epsilon = 0.95$. That means, if a particle strikes a planar surface, it has $p = 0.95$ of being reflected and $p = 0.05$ of being absorbed. The sphere, on the other hand, is totally absorbing. That is, if a particle strikes the sphere, it has $p = 1$ of being absorbed. The source for this is distributed uniformly throughout the box and radiates isotropically. The interior of the box is vacuum.

   Write a Monte Carlo code to simulate this problem and tally the number of reflections a particle experiences before it is absorbed as a function of the diameter of the sphere. Tally the number of particles absorbed by the sphere *vs.* the number of particles absorbed by the planes as a function of the diameter of the sphere.

   Use your discretion as to how many histories to execute to get a reasonable result and how fine a grid spacing in a to produce reasonable results.

Chapter 11

Monte Carlo and Numerical Quadrature

In this chapter we present a mathematical proof that the Monte Carlo method is the most efficient way of estimating tallies in 3 spatial dimensions when compared to first-order deterministic (analytic, phase-space evolution) methods. Notwithstanding the opinion that the Monte Carlo method is thought of as providing the most accurate calculation, the argument may be may be made in such a way that is independent of the physics content of the underlying algorithm or the quality of the incident radiation field.

11.1 The dimensionality of deterministic methods

For the purposes of estimating tallies from initiating electrons, photons, or neutrons, the transport process that describes the trajectories of particles is adequately described by the linear Boltzmann transport equation [DM79]:

$$\left[\frac{\partial}{\partial s} + \frac{p}{|p|} \cdot \frac{\partial}{\partial x} + \mu(x,p)\right] \psi(x,p,s) = \int \mathrm{d}p' \, \mu(x,p,p')\psi(x',p',s) \ , \tag{11.1}$$

where x is the position, p is the momentum of the particle, $(p/|p|) \cdot \partial/\partial x$ is a directional derivative (in three dimensions $\vec{\Omega} \cdot \vec{\nabla}$, for example) and s is a measure of the particle pathlength. We use the notation that x and p are multi-dimensional variables of dimensionality N_x and N_p. Conventional applications span the range $1 \leq N_{p,x} \leq 3$. The macroscopic differential scattering cross section (probability per unit length) $\mu(x,p,p')$ describes scattering from momentum p' to p at location x, and the total macroscopic cross section is defined by:

$$\mu(x,p) = \int \mathrm{d}p' \, \mu(x,p,p') \ . \tag{11.2}$$

$\psi(x,p,s)\,\mathrm{d}x\,\mathrm{d}p$ is the probability of there being a particle in $\mathrm{d}x$ about x, in $\mathrm{d}p$ about p and at pathlength s. The boundary condition to be applied is:

$$\psi(x,p,0) = \delta(x)\delta(p_0 - p)\delta(s) \ , \tag{11.3}$$

where p_0 represents the starting momentum of a particle at $s = 0$. The essential feature of Equation 11.1 insofar as this proof is concerned, is that the solution involves the computation of a $(N_x + N_p)$-dimensional integral.

A general solution may be stated formally:

$$\psi(x,p,s) = \int \mathrm{d}x' \int \mathrm{d}p'\, G(x,p,x',p',s)Q(x',p') \ , \tag{11.4}$$

where $G(x,p,x',p',s)$ is the Green's function and $Q(x',p')$ is a source. The Green's function encompasses the operations of transport (drift between points of scatter, $x' \to x$), scattering (*i.e.* change in momentum) and energy loss, $p' \to p$. The interpretation of $G(x,p,x',p',s)$ is that it is an operator that moves particles from one point in $(N_x + N_p)$-dimensional phase space, (x',p'), to another, (x,p) and can be computed from the kinematical and scattering laws of physics.

Two forms of Equation 11.4 have been employed extensively for general calculation purposes. Convolution methods integrate Equation 11.4 with respect to pathlength s and further assume (at least for the calculation of the Green's function) that the medium is effectively infinite. Thus,

$$\psi(x,p) = \int \mathrm{d}x' \int \mathrm{d}p'\, G\left(|x-x'|, \left[\frac{p}{|p|}\cdot\frac{p'}{|p'|}\right], |p'|\right) Q(x',p') \ , \tag{11.5}$$

where the Green's function is a function of the distance between the source point x' and x, the angle between the vector defined by the source p' and p and the magnitude of the momentum of the course, $|p'|$, or equivalently, the energy.

To estimate a tally using Equation 11.5 we integrate $\psi(x,p)$ over p, with an response function, $\mathcal{R}(x,p)$ [SF96]:

$$T(x) = \int \mathrm{d}x' \int \mathrm{d}p'\, F(|x-x'|,p')Q(x',p') \ , \tag{11.6}$$

where the "kernel", $F(|x-x'|,p')$, is defined by:

$$F(|x-x'|,p') = \int \mathrm{d}p\, \mathcal{R}(x,p) G\left(|x-x'|, \left[\frac{p}{|p|}\cdot\frac{p'}{|p'|}\right], |p'|\right) \ . \tag{11.7}$$

$F(|x-x'|,p')$ has the interpretation of a functional relationship that connects particle fluence at phase-space location x',p' to a tally calculated at x. This method has a known difficulty—its treatment of heterogeneities and interfaces. Heterogeneities and interfaces can be treated approximately by scaling $|x-x'|$ by the collision density. This is a exact for the part of the kernel that describes the first scatter contribution but approximate for higher-order scatter contributions. It can also be approximate to varying degrees if the scatter produces other particles with different scaling laws, such as the electron set in motion by a first Compton collision of a photon.

For calculation methods that are concerned with primary charged particles, the heterogeneity problem is more severe. The true solution in this case is reached when the pathlength steps, s in Equation 11.4 are made small [Lar92] and so, an iterative scheme is set up:

$$\begin{aligned}
\psi_1(x,p) &= \int \mathrm{d}x' \int \mathrm{d}p'\, G(x,p,x',p',\Delta s) Q(x',p') \\
\psi_2(x,p) &= \int \mathrm{d}x' \int \mathrm{d}p'\, G(x,p,x',p',\Delta s) \psi_1(x',p') \\
\psi_3(x,p) &= \int \mathrm{d}x' \int \mathrm{d}p'\, G(x,p,x',p',\Delta s) \psi_2(x',p') \\
&\cdot \\
&\cdot \\
&\cdot \\
\psi_N(x,p) &= \int \mathrm{d}x' \int \mathrm{d}p'\, G(\psi_{N-1}(x',p') \qquad (11.8)
\end{aligned}$$

which terminates when the largest energy in $\psi_N(x,p)$ has fallen below an energy threshold or there is no x remaining within the target. The picture is of the phase space represented by $\psi(x,p)$ "evolving" as s accumulates. This technique has come to be known as the "phase-space evolution" model. Heterogeneities are accounted for by forcing Δs to be "small" or of the order of the dimensions of the heterogeneities and using a $G()$ that pertains to the atomic composition of the local environment. The calculation is performed in a similar manner described for convolution. That is,

$$T(x) = \sum_{i=1}^{N} \int \mathrm{d}x' \int \mathrm{d}p'\, F(x,x',p,p',\Delta s) \psi_i(x',p') \ , \qquad (11.9)$$

where the "kernel", $F(x,x',p,p',\Delta s)$, is defined by:

$$F(x,x',p,p',\Delta s) = \int \mathrm{d}p\, \mathcal{R}(x,p) G(x,p,x',p',\Delta s)) \ . \qquad (11.10)$$

In the following analysis, we will not consider further any systematic errors associated with the treatment of heterogeneities in the case of the convolution method or with the "stepping errors" associated with incrementing s using Δs in the phase space evolution model. Furthermore, we assume that the Green's functions or response kernels can be computed "exactly"—that there is no systematic error associated with them. The important result of this discussion is to demonstrate that the dimensionality of the analytic approaches is $N_x + N_p$.

11.2 Convergence of Deterministic Solutions

The discussion of the previous section indicates that deterministic solutions are tantamount to solving a $\mathcal{D}$-dimensional integral of the form:

$$I = \int_{\mathcal{D}} \mathrm{d}u\, H(u) \ . \tag{11.11}$$

11.2.1 One dimension

Consider a 1-dimensional problem. We divide the u space into N_{cell} cells. Thus, Equation 11.11 may be re-written exactly as:

$$I = \int_{u_{\min}}^{u_{\max}} \mathrm{d}u\, H(u) = \sum_{i=1}^{N_{\mathrm{cell}}} \int_{u_i-\Delta u/2}^{u_i+\Delta u/2} \mathrm{d}u\, H(u) \ , \tag{11.12}$$

where $\Delta u = (u_{\max} - u_{\min})/N_{\mathrm{cell}}$ and u_i is the mid-point of the i^{th} cell. We assume that we can approximate $H(u)$ in the vicinity of u_i in a Taylor expansion:

$$H(u) = H(u_i) + (u-u_i)\frac{\mathrm{d}H(u)}{\mathrm{d}u}\bigg|_{u=u_i} + \frac{(u-u_i)^2}{2}\frac{\mathrm{d}^2H(u)}{\mathrm{d}u^2}\bigg|_{u=u_i} \cdots \tag{11.13}$$

Substituting Equation 11.13 into Equation 11.12 gives:

$$I = \Delta u \sum_{i=1}^{N_{\mathrm{cell}}} \left(H(u_i) + \frac{(\Delta u)^2}{24}\, \frac{\mathrm{d}^2H(u)}{\mathrm{d}u^2}\bigg|_{u=u_i} \cdots \right) \ . \tag{11.14}$$

The first term in the above summation gives the “first-order” estimate while the second term gives an estimate of its error. The first derivative terms have all vanished by virtue of taking the midpoint of the cell as the point about which to execute the Taylor expansion.

Therefore we write the fractional error as:

$$\frac{\Delta I}{I} = \frac{1}{N_{\mathrm{cell}}^2}\frac{(u_{\max}-u_{\min})^2}{24}\frac{\sum_{i=1}^{N_{\mathrm{cell}}} \mathrm{d}^2H(u_i)/\mathrm{d}u^2}{\sum_{i=1}^{N_{\mathrm{cell}}} H(u_i)} \tag{11.15}$$

where we have employed the more compact notation, $\mathrm{d}^2H(u_i)/\mathrm{d}u^2 = \mathrm{d}^2H(u)/\mathrm{d}u^2|_{u=u_i}$

11.2.2 Two dimensions

In two dimensions, we give each dimension a mesh-size of $N_{\mathrm{cell}}^{1/2}$ giving:

$$I = \int_{u_{1,\min}}^{u_{1,\max}} \mathrm{d}u_1 \int_{u_{2,\min}}^{u_{2,\max}} \mathrm{d}u_2\, H(u_1,u_2) = \sum_{i_1=1}^{N_{\mathrm{cell}}^{1/2}} \sum_{i_2=1}^{N_{\mathrm{cell}}^{1/2}} \int_{u_{1_i}-\Delta u_1/2}^{u_{1_i}+\Delta u_1/2} \mathrm{d}u_1 \int_{u_{2_i}-\Delta u_2/2}^{u_{2_i}+\Delta u_2/2} \mathrm{d}u_2\, H(u_1,u_2) \ , \tag{11.16}$$

where $\Delta u_1 = (u_{1,\max} - u_{1,\min})/N_{\rm cell}^{1/2}$ and $\Delta u_2 = (u_{2,\max} - u_{2,\min})/N_{\rm cell}^{1/2}$. The Taylor expansion of $H(u)$ then takes the form:

$$\begin{aligned} H(u_1,u_2) &= H(u_{i_1},u_{i_2}) + (u_1-u_{i_1})\partial H(u_{i_1},u_{i_2})/\partial u_1 + (u_2-u_{i_2})\partial H(u_{i_1},u_{i_2})/\partial u_2 + \\ & \frac{(u-u_{i_1})^2}{2}\partial^2 H(u_{i_1},u_{i_2})/\partial u_1^2 + \frac{(u-u_{i_2})^2}{2}\partial^2 H(u_{i_1},u_{i_2})/\partial u_2^2 + \\ & (u_1-u_{i_1})(u_2-u_{i_2})\partial^2 H(u_{i_1},u_{i_2})/(\partial u_1 \partial u_2)\cdots \end{aligned} \tag{11.17}$$

Substituting Equation 11.17 into Equation 11.16 gives:

$$I = \Delta u_1 \Delta u_2 \sum_{i_1=1}^{N_{\rm cell}^{1/2}} \sum_{i_2=1}^{N_{\rm cell}^{1/2}} \left(H(u_{i_1},u_{i_2}) + \frac{(\Delta u_1)^2}{24}\partial^2 H(u_{i_1},u_{i_2})\partial u_1^2 + \frac{(\Delta u_2)^2}{24}\partial^2 H(u_{i_1},u_{i_2})\partial u_2^2 \cdots \right) , \tag{11.18}$$

where all the terms linear in $u-u_{i_1}$ or $u-u_{i_2}$ have vanished due to the midpoint symmetry.

The first term in the above summation gives the "first-order" estimate while the next two terms give an estimate of its error.

Therefore we write the fractional error as:

$$\frac{\Delta I}{I} = \frac{1}{N_{\rm cell}}\frac{1}{24}\frac{\sum_{i_1=1}^{N_{\rm cell}^{1/2}}\sum_{i_2=1}^{N_{\rm cell}^{1/2}}\left[(u_{1,\max}-u_{1,\min})^2\partial^2 H(u_{i_1},u_{i_2})\partial u_1^2 + (u_{2,\max}-u_{2,\min})^2\partial^2 H(u_{i_1},u_{i_2})\partial u_2^2\right]}{\sum_{i_1=1}^{N_{\rm cell}^{1/2}}\sum_{i_2=1}^{N_{\rm cell}^{1/2}} H(u_{i_1},u_{i_2})} \tag{11.19}$$

11.2.3 $\mathcal{D}$ dimensions

In $\mathcal{D}$ dimensions the calculation is no more difficult than in two dimensions, only the notation is more cumbersome. One notes that the integral takes the form:

$$\begin{aligned} I &= \int_{u_{1,\min}}^{u_{1,\max}} {\rm d}u_1 \int_{u_{2,\min}}^{u_{2,\max}} {\rm d}u_2 \cdots \int_{u_{\mathcal{D},\min}}^{u_{\mathcal{D},\max}} {\rm d}u_{\mathcal{D}}\, H(u_1,u_2\cdots u_{\mathcal{D}}) \\ &= \sum_{i_1=1}^{N_{\rm cell}^{1/\mathcal{D}}} \int_{u_{i_1}-\Delta u_1/2}^{u_{i_1}+\Delta u_1/2} {\rm d}u_1 \sum_{i_2=1}^{N_{\rm cell}^{1/\mathcal{D}}} \int_{u_{i_2}-\Delta u_2/2}^{u_{i_2}+\Delta u_2/2} {\rm d}u_2 \cdots \int_{u_{i_\mathcal{D}}-\Delta u_\mathcal{D}/2}^{u_{i_\mathcal{D}}+\Delta u_\mathcal{D}/2} {\rm d}u_{\mathcal{D}} \sum_{i_\mathcal{D}=1}^{N_{\rm cell}^{1/\mathcal{D}}} H(u_1,u_2\cdots u_{\mathcal{D}}) \end{aligned} \tag{11.20}$$

The Taylor expansion takes the form

$$H(u_1,u_2\cdots u_\mathcal{D}) = H(u_{i_1},u_{i_2}\cdots u_{i_\mathcal{D}}) + \sum_{j=1}^{\mathcal{D}}(u_i - u_{i_j})\partial H(u_{i_1},u_{i_2}\cdots u_{i_\mathcal{D}})/\partial u_j +$$

$$\sum_{j=1}^{\mathcal{D}} \frac{(u_i - u_{i_j})^2}{2} \partial^2 H(u_{i_1}, u_{i_2} \cdots u_{i_\mathcal{D}})/\partial u_j^2 +$$
$$\sum_{j=1}^{\mathcal{D}} \sum_{k \neq j=1}^{\mathcal{D}} (u_i - u_{i_j})(u_i - u_{i_k}) \partial^2 H(u_{i_1}, u_{i_2} \cdots u_{i_\mathcal{D}})/\partial u_i \partial u_j \cdots \quad (11.21)$$

The linear terms of the form $(u_i - u_{i_j})$ and the bilinear terms of the form $(u_i - u_{i_j})(u_i - u_{i_k})$ for $k \neq j$ all vanish by symmetry and a relative $N^{-2/\mathcal{D}}$ is extracted from the quadratic terms after integration. The result is that:

$$\frac{\Delta I}{I} = \frac{1}{24 N_{\mathrm{cell}}^{2/\mathcal{D}}} \frac{\sum_{i_1=1}^{N_{\mathrm{cell}}^{1/\mathcal{D}}} \sum_{i_2=1}^{N_{\mathrm{cell}}^{1/\mathcal{D}}} \cdots \sum_{i_\mathcal{D}=1}^{N_{\mathrm{cell}}^{1/\mathcal{D}}} \sum_{d=1}^{\mathcal{D}} (u_{d,\max} - u_{d,\min})^2 \partial^2 H(u_{i_1}, u_{i_2} \cdots u_{i_\mathcal{D}})/\partial u_d^2}{\sum_{i_1=1}^{N_{\mathrm{cell}}^{1/\mathcal{D}}} \sum_{i_2=1}^{N_{\mathrm{cell}}^{1/\mathcal{D}}} \cdots \sum_{i_\mathcal{D}=1}^{N_{\mathrm{cell}}^{1/\mathcal{D}}} H(u_{i_1}, u_{i_2} \cdots u_{i_\mathcal{D}})} . \quad (11.22)$$

Note that the one and two-dimensional result can be obtained from the above equation. The critical feature to note is the overall $N_{\mathrm{cell}}^{-2/\mathcal{D}}$ convergence rate. The more dimensions in the problem, the slower the convergence for numerical quadrature.

11.3 Convergence of Monte Carlo solutions

An alternative approach to solving Equation 11.1 is the Monte Carlo method whereby N_{hist} particle histories are simulated. In this case, the Monte Carlo converges to the true answer according to the central limit theorem [Fel67] which is expressed as:

$$\frac{\Delta T_{\mathrm{MC}}(x)}{T_{\mathrm{MC}}(x)} = \frac{1}{\sqrt{N_{\mathrm{hist}}}} \frac{\sigma_{\mathrm{MC}}(x)}{T_{\mathrm{MC}}(x)} , \quad (11.23)$$

where $T_{\mathrm{MC}}(x)$ is the tally calculated in a voxel located at x as calculated by the Monte Carlo method and $\sigma_{\mathrm{MC}}^2(x)$ is the variance associated with the *distribution* of $T_{\mathrm{MC}}(x)$. Note that this variance $\sigma_{\mathrm{MC}}^2(x)$ is an intrinsic feature of how the particle trajectories deposit energy in the spatial voxel. It is a "constant" for a given set of initial conditions and is conventionally estimated from the sample variance. It is also assumed, for the purpose of this discussion, that the sample variance exists and is finite.

11.4 Comparison between Monte Carlo and Numerical Quadrature

The deterministic models considered in this discussion pre-calculate $F(|x - x'|, p')$ of Equation 11.7 or $F(x, x', p, p', \Delta s)$ of Equation 11.10 storing them in arrays for iterative use. Then, during the iterative calculation phase a granulated matrix operation is performed. The associated matrix product is mathematically similar to the "mid-point" N_x+N_p-multidimensional integration discussed previously:

$$T(x) = \int_{\mathcal{D}} \mathrm{d}u\, H(u, x) \ , \tag{11.24}$$

where $\mathcal{D} = N_x + N_p$ and $u = (x_1, x_2 \cdots x_{N_x}, p_1, p_2 \cdots p_{N_p})$. That is, u is a multidimensional variable that encompasses both space and momentum. In the case of photon convolution, $H(u, x)$ can be inferred from Equation 11.6 and takes the explicit form:

$$H(u, x) = \int \mathrm{d}p\, F(|x - x'|, p')Q(x', p') \ . \tag{11.25}$$

There is a similar expression for the phase space evolution model.

The "mid-point" integration represents a "first-order" deterministic technique and is applied more generally than the convolution or phase space evolution applications. As shown previously, the convergence of this technique obeys the relationship:

$$\frac{\Delta T_{\mathrm{NMC}}(x)}{T_{\mathrm{NMC}}(x)} = \frac{1}{N_{\mathrm{cell}}^{2/\mathcal{D}}} \frac{\sigma_{\mathrm{NMC}}(x)}{T_{\mathrm{NMC}}(x)} \ , \tag{11.26}$$

where $T_{\mathrm{NMC}}(x)$ is the tally in a spatial voxel in an arbitrary N_x-dimensional geometry calculated by a non-Monte Carlo method where N_p momentum components are considered. The D-dimensional phase space has been divided into N_{cell} "cells" equally divided among all the dimensions so that the "mesh-size" of each phase space dimension is $N_{\mathrm{cell}}^{1/\mathcal{D}}$. The constant of proportionality as derived previously is:

$$\sigma_{\mathrm{NMC}}(x) = \frac{1}{24} \sum_{i_1=1}^{N_{\mathrm{cell}}^{1/\mathcal{D}}} \sum_{i_2=1}^{N_{\mathrm{cell}}^{1/\mathcal{D}}} \cdots \sum_{i_{\mathcal{D}}=1}^{N_{\mathrm{cell}}^{1/\mathcal{D}}} \sum_{d=1}^{\mathcal{D}} (u_{d,\mathrm{max}} - u_{d,\mathrm{min}})^2 \partial^2 H(u_{i_1}, u_{i_2} \cdots u_{i_{\mathcal{D}}})/\partial u_d^2 \ , \tag{11.27}$$

where the u-space of $H(u)$ has been partitioned in the same manner as the phase space described above. $u_{d,\mathrm{min}}$ is the minimum value of u_d while $u_{d,\mathrm{max}}$ is its maximum value. u_{i_j} is the midpoint of the cell in the j^{th} dimension at the i_j^{th} mesh index.

The equation for the proportionality factor is quite complicated. However, the important point to notice is that it depends only on the second derivatives of $H(u)$ with respect to the phase-space variables, u. Moreover, the non-Monte Carlo proportionality factor is quite different from the Monte Carlo proportionality factor. It would be difficult to predict which would be smaller and almost certainly would be application dependent.

We now assume that the computation time in either case is proportional to N_{hist} or cell, That is, $T_{\mathrm{MC}} = \alpha_{\mathrm{MC}} N_{\mathrm{hist}}$ and $T_{\mathrm{NMC}} = \alpha_{\mathrm{NMC}} N_{\mathrm{cell}}$. In the Monte Carlo case, the computation time is simply N_{hist} times the average computation time/history. In the non-Monte Carlo case, the matrix operation can potentially attempt to connect every cell in the $\mathcal{D}$-dimensional phase space to the tally at point x. Thus, a certain number of floating-point and integer operations are required for each cell in the problem.

Consider the convergence of the Monte Carlo and non-Monte Carlo method. Using the above relationships, one can show that:

$$\frac{\Delta T_{\rm MC}(x)/T_{\rm MC}(x)}{\Delta T_{\rm NMC}(x)/T_{\rm NMC}(x)} = \left(\frac{\sigma_{\rm NMC}(x)}{\sigma_{\rm MC}(x)}\right)\left(\frac{\alpha_{\rm NMC}^{\mathcal{D}}}{\alpha_{\rm MC}}\right)^{1/2} t^{(4-\mathcal{D})/2\mathcal{D}} \ , \tag{11.28}$$

where t is the time measuring computational effort for either method. We have assumed that the two calculational techniques are the same so that given enough time $D_{\rm MC}(x) \approx D_{\rm NMC}(x)$. One sees that given longer enough, the Monte Carlo method is always more advantageous for $\mathcal{D} > 4$. We also note that inefficient programming in the non-Monte Carlo method is severely penalized in this comparison of the two methods.

Assume that one desires to do a calculation to a prescribed $\varepsilon = \Delta T(x)/T(x)$. Using the relations derived so far, we calculate the relative amount time to execute the task to be:

$$\frac{t_{\rm NMC}}{t_{\rm MC}} = \left(\frac{\alpha_{\rm MC}}{\alpha_{\rm NMC}}\right)\left(\frac{[\sigma_{\rm NMC}(x)/T_{\rm NMC}(x)]^{\mathcal{D}/2}}{\sigma_{\rm MC}(x)/T_{\rm MC}(x)}\right)\varepsilon^{(4-\mathcal{D})/2} \ , \tag{11.29}$$

which again shows an advantage for the Monte Carlo method for $\mathcal{D} > 4$. Of course, this conclusion depends somewhat upon assumptions of the efficiency ratio $\alpha_{\rm MC}/\alpha_{\rm NMC}$ which would be dependent on the details of the calculational technique. Our conclusion is also dependent on the ratio $[\{\sigma_{\rm NMC}(x)/T_{\rm NMC}(x)\}^{\mathcal{D}/2}]/[\sigma_{\rm MC}(x)/T_{\rm MC}(x)]$ which relates to the detailed shape of the response functions. For distributions that can vary rapidly the Monte Carlo method is bound to be favored. When the distributions are flat, non-Monte Carlo techniques may be favored.

Nonetheless, at some level of complexity (large number of $N_{\rm cell}$'s required) Monte Carlo becomes more advantageous. Whether or not one's application crosses this complexity "threshold" has to be determined on a case-by-case-basis.

Smaller dimensional problems will favor the use of non-Monte Carlo techniques. The degree of the advantage will depend again on the details of the application.

Bibliography

[DM79] J. J. Duderstadt and W. M. Martin. *Transport Theory* . Wiley, New York, 1979.

[Fel67] W. Feller. *An introduction to probability theory and its applications, Volume I, 3rd Edition*. Wiley, New York, 1967.

[Lar92] E. W. Larsen. A theoretical derivation of the condensed history algorithm. *Ann. Nucl. Energy*, 19:701 – 714, 1992.

[SF96] J. K. Shultis and R. E. Faw. *Radiation Shielding* . Prentice Hall, Upper Saddle River, 1996.

Problems

1. Solve the following integral both by Monte Carlo (seeding an D-dimensional unit cube and also by simple mid-point summation :

$$I = \prod_{i=1}^{D} \int_0^1 \,\mathrm{d}x_i e^{-x_i}$$

for $D = 1, 2, 3, 4, 5, 6, 7, 8$. D is the dimensionality of the problem. Make the number of cells, N_{cell}, and the number of histories, N_{hist} equal in each case, so employ the following table:

D	1	2	3	4	5	6	7	8
N_{cell}	65536	256^2	40^3	16^4	9^5	6^6	5^7	4^8
N_{hist}	65536	65536	64000	65536	59049	46656	78125	65536

Compare with the simple mathematical result $(1 - 1/e)^D$. What conclusions can you make regarding the computational efficiency of either technique as the dimensionality increases?

Note: At high dimensionality, the random number generator is severely tested!

Chapter 12

Photon Monte Carlo Simulation

> *"I could have done it in a much more complicated way"*
> *said the red Queen, immensely proud.*
> Lewis Carroll

In this chapter we discuss the basic mechanism by which the simulation of photon interaction and transport is undertaken. We start with a review of the basic interaction processes that are involved, some common simplifications and the relative importance of the various processes. We discuss when and how one goes about choosing, by random selection, which process occurs. We discuss the rudimentary geometry involved in the transport and deflection of photons. We conclude with a schematic presentation of the logic flow executed by a typical photon Monte Carlo transport algorithm. This chapter will only sketch the bare minimum required to construct a photon Monte Carlo code. A particularly good reference for a description of basic interaction mechanisms is the excellent book [Eva55] by Robley Evans, *The Atomic Nucleus.* This book should be in the bookshelf of anyone undertaking a career in the radiation sciences. Simpler descriptions of photon interaction processes are useful as well and are included in many common textbooks [JC83, Att86, SF96].

12.1 Basic photon interaction processes

We now give a brief discussion of the photon interaction processes that should be modeled by a photon Monte Carlo code, namely:

- Pair production in the nuclear field
- The Compton interaction (incoherent scattering)

- The photoelectric interaction
- The Rayleigh interaction (coherent scattering)

12.1.1 Pair production in the nuclear field

As seen in Figure 12.1, a photon can interact in the field of a nucleus, annihilate and produce an electron-positron pair. A third body, usually a nucleus, is required to be present to conserve energy and momentum. This interaction scales as Z^2 for different nuclei. Thus, materials containing high atomic number materials more readily convert photons into charged particles than do low atomic number materials. This interaction is the quantum "analog" of the bremsstrahlung interaction, which we will encounter in Chapter 13 *Electron Monte Carlo simulation.* At high energies, greater than 50 MeV or so in all materials, the pair and bremsstrahlung interactions dominate. The pair interaction gives rise to charged particles in the form of electrons and positrons (muons at very high energy) and the bremsstrahlung interaction of the electrons and positrons leads to more photons. Thus there is a "cascade" process that quickly converts high energy electromagnetic particles into copious amounts of lower energy electromagnetic particles. Hence, a high-energy photon or electron beam not only has "high energy", it is also able to deposit a lot of its energy near one place by virtue of this cascade phenomenon. A picture of this process is given in Figure 12.2.

The high-energy limit of the pair production cross section per nucleus takes the form:

$$\lim_{\alpha \to \infty} \sigma_{\mathrm{pp}}(\alpha) = \sigma_0^{\mathrm{pp}} Z^2 \left(\ln(2\alpha) - \frac{109}{42} \right) , \tag{12.1}$$

where $\alpha = E_\gamma / m_e c^2$, that is, the energy of the photon divided by the rest mass energy[1] of the electron ($0.51099907 \pm 0.00000015$ MeV) and $\sigma_0^{\mathrm{pp}} = 1.80 \times 10^{-27}$ cm^2/nucleus. We note that the cross section grows logarithmically with incoming photon energy.

The kinetic energy distribution of the electrons and positrons is remarkably "flat" except near the kinematic extremes of $K_\pm = 0$ and $K_\pm = E_\gamma - 2m_e c^2$. Note as well that the rest-mass energy of the electron-positron pair must be created and so this interaction has a threshold at $E_\gamma = 2m_e c^2$. It is exactly zero below this energy.

Occasionally it is one of the *electrons* in the atomic cloud surrounding the nucleus that interacts with the incoming photon and provides the necessary third body for momentum and energy conservation. This interaction channel is suppressed by a factor of $1/Z$ relative to the nucleus-participating channel as well as additional phase-space and Pauli exclusion differences. In this case, the atomic electron is ejected with two electrons and one positron emitted. This is called "triplet" production. It is common to include the effects of triplet production by "scaling up" the two-body reaction channel and ignoring the 3-body kinematics. This is a good approximation for all but the low-Z atoms.

[1] The latest information on particle data is available on the web at: `http://pdg.lbl.gov/pdg.html` This web page is maintained by the Particle Data Group at the Lawrence Berkeley laboratory.

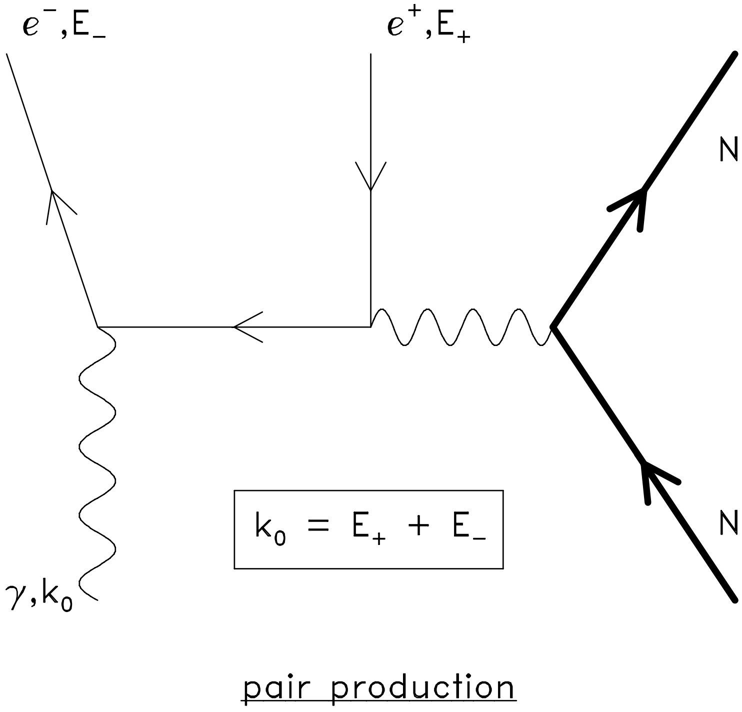

Figure 12.1: The Feynman diagram depicting pair production in the field of a nucleus. Occasionally (suppressed by a factor of $1/Z$), "triplet" production occurs whereby the incoming photon interacts with one of the electrons in the atomic cloud resulting in a final state with two electrons and one positron. (Picture not shown.)

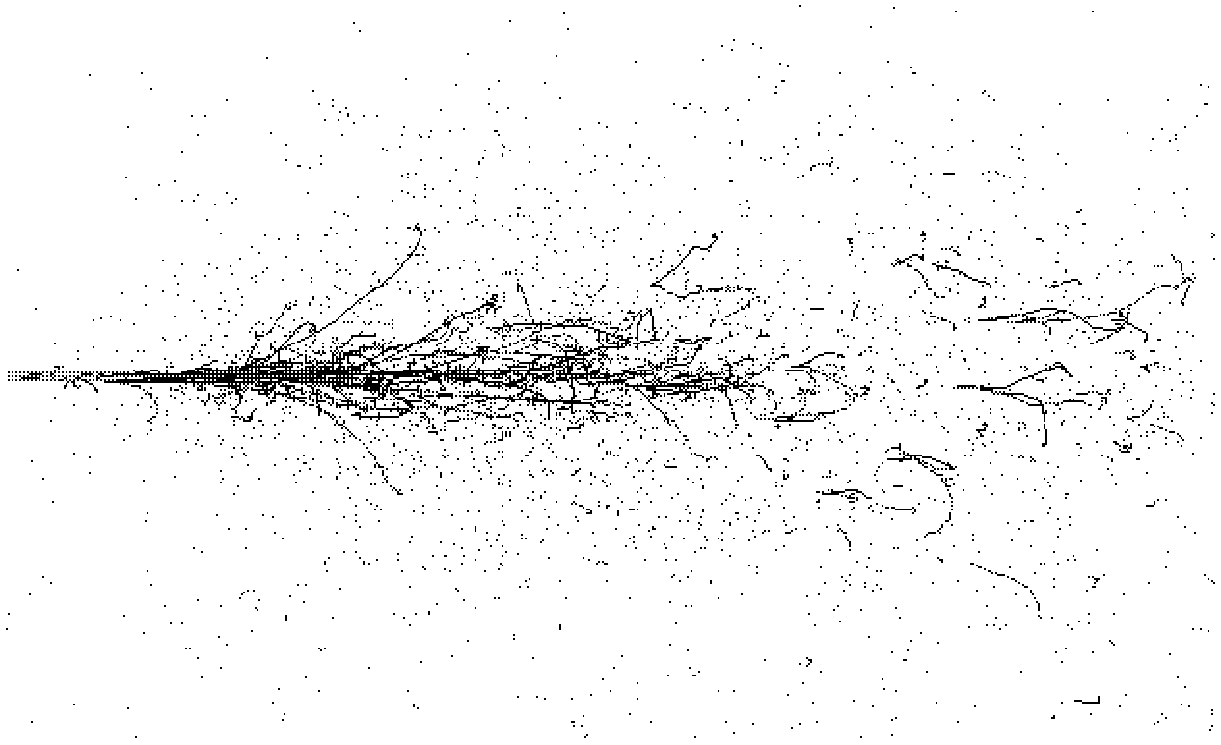

Figure 12.2: A simulation of the cascade resulting from five 1.0 GeV electrons incident from the left on a target. The electrons produce photons which produce electron-positron pairs and so on until the energy of the particles falls below the cascade region. Electron and positron tracks are shown with black lines. Photon tracks are not shown explaining why some electrons and positrons appear to be "disconnected". This simulation depicted here was produced by the EGS4 code [NHR85, BHNR94] and the system for viewing the trajectories is called EGS_Windows [BW91].

Further reading on the pair production interaction can be found in the reviews by Davies, Bethe, Maximon [DBM54], Motz, Olsen, and Koch [MOK69], and Tsai [Tsa74].

12.1.2 The Compton interaction (incoherent scattering)

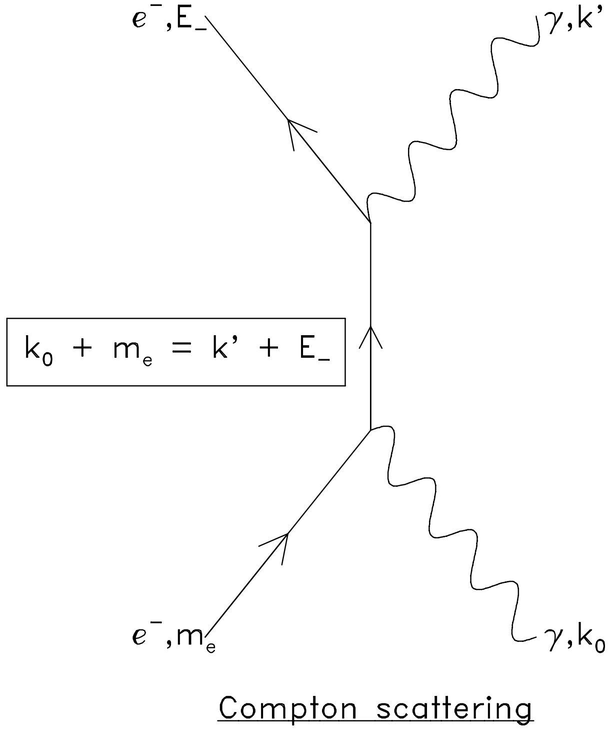

Figure 12.3: The Feynman diagram depicting the Compton interaction in free space. The photon strikes an electron assumed to be "at rest". The electron is set into motion and the photon recoils with less energy.

The Compton interaction [CA35] is an inelastic "bounce" of a photon from an electron in the atomic shell of a nucleus. It is also known as "incoherent" scattering in recognition of the fact that the recoil photon is reduced in energy. A Feynman diagram depicting this process is given in Figure 12.3. At large energies, the Compton interaction approaches asymptotically:

$$\lim_{\alpha \to \infty} \sigma_{\mathrm{inc}}(\alpha) = \sigma_0^{\mathrm{inc}} \frac{Z}{\alpha} \ , \tag{12.2}$$

where $\sigma_0^{\mathrm{inc}} = 3.33 \times 10^{-25}$ cm^2/nucleus. It is proportional to Z (*i.e.* the number of electrons) and falls off as $1/E_\gamma$. Thus, the Compton cross section per unit mass is nearly a constant independent of material and the energy-weighted cross section is nearly a constant independent of energy. Unlike the pair production cross section, the Compton cross section decreases with increased energy.

At low energies, the Compton cross section becomes a constant with energy. That is,

$$\lim_{\alpha \to 0} \sigma_{\mathrm{inc}}(\alpha) = 2\sigma_0^{\mathrm{inc}} Z \ . \tag{12.3}$$

This is the classical limit and it corresponds to Thomson scattering, which describes the scattering of light from “free” (unbound) electrons. In almost all applications, the electrons are bound to atoms and this binding has a profound effect on the cross section at low energies. However, above about 100 keV on can consider these bound electrons as “free”, and ignore atomic binding effects. As seen in Figure 12.4, this is a good approximation for photon energies down to 100 of keV or so, for most materials. This lower bound is defined by the K-shell energy although the effects can have influence greatly above it, particularly for the low-Z elements. Below this energy the cross section is depressed since the K-shell electrons are too tightly bound to be liberated by the incoming photon. The unbound Compton differential cross section is taken from the Klein-Nishina cross section [KN29], derived in lowest order Quantum Electrodynamics, without any further approximation.

It is possible to improve the modeling of the Compton interaction. Namito and Hirayama [NH91] have considered the effect of binding for the Compton effect as well as allowing for the transport of polarised photons for both the Compton and Rayleigh interactions.

12.1.3 Photoelectric interaction

The dominant low energy photon process is the photoelectric effect. In this case the photon gets absorbed by an electron of an atom resulting in escape of the electron from the atom and accompanying small energy photons as the electron cloud of the atom settles into its ground state. The theory concerning this phenomenon is not complete and exceedingly complicated. The cross section formulae are usually in the form of numerical fits and take the form:

$$\sigma_{\mathrm{ph}}(E_\gamma) \propto \frac{Z^m}{E_\gamma^n} \ , \tag{12.4}$$

where the exponent on Z ranges from 4 (low energy, below 100 keV) to 4.6 (high energy, above 500 keV) and the exponent on E_γ ranges from 3 (low energy, below 100 keV) to 1

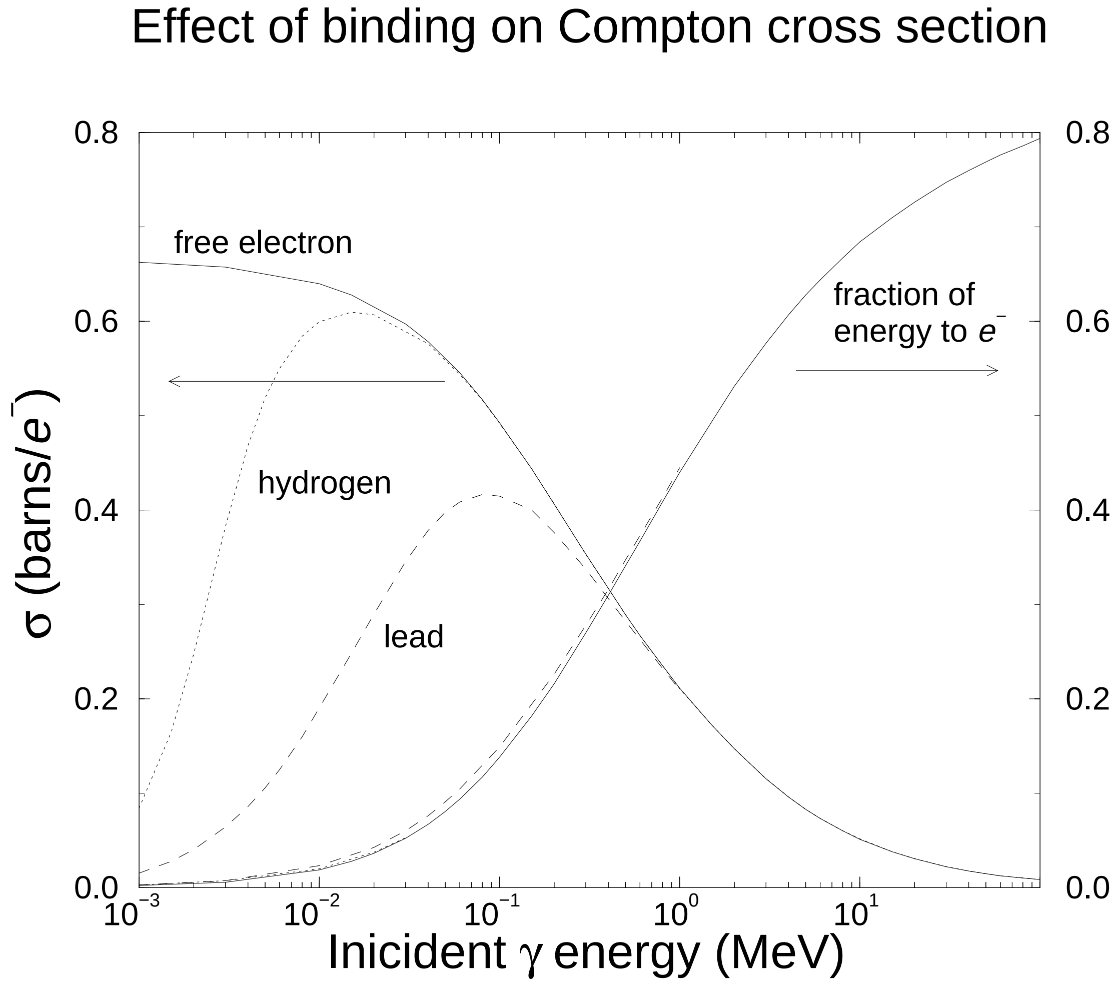

Figure 12.4: The effect of atomic binding on the Compton cross section.

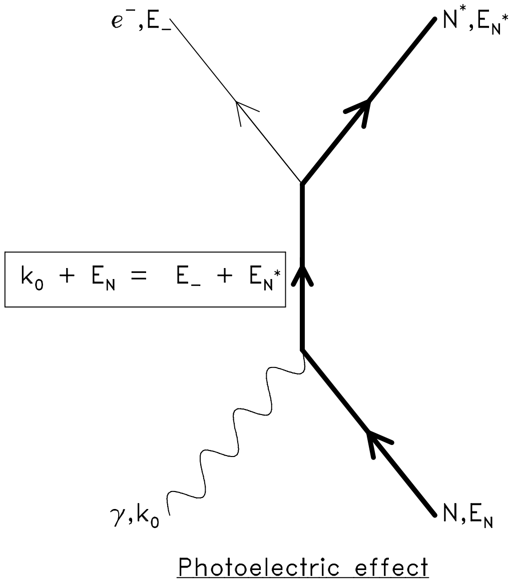

Figure 12.5: Photoelectric effect

(high energy, above 500 keV). Note that the high-energy fall-off is the same as the Compton interaction. However, the high-energy photoelectric cross section is depressed by a factor of about $Z^{3.6}10^{-8}$ relative to the Compton cross section and so is negligible in comparison to the Compton cross section at high energies.

A useful approximation that applies in the regime where the photoelectric effect is dominant is:

$$\sigma_{\mathrm{ph}}(E_\gamma) \propto \frac{Z^4}{E_\gamma^3} , \tag{12.5}$$

which is often employed for simple analytic calculations. However, most Monte Carlo codes employ a table look-up for the photoelectric interaction.

Angular distributions of the photoelectron can be determined according to the theory of Sauter [Sau31]. Although Sauter's theory is relativistic, it appears to work in the non-relativistic regime as well.

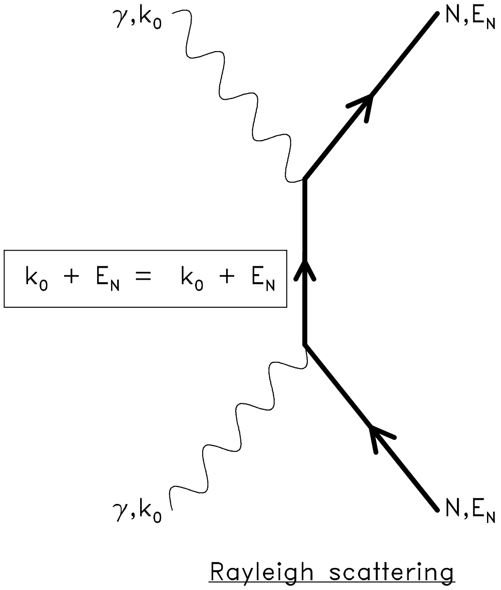

Figure 12.6: Rayleigh scattering

12.1.4 Rayleigh (coherent) interaction

Now we consider the Rayleigh interaction, also known as coherent scattering. In terms of cross section, the Rayleigh cross section is at least an order of magnitude *less* that the photoelectric cross section. However, it is still important! As can be seen from the Feynman diagram in Figure 12.6, the distinguishing feature of this interaction in contrast to the photoelectric interaction is that there is a photon in the final state. Indeed, if low energy photons impinge on an optically thick shield both Compton and Rayleigh scattered photons will emerge from the far side. Moreover, the proportions will be a sensitive function of the incoming energy.

The coherent interaction is an *elastic* (no energy loss) scattering from atoms. It is not good enough to treat molecules as if they are made up of independent atoms. A good demonstration of the importance of molecular structure was demonstrated by Johns and Yaffe [JY83].

The Rayleigh differential cross section has the following form:

$$\sigma_{\text{coh}}(E_\gamma, \Theta) = \frac{r_e^2}{2}(1 + \cos^2\Theta)[F(q, Z)]^2 \ , \tag{12.6}$$

where r_e is the classical electron radius (2.8179×10^{-13} cm), q is the *momentum-transfer parameter*, $q = (E_\gamma/hc)\sin(\Theta/2)$, and $F(q, Z)$ is the *atomic form factor.* $F(q, Z)$ approaches Z as q goes to zero either by virtue of E_γ going to zero or Θ going to zero. The atomic form factor also falls off rapidly with angle although the Z-dependence increases with angle to approximately $Z^{3/2}$.

The tabulation of the form factors published by Hubbell and Øverbø [HØ79].

12.1.5 Relative importance of various processes

We now consider the relative importance of the various processes involved.

For carbon, a moderately low-Z material, the relative strengths of the photon interactions versus energy is shown in Figure 12.7. For this material we note three distinct regions of single interaction dominance: photoelectric below 20 keV, pair above 30 MeV and Compton in between. The almost order of magnitude depression of the Rayleigh and triplet contributions is some justification for the relatively crude approximations we have discussed. For lead, shown in Figure 12.8, there are several differences and many similarities. The same comment about the relative unimportance of the Rayleigh and triplet cross sections applies. The "Compton dominance" section is much smaller, now extending only from 700 keV to 4 MeV. We also note quite a complicated structure below about 90 keV, the K-shell binding energy of the lead atom. Below this threshold, atomic structure effects become very important.

Finally, we consider the total cross section versus energy for the materials hydrogen, water and lead, shown in Figure 12.9. The total cross section is plotted in the units cm^2/g. The Compton dominance regions are equivalent except for a relative $\overline{A/Z}$ factor. At high energy the Z^2 dependence of pair production is evident in the lead. At lower energies the $Z^n (n > 4)$ dependence of the photoelectric cross section is quite evident.

12.2 Photon transport logic

We now discuss a simplified version of photon transport logic. It is simplified by ignoring electron creation and considering that the transport occurs in only a single volume element and a single medium.

This photon transport logic is schematised in Figure 12.10. Imagine that an initial photon's parameters are present at the top of an array called `STACK`. `STACK` is an array that retains particle phase space characteristics for processing. We also imagine that there is a photon

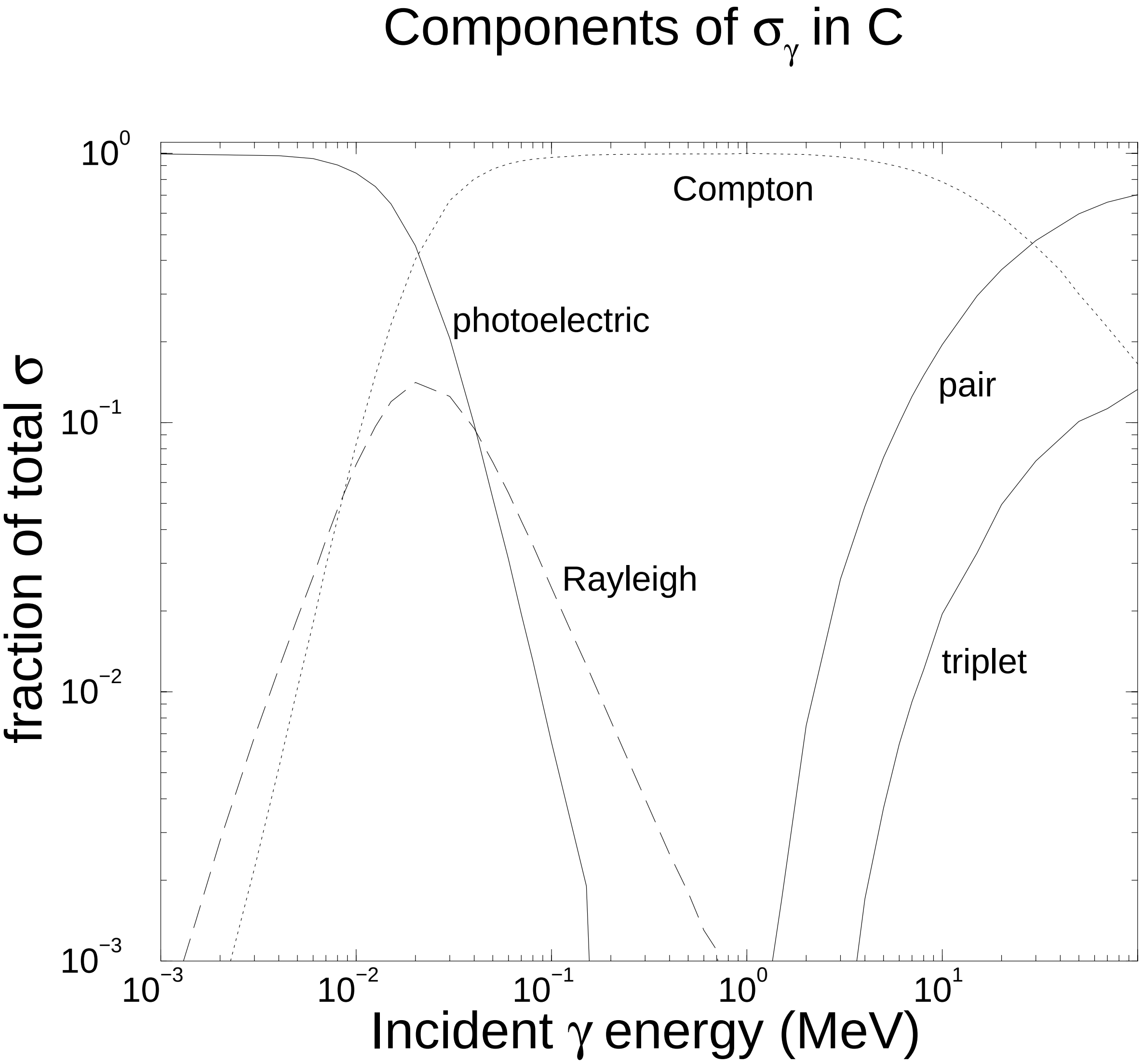

Figure 12.7: Components of the photon cross section in Carbon.

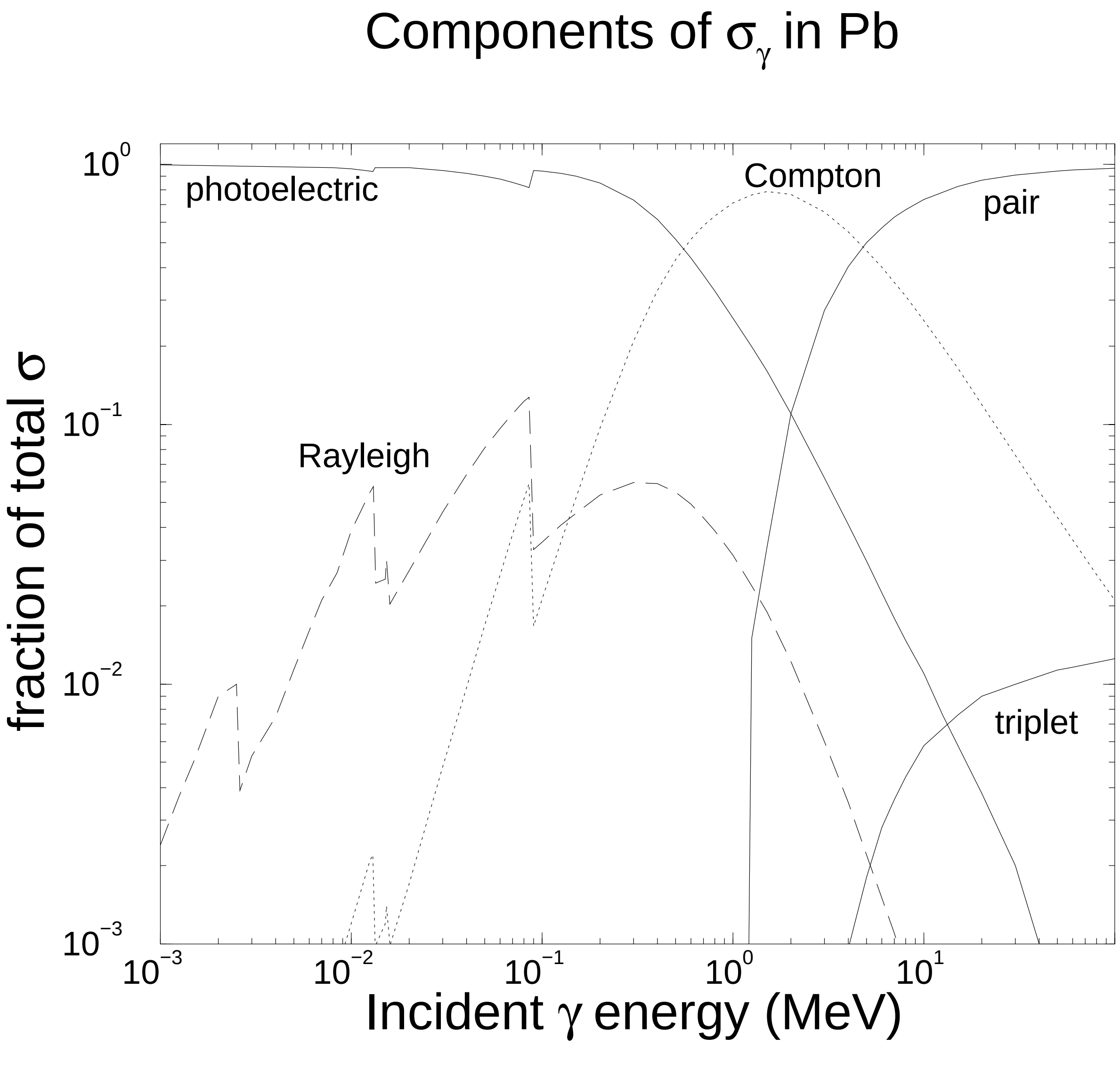

Figure 12.8: Components of the photon cross section in Lead.

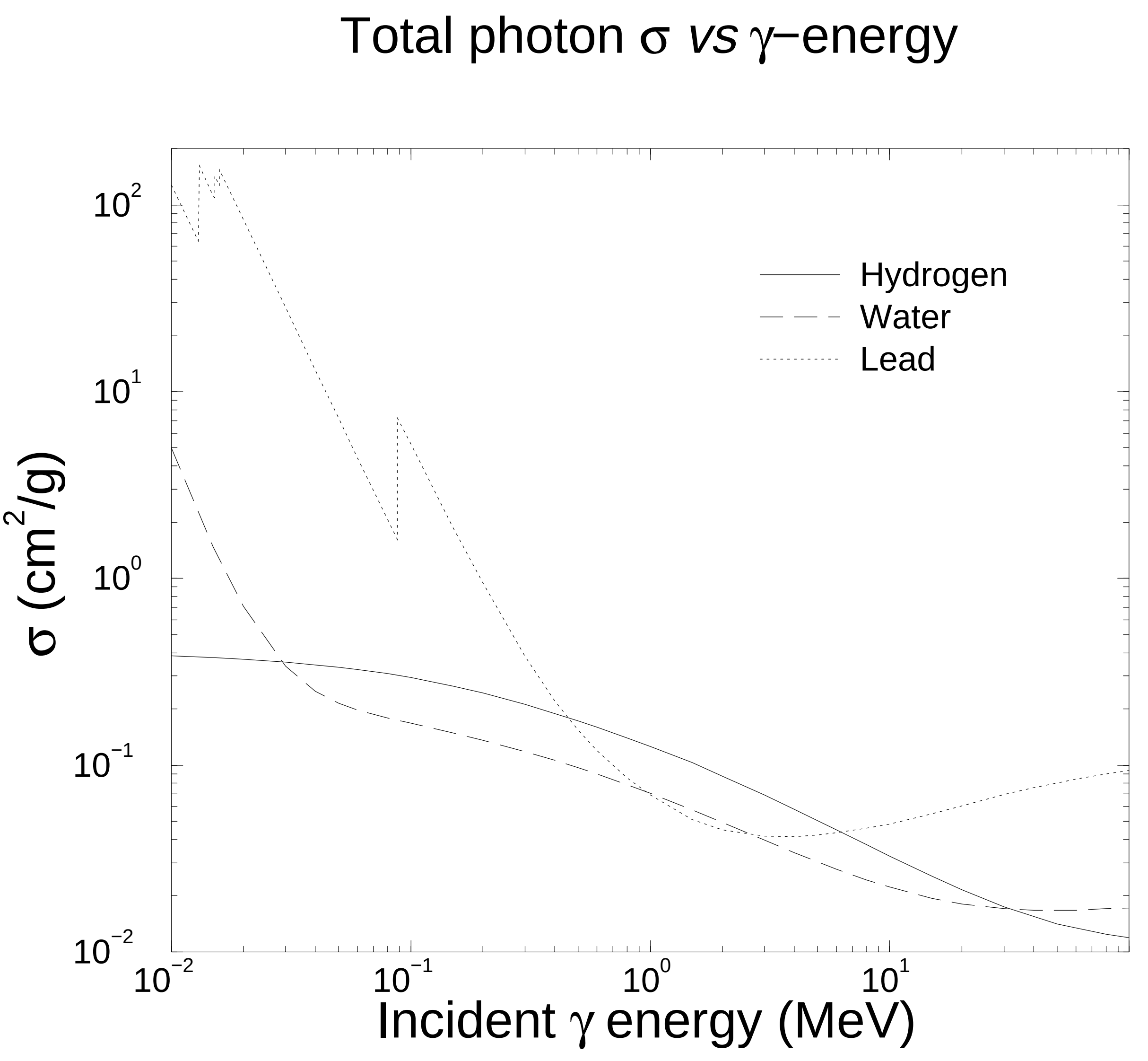

Figure 12.9: Total photon cross section *vs.* photon energy.

transport cutoff defined. Photons that fall below this cutoff are absorbed "on the spot". We consider that they do not contribute significantly to any tallies of interest and can be ignored. Physically, this step is not really necessary—it is only a time-saving manoeuvre. In "real life" low-energy photons are absorbed by the photoelectric process and vanish. (We will see that electrons are more complicated. *Electrons are <u>always</u> more complicated.*

The logic flow of photon transport proceeds as follow. The initial characteristics of a photon entering the transport routine and first tested to see if the energy is below the transport cutoff. If it is below the cutoff, the history is terminated. If the `STACK` is empty then a new particle history is started. If the energy is above the cutoff then the distance to the next interaction site is chosen, following the discussion in Chapter 8, *Transport in media, interaction models.* The photon is then transported, that is "stepped" to the point of interaction. (If the geometry is more complicated than just one region, transport through different elements of the geometry would be taken care of here.) If the photon, by virtue of its transport, has left the volume defining the problem then it is discarded. Otherwise, the branching distribution is sampled to see which interaction occurs. Having done this, the surviving particles (new ones may be created, some disappear, the characteristics of the initial one will almost certainly change) have their energies, directions and other characteristics chosen from the appropriate distributions. The surviving particles are put on the `STACK`. Lowest energy ones should be put on the top of the `STACK` to keep the size of the `STACK` as small as possible. Then the whole process takes place again until the `STACK` is empty and all the incident particles are used up.

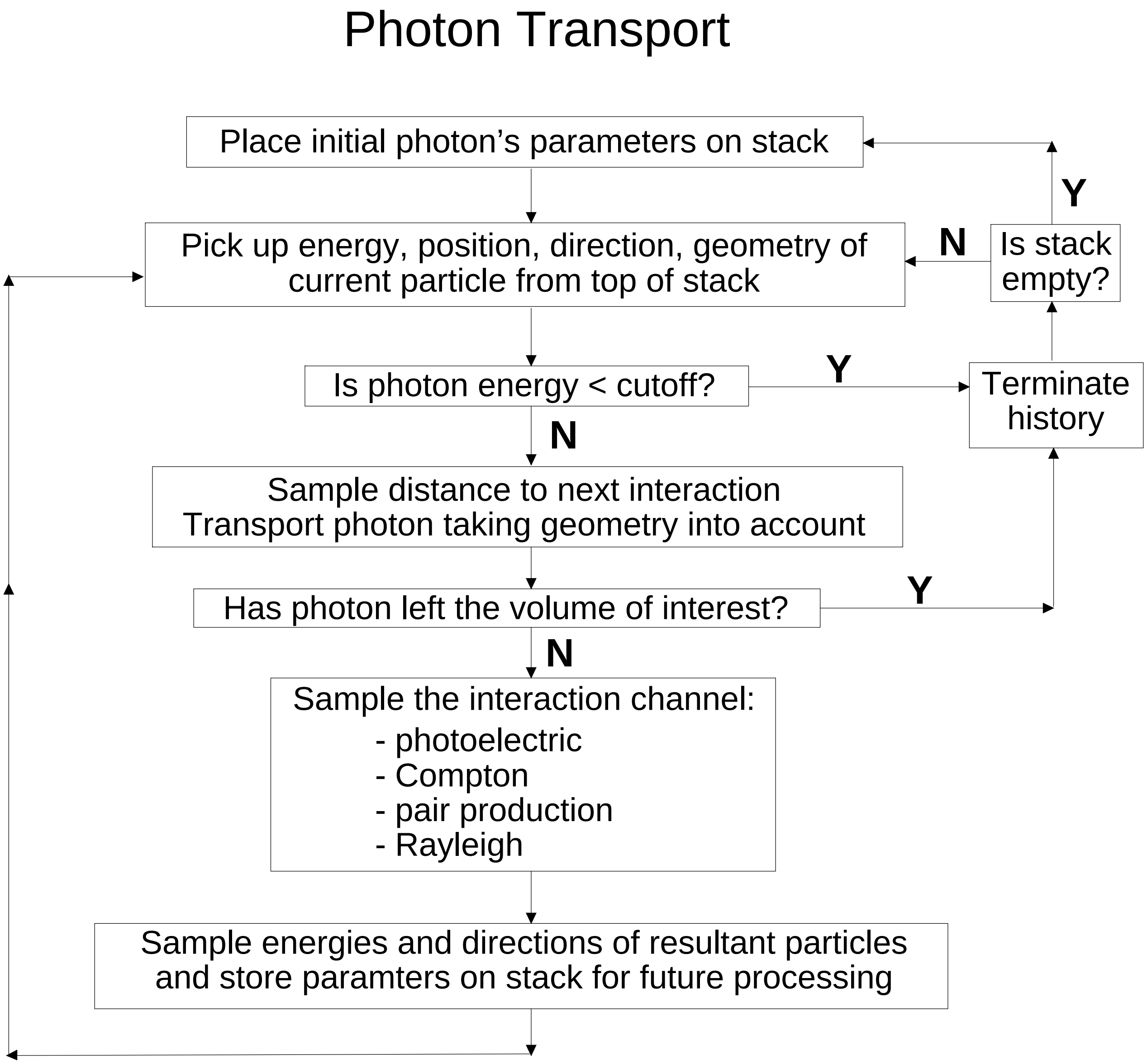
Photon Transport
Place initial photon's parameters on stack
Y
N
Is stack empty?
Pick up energy, position, direction, geometry of current particle from top of stack
Is photon energy < cutoff?
Y
Terminate history
N
Sample distance to next interaction
Transport photon taking geometry into account
Has photon left the volume of interest?
Y
N
Sample the interaction channel:
- photoelectric
- Compton
- pair production
- Rayleigh
Sample energies and directions of resultant particles and store paramters on stack for future processing

Figure 12.10: "Bare-bones" photon transport logic.

Bibliography

[Att86] F. H. Attix. *Introduction to Radiological Physics and Radiation Dosimetry.* Wiley, New York, 1986.

[BHNR94] A. F. Bielajew, H. Hirayama, W. R. Nelson, and D. W. O. Rogers. History, overview and recent improvements of EGS4. *National Research Council of Canada Report PIRS-0436*, 1994.

[BW91] A. F. Bielajew and P. E. Weibe. EGS-Windows - A Graphical Interface to EGS. *NRCC Report: PIRS-0274*, 1991.

[CA35] A. H. Compton and S. K. Allison. X-rays in theory and experiment. *(D. Van Nostrand Co. Inc, New York)*, 1935.

[DBM54] H. Davies, H. A. Bethe, and L. C. Maximon. Theory of bremsstrahlung and pair production. II. Integral cross sections for pair production. *Phys. Rev.*, 93:788, 1954.

[Eva55] R. D. Evans. *The Atomic Nucleus.* McGraw-Hill, New York, 1955.

[HØ79] J. H. Hubbell and I. Øverbø. Relativistic atomic form factors and photon coherent scattering cross sections. *J. Phys. Chem. Ref. Data*, 9:69, 1979.

[JC83] H. E. Johns and J. R. Cunningham. *The Physics of Radiology, Fourth Edition.* Charles C. Thomas, Springfield, Illinois, 1983.

[JY83] P. C. Johns and M. J. Yaffe. Coherent scatter in diagnostic radiology. *Med. Phys.*, 10:40, 1983.

[KN29] O. Klein and Y. Nishina. . *Z. für Physik*, 52:853, 1929.

[MOK69] J. W. Motz, H. A. Olsen, and H. W. Koch. Pair production by photons. *Rev. Mod. Phys.*, 41:581 – 639, 1969.

[NH91] Y. Namito and H. Hirayama. Improvement of low energy photon transport calculation by EGS4 – electron bound effect in Compton scattering. *Japan Atomic Energy Society, Osaka*, page 401, 1991.

[NHR85] W. R. Nelson, H. Hirayama, and D. W. O. Rogers. The EGS4 Code System. Report SLAC–265, Stanford Linear Accelerator Center, Stanford, Calif, 1985.

[Sau31] F. Sauter. Über den atomaren Photoeffekt in der K-Schale nach der relativistischen Wellenmechanik Diracs. *Ann. Physik*, 11:454 – 488, 1931.

[SF96] J. K. Shultis and R. E. Faw. *Radiation Shielding* . Prentice Hall, Upper Saddle River, 1996.

[Tsa74] Y. S. Tsai. Pair Production and Bremsstrahlung of Charged Leptons. *Rev. Mod. Phys.*, 46:815, 1974.

Problems

1.

Chapter 13

Electron Monte Carlo Simulation

In this chapter we discuss the electron and positron interactions and discuss the approximations made in their implementation. We give a brief outline of the electron transport logic used in Monte Carlo simulations.

The transport of electrons (and positrons) is considerably more complicated than for photons. Like photons, electrons are subject to violent interactions. The following are the "catastrophic" interactions:

- large energy-loss Møller scattering ($e^-e^- \longrightarrow e^-e^-$),
- large energy-loss Bhabha scattering ($e^+e^- \longrightarrow e^+e^-$),
- hard bremsstrahlung emission ($e^\pm \mathrm{N} \longrightarrow e^\pm\gamma\mathrm{N}$), and
- positron annihilation "in-flight" and at rest ($e^+e^- \longrightarrow \gamma\gamma$).

It is possible to sample the above interactions discretely in a reasonable amount of computing time for many practical problems. In addition to the catastrophic events, there are also "soft" events. Detailed modeling of the soft events can usually be approximated by summing the effects of many soft events into virtual "large-effect" events. These "soft" events are:

- low-energy Møller (Bhabha) scattering (modeled as part of the collision stopping power),
- atomic excitation ($e^\pm \mathrm{N} \longrightarrow e^\pm \mathrm{N}^*$) (modeled as another part of the collision stopping power),
- soft bremsstrahlung (modeled as radiative stopping power), and
- elastic electron (positron) multiple scattering from atoms, ($e^\pm \mathrm{N} \longrightarrow e^\pm \mathrm{N}$).

Strictly speaking, an elastic large angle scattering from a nucleus should really be considered to be a "catastrophic" interaction but this is not the usual convention. (Perhaps it should be.) For problems of the sort we consider, it is impractical to model all these interactions discretely. Instead, well-established statistical theories are used to describe these "soft" interactions by accounting for them in a cumulative sense including the effect of many such interactions at the same time. These are the so-called "statistically grouped" interactions.

13.1 Catastrophic interactions

We have almost complete flexibility in defining the threshold between "catastrophic" and "statistically grouped" interactions. The location of this threshold should be chosen by the demands of the physics of the problem and by the accuracy required in the final result.

13.1.1 Hard bremsstrahlung production

As depicted by the Feynman diagram in fig. 13.1, bremsstrahlung production is the creation of photons by electrons (or positrons) in the field of an atom. There are actually two possibilities. The predominant mode is the interaction with the atomic nucleus. This effect dominates by a factor of about Z over the three-body case where an atomic electron recoils ($e^{\pm}\mathrm{N} \longrightarrow e^{\pm}e^{-}\gamma\mathrm{N}^{*}$). Bremsstrahlung is the quantum analogue of synchrotron radiation, the radiation from accelerated charges predicted by Maxwell's equations. The de-acceleration and acceleration of an electron scattering from nuclei can be quite violent, resulting in very high energy quanta, up to and including the total kinetic energy of the incoming charged particle.

The two-body effect can be taken into account through the total cross section and angular distribution kinematics. The three-body case is conventionally treated only by inclusion in the total cross section of the two body-process. The two-body process can be modeled using one of the Koch and Motz [KM59] formulae. The bremsstrahlung cross section scales with $Z(Z + \xi(Z))$, where $\xi(Z)$ is the factor accounting for three-body case where the interaction is with an atomic electron. These factors comes are taken from the work of Tsai [Tsa74]. The total cross section depends approximately like $1/E_{\gamma}$.

13.1.2 Møller (Bhabha) scattering

Møller and Bhabha scattering are collisions of incident electrons or positrons with atomic electrons. It is conventional to assume that these atomic electrons are "free" ignoring their atomic binding energy. At first glance the Møller and Bhabha interactions appear to be quite similar. Referring to fig. 13.2, we see very little difference between them. In reality, however, they are, owing to the identity of the participant particles. The electrons in the $e^{-}e^{+}$ pair can annihilate and be recreated, contributing an extra interaction channel to the cross section. The thresholds for these interactions are different as well. In the $e^{-}e^{-}$ case, the "primary" electron can only give at most half its energy to the target electron if we adopt the convention that the higher energy electron is always denoted "the primary". This is because the two electrons are indistinguishable. In the e^+e^- case the positron can give up all its energy to the atomic electron.

Møller and Bhabha cross sections scale with Z for different media. The cross section scales approximately as $1/v^2$, where v is the velocity of the scattered electron. Many more low energy secondary particles are produced from the Møller interaction than from the bremsstrahlung interaction.

13.1.3 Positron annihilation

Two photon annihilation is depicted in fig. 13.3. Two-photon "in-flight" annihilation can be modeled using the cross section formulae of Heitler [Hei54]. It is conventional to consider the atomic electrons to be free, ignoring binding effects. Three and higher-photon annihilations ($e^+e^- \longrightarrow n\gamma[n > 2]$) as well as one-photon annihilation which is possible in the Coulomb field of a nucleus ($e^+e^-\mathrm{N} \longrightarrow \gamma\mathrm{N}^*$) can be ignored as well. The higher-order processes are very much suppressed relative to the two-body process (by at least a factor of 1/137) while the one-body process competes with the two-photon process only at very high energies where the cross section becomes very small. If a positron survives until it reaches the transport cut-off energy it can be converted it into two photons (annihilation at rest), with or without modeling the residual drift before annihilation.

13.2 Statistically grouped interactions

13.2.1 "Continuous" energy loss

One method to account for the energy loss to sub-threshold (soft bremsstrahlung and soft collisions) is to assume that the energy is lost continuously along its path. The formalism that may be used is the Bethe-Bloch theory of charged particle energy loss [Bet30, Bet32, Blo33] as expressed by Berger and Seltzer [BS64] and in ICRU 37 [ICR84]. This continuous energy loss scales with the Z of the medium for the collision contribution and Z^2 for the radiative part. Charged particles can also polarise the medium in which they travel. This "density effect" is important at high energies and for dense media. Default density effect parameters are available from a 1982 compilation by Sternheimer, Seltzer and Berger [SSB82] and state-of-the-art compilations (as defined by the stopping-power guru Berger who distributes a PC-based stopping power program [Ber92]).

Again, atomic binding effects are treated rather crudely by the Bethe-Bloch formalism. It assumes that each electron can be treated as if it were bound by an average binding potential. The use of more refined theories does not seem advantageous unless one wants to study electron transport below the K-shell binding energy of the highest atomic number element in the problem.

The stopping power versus energy for different materials is shown in fig. 13.4. The difference in the collision part is due mostly to the difference in ionisation potentials of the various atoms and partly to a $\overline{Z}/\overline{A}$ difference, because the vertical scale is plotted in MeV/(g/cm^2), a normalisation by atomic weight rather than electron density. Note that at high energy the argon line rises above the carbon line. Argon, being a gas, is reduced less by the density effect at this energy. The radiative contribution reflects mostly the relative Z^2 dependence of bremsstrahlung production.

The collisional energy loss by electrons and positrons is different for the same reasons described in the "catastrophic" interaction section. Annihilation is generally not treated as part of the positron slowing down process and is treated discretely as a "catastrophic" event. The differences are reflected in fig. 13.5, the positron/electron collision stopping power. The positron radiative stopping power is reduced with respect to the electron radiative stopping power. At 1 MeV this difference is a few percent in carbon and 60% in lead. This relative difference is depicted in fig. 13.6.

13.2.2 Multiple scattering

Elastic scattering of electrons and positrons from nuclei is predominantly small angle with the occasional large-angle scattering event. If it were not for screening by the atomic electrons, the cross section would be infinite. The cross sections are, nonetheless, very large. There are several statistical theories that deal with multiple scattering. Some of these theories assume that the charged particle has interacted enough times so that these interactions may be grouped together. The most popular such theory is the Fermi-Eyges theory [Eyg48], a small angle theory. This theory neglects large angle scattering and is unsuitable for accurate electron transport unless large angle scattering is somehow included (perhaps as a catastrophic interaction). The most accurate theory is that of Goudsmit and Saunderson [GS40a, GS40b]. This theory does not require that many atoms participate in the production of a multiple scattering angle. However, calculation times required to produce few-atom distributions can get very long, can have intrinsic numerical difficulties and are not efficient computationally for Monte Carlo codes such as EGS4 [NHR85, BHNR94] where the physics and geometry adjust the electron step-length dynamically. A fixed step-size scheme permits an efficient implementation of Goudsmit-Saunderson theory and this has been done in ETRAN [Sel89, Sel91], ITS [HM84, Hal89, HKM$^+$92] and MCNP [Bri86, Bri93, Bri97]. Apart from accounting for energy-loss during the course of a step, there is no intrinsic difficulty with large steps either. EGS4 uses the Molière theory [Mol47, Mol48] which produces results as good as Goudsmit-Saunderson for many applications and is much easier to implement in EGS4's transport scheme.

The Molière theory, although originally designed as a small angle theory has been shown with small modifications to predict large angle scattering quite successfully [Bet53, Bie94]. The Molière theory includes the contribution of single event large angle scattering, for example, an electron backscatter from a single atom. The Molière theory ignores differences in the scattering of electrons and positrons, and uses the screened Rutherford cross sections instead of the more accurate Mott cross sections. However, the differences are known to be small. Owing to analytic approximations made by Molière theory, this theory requires a minimum step-size as it breaks down numerically if less than 25 atoms or so participate in the development of the angular distribution [Bie94, AMB93]. A recent development [KB98] has surmounted this difficulty. Apart from accounting for energy loss, there is also a large step-size restriction because the Molière theory is couched in a small-angle formalism. Beyond this there are other corrections that can be applied [Bet53, Win87] related to the mathematical connection between the small-angle and any-angle theories.

13.3 Electron transport "mechanics"

13.3.1 Typical electron tracks

A typical Monte Carlo electron track simulation is shown in fig. 13.7. An electron is being transported through a medium. Along the way energy is being lost "continuously" to subthreshold knock-on electrons and bremsstrahlung. The track is broken up into small straight-line segments called *multiple scattering substeps.* In this case the length of these substeps was chosen so that the electron lost 4% of its energy during each step. At the end of each of these steps the multiple scattering angle is selected according to some theoretical distribution. Catastrophic events, here a single knock-on electron, sets other particles in motion. These particles are followed separately in the same fashion. The original particle, if it is does not fall below the transport threshold, is also transported. In general terms, this is exactly what the any electron transport logic simulates.

13.3.2 Typical multiple scattering substeps

Now we demonstrate in fig. 13.8 what a multiple scattering substep *should* look like.

A single electron step is characterised by the length of total curved path-length to the end point of the step, t. (This is a reasonable parameter to use because the number of atoms encountered along the way should be proportional to t.) At the end of the step the deflection from the initial direction, Θ, is sampled. Associated with the step is the average projected distance along the original direction of motion, s. There is no satisfactory theory for the relation between s and t! The lateral deflection, ρ, the distance transported perpendicular to the original direction of motion, is often ignored by electron Monte Carlo codes. This is *not* to say that lateral transport is not modelled! Recalling fig. 13.7, we see that such lateral deflections do occur as a result of multiple scattering. It is only the lateral deflection during the course of a substep which is ignored. One can guess that if the multiple scattering steps are small enough, the electron track may be simulated more exactly.

13.4 Examples of electron transport

13.4.1 Effect of physical modeling on a 20 MeV e^- depth-dose curve

In this section we will study the effects on the depth-dose curve of turning on and off various physical processes. Figure 13.9 presents two CSDA calculations (*i.e.* no secondaries are created and energy-loss straggling is not taken into account). For the histogram, no multiple scattering is modeled and hence there is a large peak at the end of the range of the particles because they all reach the same depth before being terminating and depositing their residual kinetic energy (189 keV in this case). Note that the size of this peak is very much a calculational artefact which depends on how thick the layer is in which the histories terminate. The curve with the stars includes the effect of multiple scattering. This leads to a lateral spreading of the electrons which shortens the depth of penetration of most electrons and increases the dose at shallower depths because the fluence has increased. In this case, the depth-straggling is entirely caused by the lateral scattering since every electron has traveled the same distance.

Figure 13.10 presents three depth-dose curves calculated with all multiple scattering turned off - *i.e.* the electrons travel in straight lines (except for some minor deflections when secondary electrons are created). In the cases including energy-loss straggling, a depth straggling is introduced because the actual distance traveled by the electrons varies, depending on how much energy they give up to secondaries. Two features are worth noting. Firstly, the energy-loss straggling induced by the creation of bremsstrahlung photons plays a significant role despite the fact that far fewer secondary photons are produced than electrons. They do, however, have a larger mean energy. Secondly, the inclusion of secondary electron transport in the calculation leads to a dose buildup region near the surface. Figure 13.11 presents a combination of the effects in the previous two figures. The extremes of no energy-loss straggling and the full simulation are shown to bracket the results in which energy-loss straggling from either the creation of bremsstrahlung or knock-on electrons is included. The bremsstrahlung straggling has more of an effect, especially near the peak of the depth-dose curve.

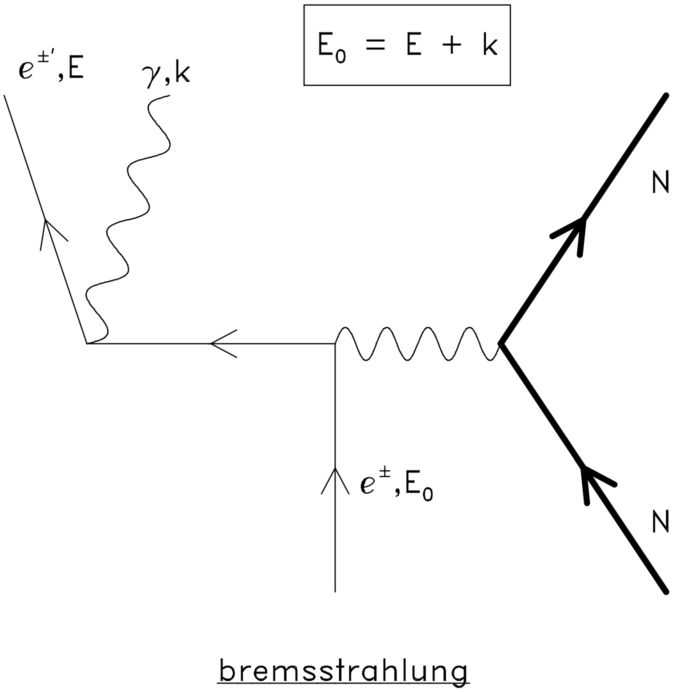

Figure 13.1: Hard bremsstrahlung production in the field of an atom as depicted by a Feynman diagram. There are two possibilities. The predominant mode (shown here) is a two-body interaction where the nucleus recoils. This effect dominates by a factor of about Z^2 over the three-body case where an atomic electron recoils (not shown).

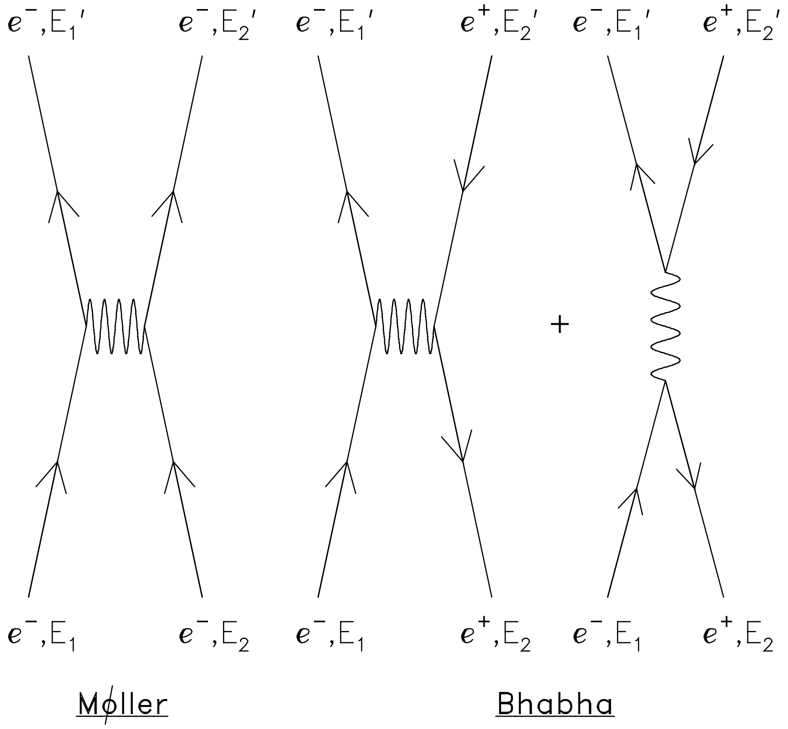

Figure 13.2: Feynman diagrams depicting the Møller and Bhabha interactions. Note the extra interaction channel in the case of the Bhabha interaction.

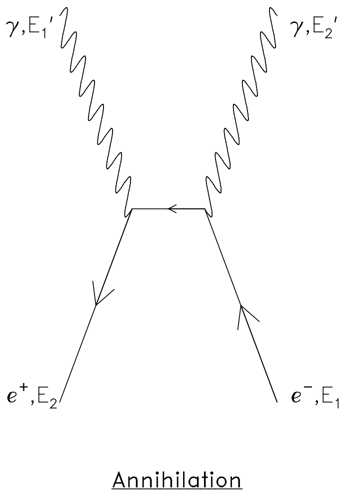

Figure 13.3: Feynman diagram depicting two-photon positron annihilation.

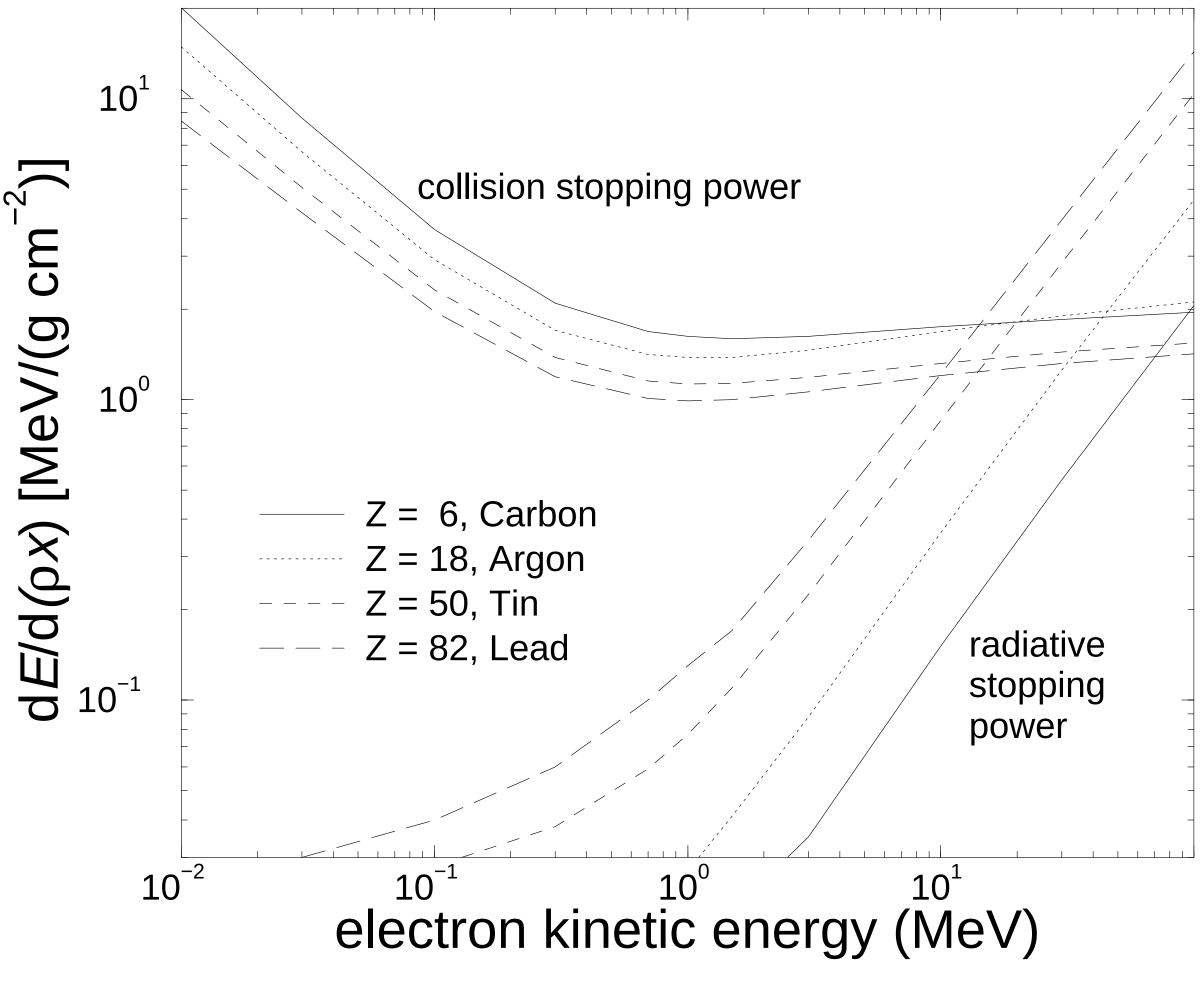

Figure 13.4: Stopping power *versus* energy.

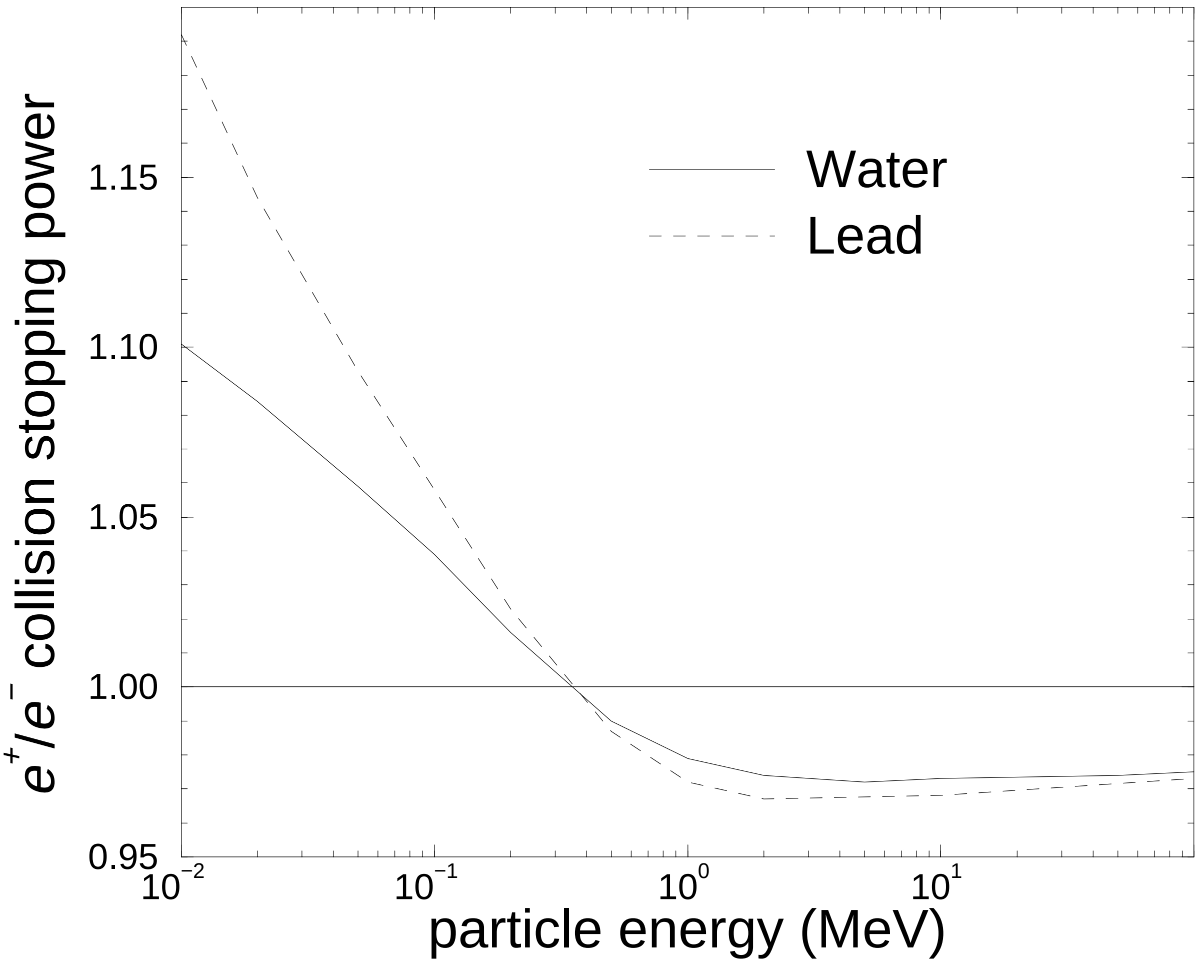

Figure 13.5: Positron/electron collision stopping power.

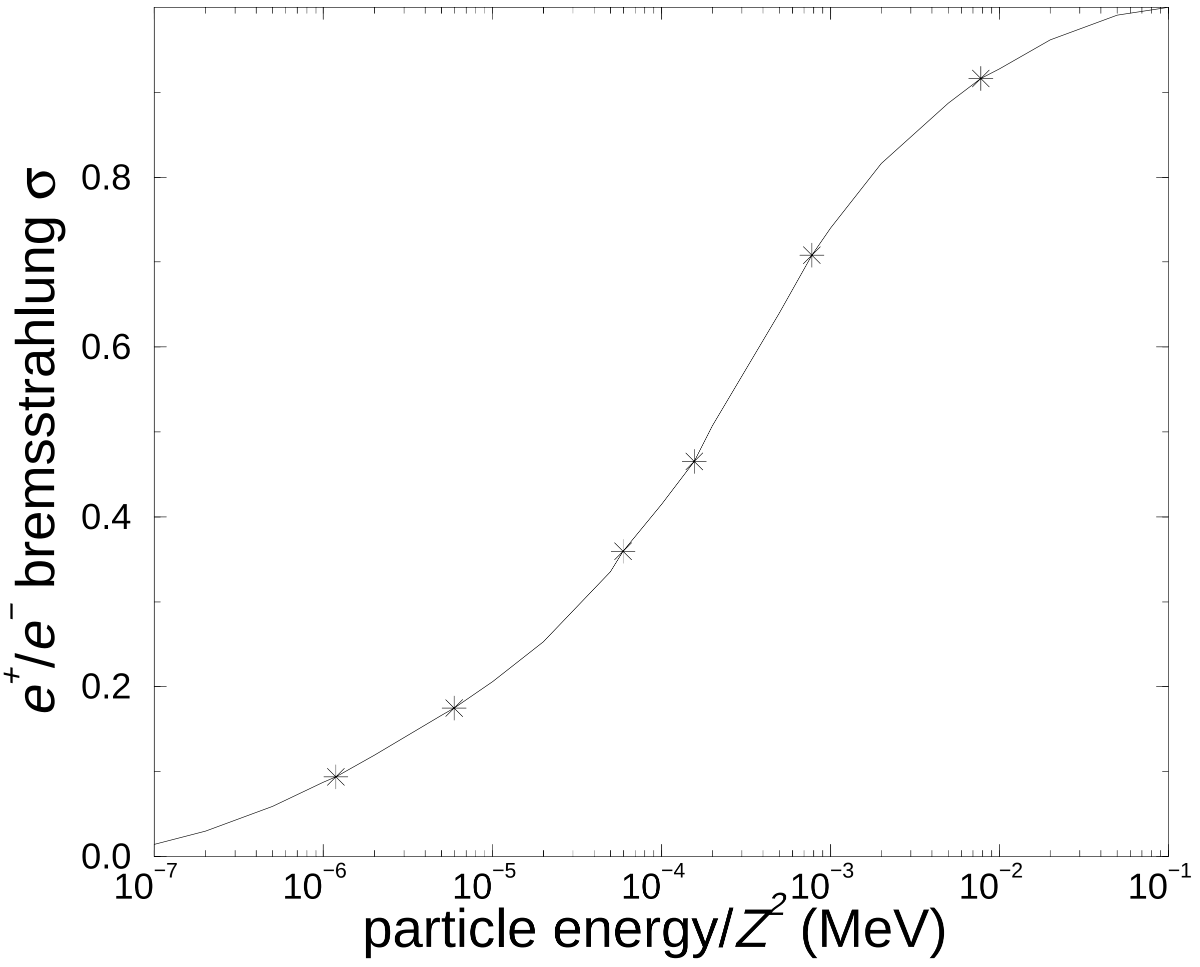

Figure 13.6: Positron/electron bremsstrahlung cross section.

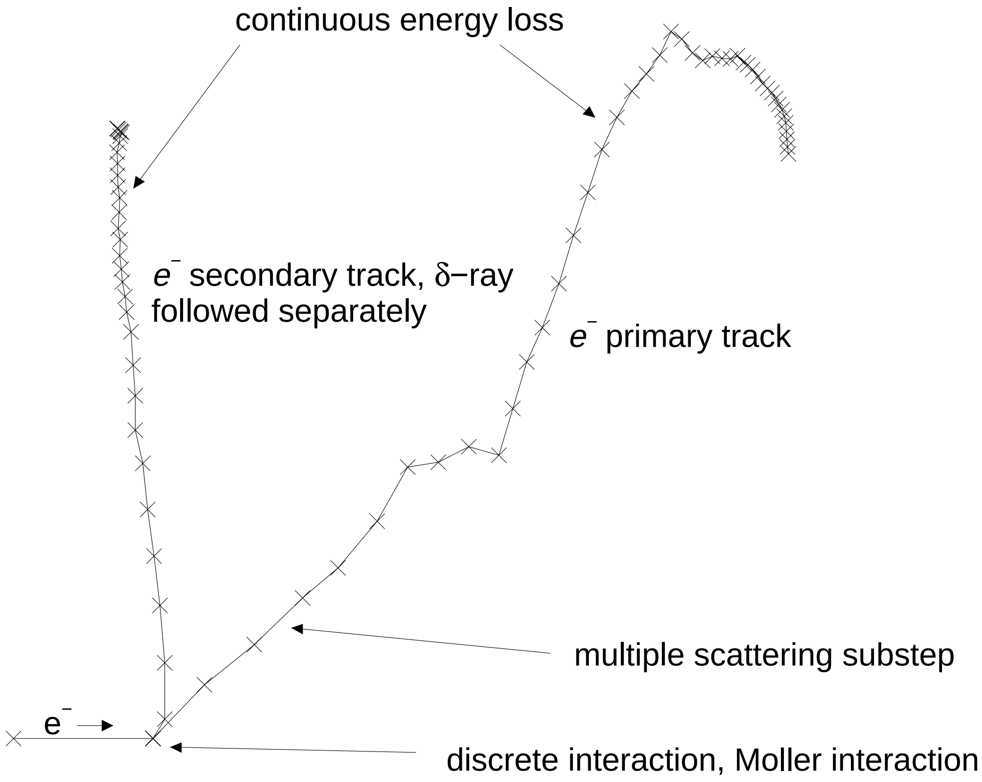
continuous energy loss
e^- secondary track, δ−ray
followed separately
e^- primary track
multiple scattering substep
e^-
discrete interaction, Moller interaction

Figure 13.7: A typical electron track simulation. The vertical scale has been exaggerated somewhat.

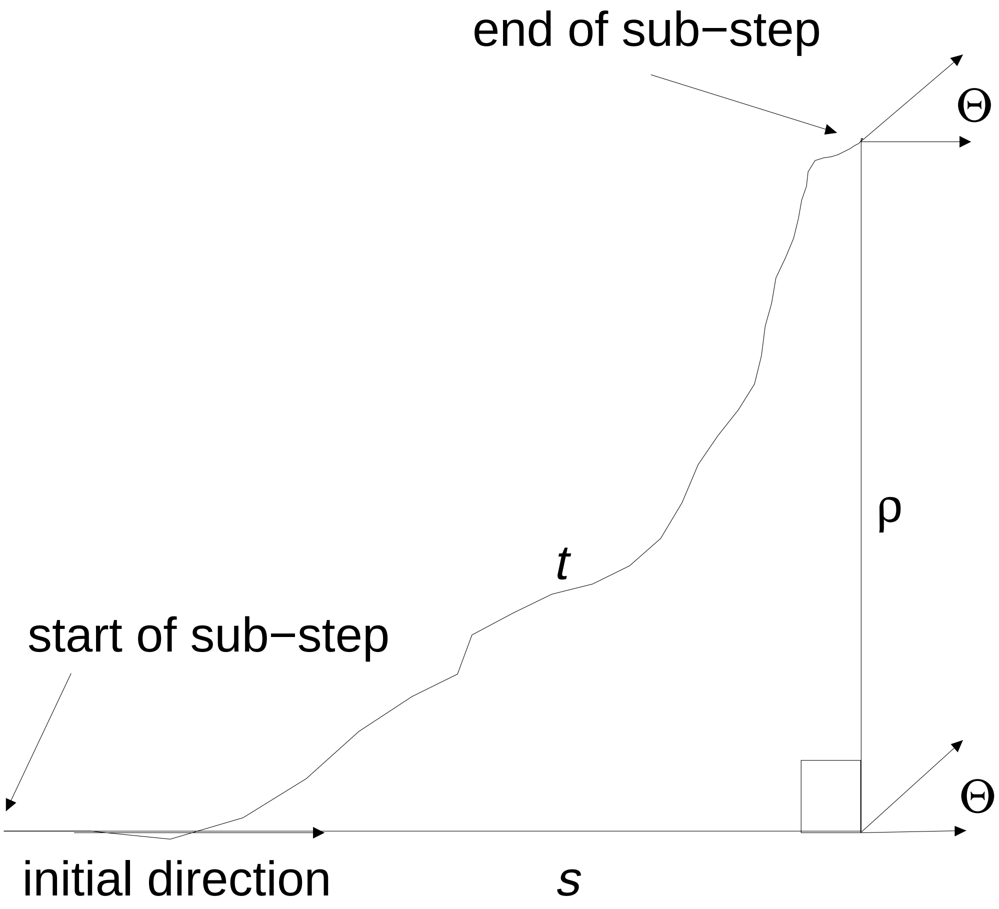

Figure 13.8: A typical multiple scattering substep.

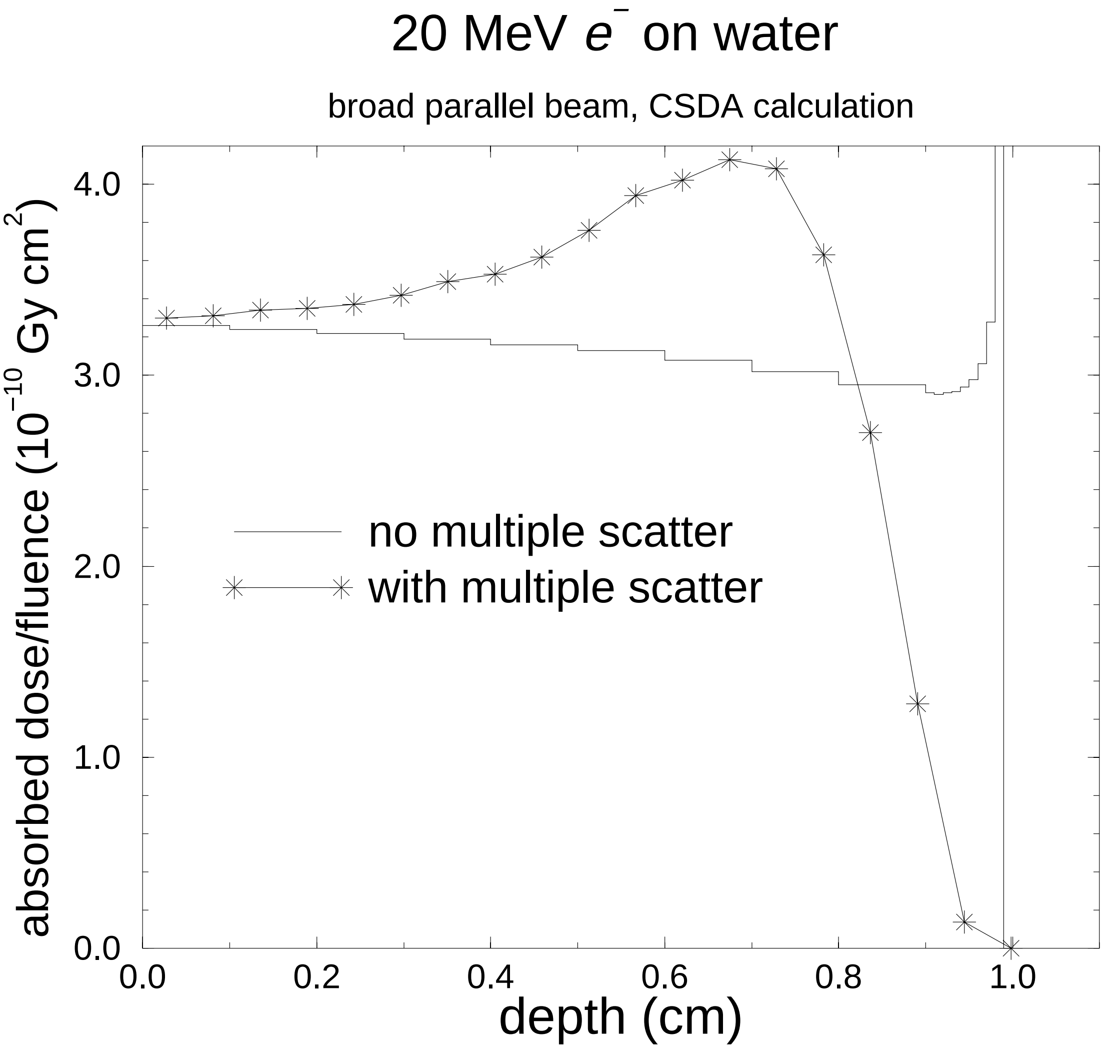

Figure 13.9: Depth-dose curve for a broad parallel beam (BPB) of 20 MeV electrons incident on a water slab. The histogram represents a CSDA calculation in which multiple scattering has been turned off, and the stars show a CSDA calculation which includes multiple scattering.

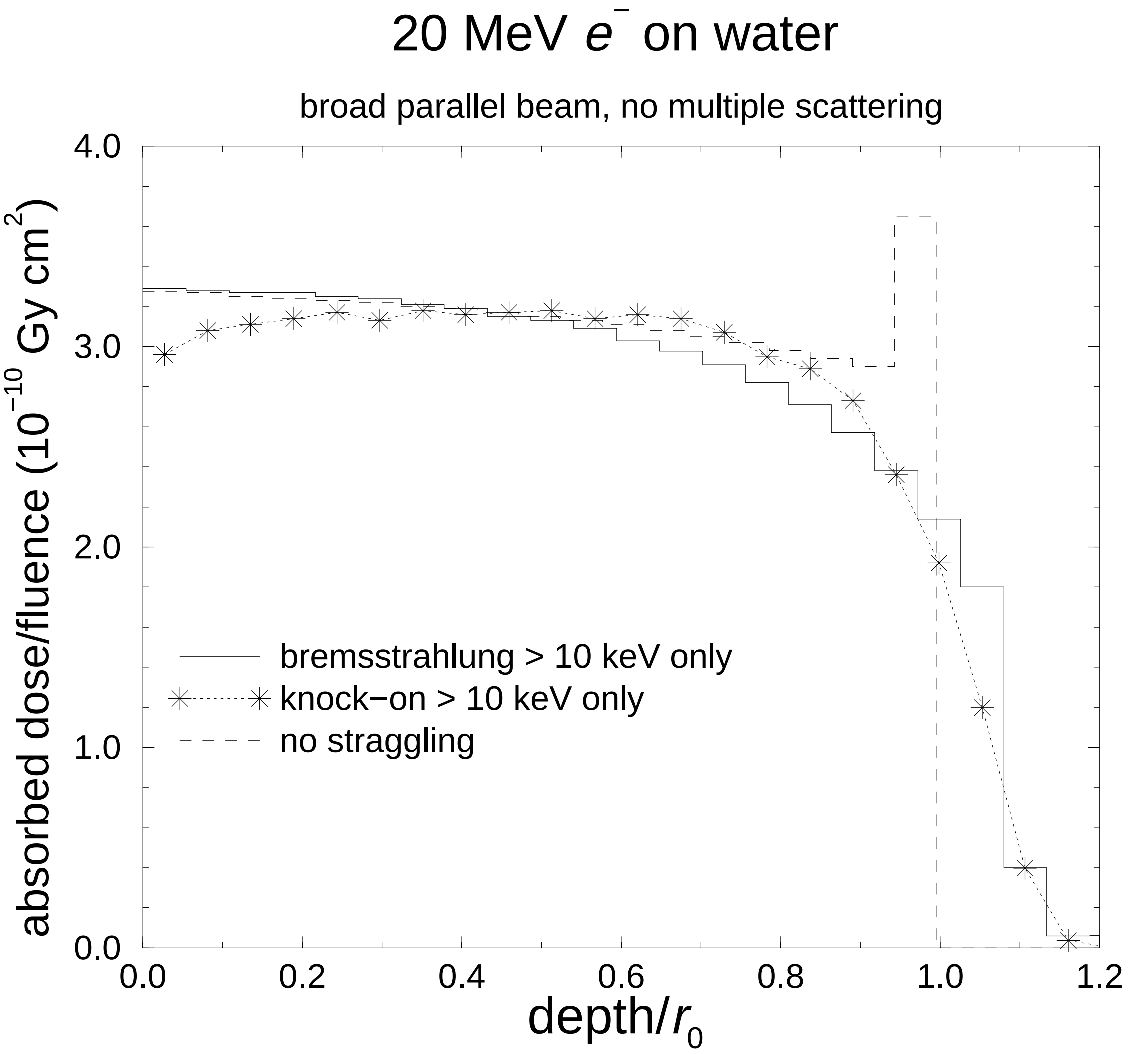

Figure 13.10: Depth-dose curves for a BPB of 20 MeV electrons incident on a water slab, but with multiple scattering turned off. The dashed histogram calculation models no straggling and is the same simulation as given by the histogram in fig. 13.9. Note the difference caused by the different bin size. The solid histogram includes energy-loss straggling due to the creation of bremsstrahlung photons with an energy above 10 keV. The curve denoted by the stars includes only that energy-loss straggling induced by the creation of knock-on electrons with an energy above 10 keV.

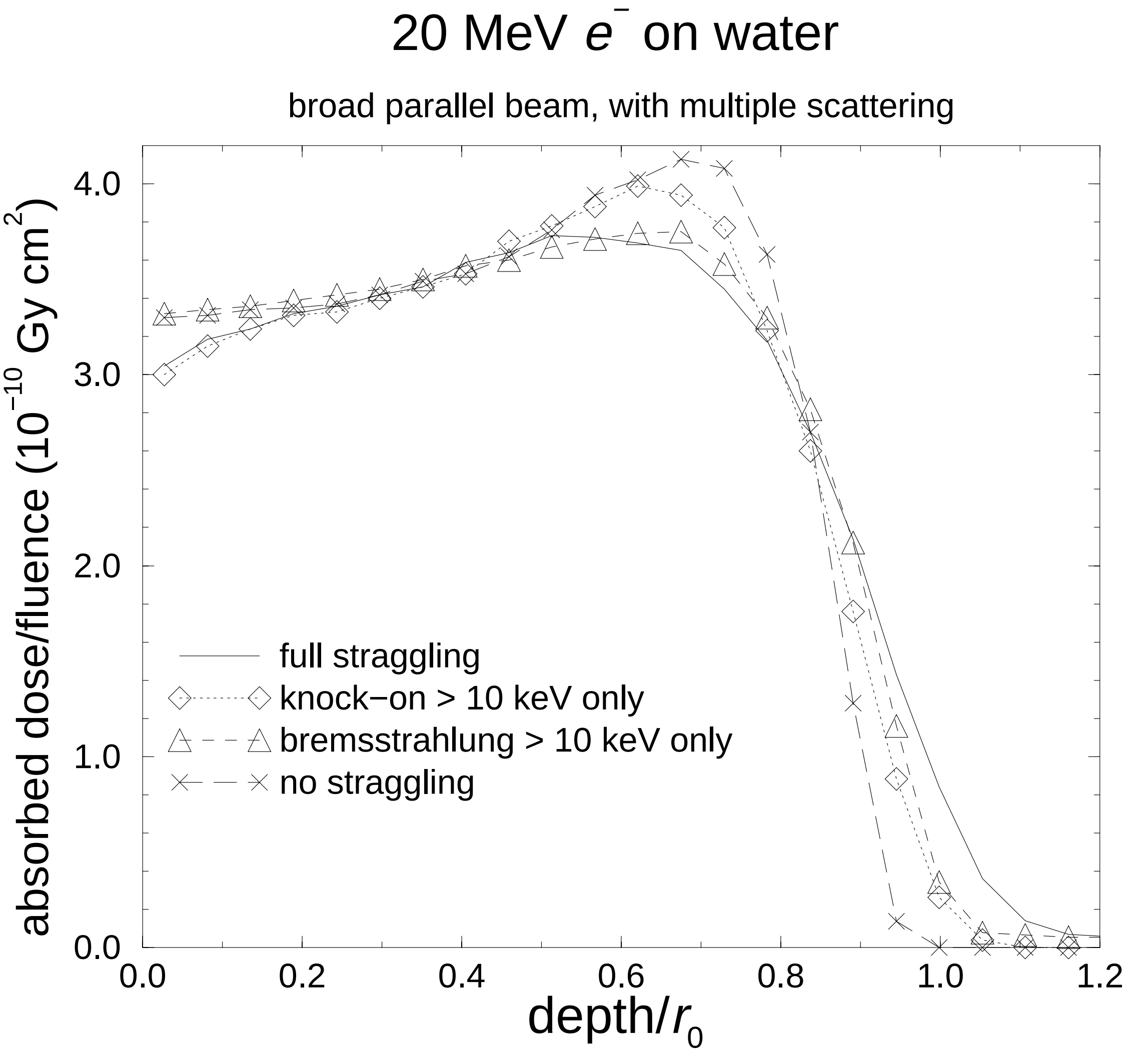

Figure 13.11: BPB of 20 MeV electrons on water with multiple scattering included in all cases and various amounts of energy-loss straggling included by turning on the creation of secondary photons and electrons above a 10 keV threshold.

13.5 Electron transport logic

Figure 13.12 is a schematic flow chart showing the essential differences between different kinds of electron transport algorithms. EGS4 is a "class II" algorithm which samples interactions discretely and correlates the energy loss to secondary particles with an equal loss in the energy of the primary electron (positron).

There is a close similarity between this flow chart and the photon transport flow chart. The essential differences are the nature of the particle interactions as well as the additional continuous energy-loss mechanism and multiple scattering. Positrons are treated by the same subroutine in EGS4 although it is not shown in fig. 13.12.

Imagine that an electron's parameters (energy, direction, etc.) are on top of the particle stack. (`STACK` is an array containing the phase-space parameters of particles awaiting transport.) The electron transport routine, picks up these parameters and first asks if the energy of this particle is greater than the transport cutoff energy, called `ECUT`. If it is not, the electron is discarded. (This is not to that the particle is simply thrown away! "Discard" means that the scoring routines are informed that an electron is about to be taken off the transport stack.) If there is no electron on the top of the stack, control is given to the photon transport routine. Otherwise, the next electron in the stack is picked up and transported. If the original electron's energy was great enough to be transported, the distance to the next catastrophic interaction point is determined, exactly as in the photon case. The multiple scattering step-size t is then selected and the particle transported, taking into account the constraints of the geometry. After the transport, the multiple scattering angle is selected and the electron's direction adjusted. The continuous energy loss is then deducted. If the electron, as a result of its transport, has left the geometry defining the problem, it is discarded. Otherwise, its energy is tested to see if it has fallen below the cutoff as a result of its transport. If the electron has not yet reached the point of interaction a new multiple scattering step is effected. This innermost loop undergoes the heaviest use in most calculations because often many multiple scattering steps occur between points of interaction (see fig. 13.7). If the distance to a discrete interaction has been reached, then the type of interaction is chosen. Secondary particles resulting from the interaction are placed on the stack as dictated by the differential cross sections, lower energies on top to prevent stack overflows. The energy and direction of the original electron are adjusted and the process starts all over again.

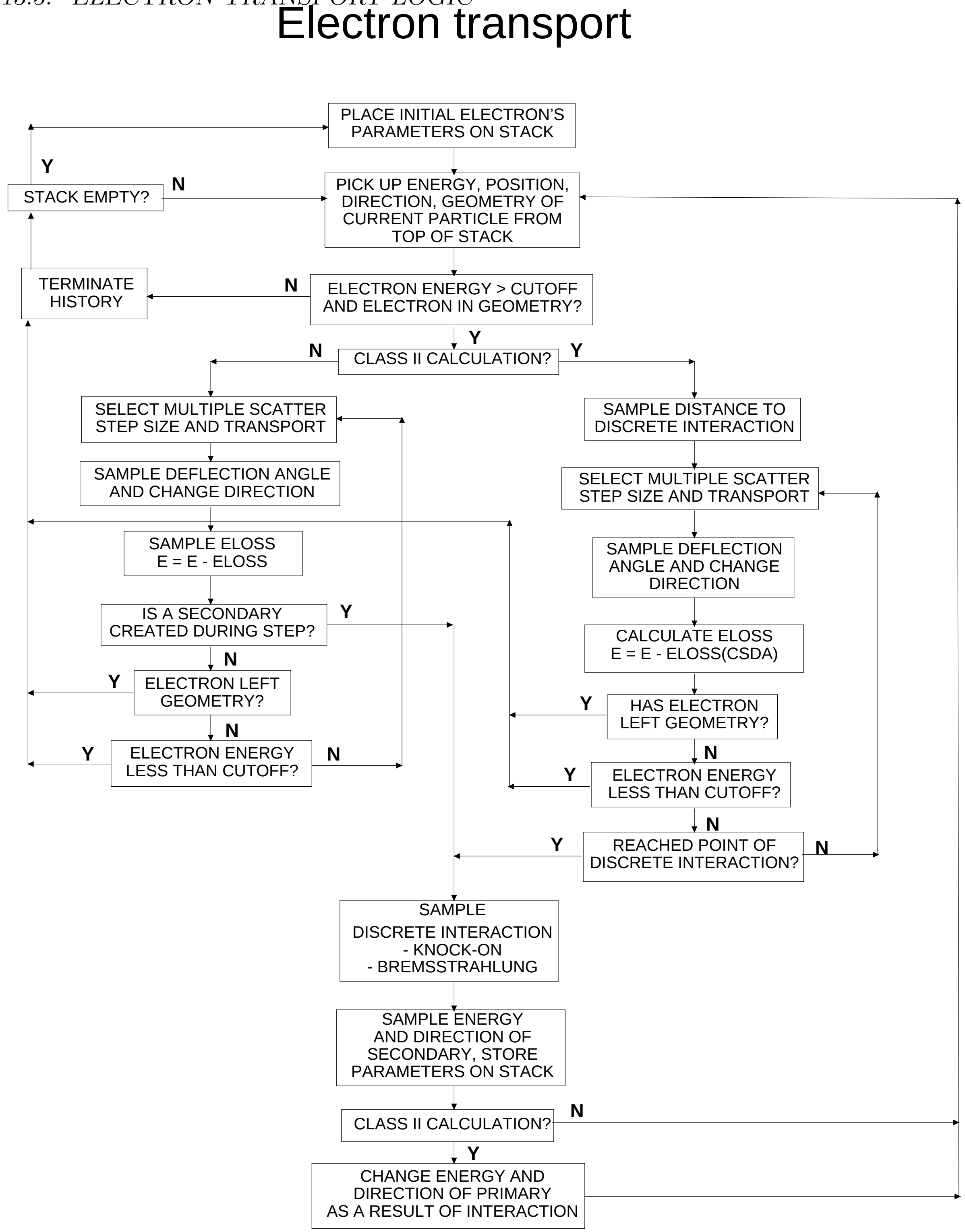
Electron transport
PLACE INITIAL ELECTRON'S PARAMETERS ON STACK
Y
STACK EMPTY?
N
PICK UP ENERGY, POSITION, DIRECTION, GEOMETRY OF CURRENT PARTICLE FROM TOP OF STACK
TERMINATE HISTORY
N
ELECTRON ENERGY > CUTOFF AND ELECTRON IN GEOMETRY?
Y
N
CLASS II CALCULATION?
Y
SELECT MULTIPLE SCATTER STEP SIZE AND TRANSPORT
SAMPLE DISTANCE TO DISCRETE INTERACTION
SAMPLE DEFLECTION ANGLE AND CHANGE DIRECTION
SELECT MULTIPLE SCATTER STEP SIZE AND TRANSPORT
SAMPLE ELOSS E = E - ELOSS
SAMPLE DEFLECTION ANGLE AND CHANGE DIRECTION
IS A SECONDARY CREATED DURING STEP?
Y
CALCULATE ELOSS E = E - ELOSS(CSDA)
N
Y
ELECTRON LEFT GEOMETRY?
Y
HAS ELECTRON LEFT GEOMETRY?
N
Y
ELECTRON ENERGY LESS THAN CUTOFF?
N
N
Y
ELECTRON ENERGY LESS THAN CUTOFF?
N
Y
REACHED POINT OF DISCRETE INTERACTION?
N
SAMPLE DISCRETE INTERACTION - KNOCK-ON - BREMSSTRAHLUNG
SAMPLE ENERGY AND DIRECTION OF SECONDARY, STORE PARAMETERS ON STACK
CLASS II CALCULATION?
N
Y
CHANGE ENERGY AND DIRECTION OF PRIMARY AS A RESULT OF INTERACTION

Figure 13.12: Flow chart for electron transport. Much detail is left out.

Bibliography

[AMB93] P. Andreo, J. Medin, and A. F. Bielajew. Constraints on the multiple scattering theory of Molière in Monte Carlo simulations of the transport of charged particles. *Med. Phys.*, 20:1315 – 1325, 1993.

[Ber92] M. J. Berger. ESTAR, PSTAR and ASTAR: Computer Programs for Calculating Stopping-Power and Ranges for Electrons, Protons, and Helium Ions. *NIST Report NISTIR-4999 (Washington DC)*, 1992.

[Bet30] H. A. Bethe. Theory of passage of swift corpuscular rays through matter. *Ann. Physik*, 5:325, 1930.

[Bet32] H. A. Bethe. Scattering of electrons. *Z. für Physik*, 76:293, 1932.

[Bet53] H. A. Bethe. Molière's theory of multiple scattering. *Phys. Rev.*, 89:1256 – 1266, 1953.

[BHNR94] A. F. Bielajew, H. Hirayama, W. R. Nelson, and D. W. O. Rogers. History, overview and recent improvements of EGS4. *National Research Council of Canada Report PIRS-0436*, 1994.

[Bie94] A. F. Bielajew. Plural and multiple small-angle scattering from a screened Rutherford cross section. *Nucl. Inst. and Meth.*, B86:257 – 269, 1994.

[Blo33] F. Bloch. Stopping power of atoms with several electrons. *Z. für Physik*, 81:363, 1933.

[Bri86] J. Briesmeister. MCNP—A general purpose Monte Carlo code for neutron and photon transport, Version 3A. *Los Alamos National Laboratory Report LA-7396-M (Los Alamos, NM)*, 1986.

[Bri93] J. F. Briesmeister. MCNP—A general Monte Carlo N-particle transport code. *Los Alamos National Laboratory Report LA-12625-M (Los Alamos, NM)*, 1993.

[Bri97] J. F. Briesmeister. MCNP—A general Monte Carlo N-particle transport code. *Los Alamos National Laboratory Report LA-12625-M, Version 4B (Los Alamos, NM)*, 1997.

[BS64] M. J. Berger and S. M. Seltzer. Tables of energy losses and ranges of electrons and positrons. *NASA Report SP-3012 (Washington DC)*, 1964.

[Eyg48] L. Eyges. Multiple scattering with energy loss. *Phys. Rev.*, 74:1534, 1948.

[GS40a] S. A. Goudsmit and J. L. Saunderson. Multiple scattering of electrons. *Phys. Rev.*, 57:24 – 29, 1940.

[GS40b] S. A. Goudsmit and J. L. Saunderson. Multiple scattering of electrons. II. *Phys. Rev.*, 58:36 – 42, 1940.

[Hal89] J. Halbleib. Structure and Operation of the ITS code system. In T.M. Jenkins, W.R. Nelson, A. Rindi, A.E. Nahum, and D.W.O. Rogers, editors, *Monte Carlo Transport of Electrons and Photons*, pages 249 – 262. Plenum Press, New York, 1989.

[Hei54] W. Heitler. The quantum theory of radiation. *(Clarendon Press, Oxford)*, 1954.

[HKM$^+$92] J. A. Halbleib, R. P. Kensek, T. A. Mehlhorn, G. D. Valdez, S. M. Seltzer, and M. J. Berger. ITS Version 3.0: The Integrated TIGER Series of coupled electron/photon Monte Carlo transport codes. *Sandia report SAND91-1634*, 1992.

[HM84] J. A. Halbleib and T. A. Mehlhorn. ITS: The integrated TIGER series of coupled electron/photon Monte Carlo transport codes. *Sandia Report SAND84-0573*, 1984.

[ICR84] ICRU. Stopping powers for electrons and positrons. ICRU Report 37, ICRU, Washington D.C., 1984.

[KB98] I. Kawrakow and A. F. Bielajew. On the representation of electron multiple elastic-scattering distributions for Monte Carlo calculations. *Nuclear Instruments and Methods*, B134:325 – 336, 1998.

[KM59] H. W. Koch and J. W. Motz. Bremsstrahlung cross-section formulas and related data. *Rev. Mod. Phys.*, 31:920 – 955, 1959.

[Mol47] G. Z. Molière. Theorie der Streuung schneller geladener Teilchen. I. Einzelstreuung am abgeschirmten Coulomb-Field. *Z. Naturforsch*, 2a:133 – 145, 1947.

[Mol48] G. Z. Molière. Theorie der Streuung schneller geladener Teilchen. II. Mehrfach- und Vielfachstreuung. *Z. Naturforsch*, 3a:78 – 97, 1948.

[NHR85] W. R. Nelson, H. Hirayama, and D. W. O. Rogers. The EGS4 Code System. Report SLAC–265, Stanford Linear Accelerator Center, Stanford, Calif, 1985.

[Sel89] S. M. Seltzer. An overview of ETRAN Monte Carlo methods. In T.M. Jenkins, W.R. Nelson, A. Rindi, A.E. Nahum, and D.W.O. Rogers, editors, *Monte Carlo Transport of Electrons and Photons*, pages 153 – 182. Plenum Press, New York, 1989.

[Sel91] S. M. Seltzer. Electron-photon Monte Carlo calculations: the ETRAN code. *Int'l J of Appl. Radiation and Isotopes*, 42:917 – 941, 1991.

[SSB82] R. M. Sternheimer, S. M. Seltzer, and M. J. Berger. Density effect for the ionization loss of charged particles in various substances. *Phys. Rev.*, B26:6067, 1982.

[Tsa74] Y. S. Tsai. Pair Production and Bremsstrahlung of Charged Leptons. *Rev. Mod. Phys.*, 46:815, 1974.

[Win87] K. B. Winterbon. Finite-angle multiple scattering. *Nuclear Instruments and Methods*, B21:1 – 7, 1987.

Problems

1.

Chapter 14

Electron step-size artefacts and PRESTA

In the first half of this chapter we shall discuss electron step-size artefacts, the reasons for calculation of spurious results under some circumstances, and simple ways by which these calculational anomalies may be avoided. In the second half of the chapter, we shall discuss a more sophisticated electron transport algorithm, called PRESTA, which solves this problem of step-size dependence in most cases.

The chapter will proceed within the context of the EGS4 code [NHR85] although the ideas put forth apply to all electron transport codes which use condensed-history methods. Calculations which signal the existence of step-size anomalies will be presented along with the improvements to the electron transport algorithm which were used to circumvent the problem.

14.1 Electron step-size artefacts

14.1.1 What is an electron step-size artefact?

An electron step-size artefact is characterized by the dependence of some calculated result upon arbitrary "non-physics" parameters of the electron transport. This is illustrated by the example given in fig. 16.1. In this example, 1 MeV electrons were incident normally upon a $3r_0/2$ thick slab of water. The quantity r_0 is the range calculated in the continuous-slowing-down approximation (CSDA). The energy deposited between $r_0/2$ and r_0 was scored. Two ways of calculating the energy deposition are depicted. The first, (EGS (with PLC), lower dashed line) is the default EGS calculation. "PLC" stands for path-length correction, which includes the effect of electron path curvature for each electron step. The second, (EGS (no

Figure 14.1: The relative energy deposit from 1 MeV electrons incident normally on a $3r_0/2$ slab of water. The energy deposited between $r_0/2$ and r_0 is shown. The upper dashed line is an EGS calculation without the electron step-size shortened by ESTEPE and without path-length corrections (PLC's). The lower dashed line is an EGS calculation without ESTEPE control and including the default PLC employed by EGS.

PLC), upper dashed line) neglects this correction. Rogers [Rog84b] added an electron step-size limit, ESTEPE, the maximum allowable fractional kinetic energy loss per electron step to "continuous" energy-loss processes in order to obtain better agreement in the low-energy region. One notices a dramatic increase in the "with PLC" curve with smaller ESTEPE and a commensurate decrease in the "no PLC" curve. Why should a calculated result depend so strongly on an arbitrary parameter such as electron step-size unless some basic constraints of the underlying theory are being violated? What is the role of path-length corrections? Does the electron transport algorithm have enough physics' content to accurately simulate electron transport? An even more important question is "*What is the correct answer?*". (If a correct answer is to be obtained for a case that exhibits step-size dependence, it is always found at smaller step-sizes within certain constraints that we shall discuss later.)

As another example of dramatic step-size effects, consider the irradiation geometry depicted in fig. 16.2. In this case, a 1 MeV zero-area beam of electrons was incident on the center of the end of an air tube which was 2 mm in diameter and 20 cm long. The results are plotted in fig. 16.3. The dose deposited in the air cylinder was scored as a function of SMAX, the maximum geometrical step-length allowed. This parameter was also introduced by Rogers [Rog84b] in adapting the EGS code to low-energy simulations. The default EGS simulation (equivalent to setting SMAX = 20 cm, the length of the tube) is wrong since most often the electrons only take one step through the tube, as depicted in fig. 16.4. All the "continuous" energy deposition associated with this step is deposited within the air tube resulting in too high a value being calculated. Reducing SMAX to 10 cm, half the length of the tube, almost halves the energy deposition, as seen in fig. 16.3. In this case, most of the electrons that are transported 10 cm immediately scatter out of the tube, as depicted in fig. 16.5. Further reduction of SMAX reduces the energy deposited to the tube as the electron transport simulation becomes more and more accurate. Finally, a flat region of "convergence" is obtained in the vicinity of 0.2 to 1.0 cm, a scale of magnitude comparable to the diameter of the tube. As seen in fig. 14.6, the small transport steps allow the electron to escape the tube or be transported down it, in accord with the random selection of the

Figure 14.2: The irradiation geometry of the "thin tube" simulation. A zero-area beam of 1 MeV electrons was incident on the center of the end of a 2 mm diameter, 20 cm long tube of air.

Figure 14.3: The relative dose deposited in the air cylinder is plotted as a function of SMAX. In the “default EGS” case, the electrons are usually transported the length of the tube resulting in an anomalously high calculated dose to the tube.

Figure 14.4: In the default EGS calculation the electrons most often travel the length of the tube. Note that the vertical scale in this and the next two figures is greatly exaggerated. The tube is actually 2 mm in diameter and 20 cm long. The ×’s mark the end-points of each electron step.

Figure 14.5: If SMAX is reduced to 10 cm, half the length of the tube, then the electrons that are transported 10 cm usually scatter out of the tube immediately.

multiple scattering angle for each step. In this region, the transport is being simulated more or less accurately.

Figure 14.6: When the transport steps are shortened to a length comparable to the diameter of the tube, the electron may or may not scatter out of the tube, obeying the probabilistic laws of multiple scattering.

At step-sizes in the vicinity of 1 mm and smaller we observe another artefact in fig. 16.3. We again notice anomalously high results. The reason for the occurrence of this artefact has to do with the minimum step-size that can be accommodated by the multiple scattering formalism used by the EGS code. (EGS uses the Molière formalism [Mol47, Mol48] as expressed by Bethe [Bet53].) At these smaller step-sizes multiple scattering formalism should be replaced by a “few-scattering” or “single-scattering” formalism. EGS does not do this but rather “switches off” the multiple scattering formalism and no elastic electron-electron or electron-nucleus scattering is modelled. Once more the electrons are transported in straight lines down the length of the tube. We must, therefore, qualify a statement expressed earlier in the chapter. If a correct answer is to be obtained with the EGS code for a case that exhibits a step-size dependence, it is obtained by using small step-sizes with the proviso that the multiple scattering is not “switched off” for a substantial number of the electron transport steps. The various limits on transport step-size will be discussed in more detail later in the chapter.

The previous example was contrived to show large changes in calculated results with step-size. It represents the extreme limit of what the EGS code is capable of. As a final example we show a large step-size dependence for a case where the electrons are almost in a state of equilibrium. This is the case of a thick-walled ion chamber exposed to ^{60}Co photons. We show, in fig. 14.7, the variation with ESTEPE of the calculated response of a 0.5 g/cm^2 carbon-walled

Figure 14.7: Calculated response of a thick-walled carbon chamber (0.5 g/cm^2 carbon walls, 2 cm diameter, 2 mm thick cylindrical air cavity), exposed to 1.25 MeV photons incident normally on a flat circular end.

ion chamber with a cylindrical air cavity 2 mm in depth and 2 cm in diameter exposed to a monoenergetic beam of 1.25 MeV photons incident normally upon one of the flat ends. The results are normalized to the theoretical predictions of Spencer-Attix theory [SA55] corrected for photon attenuation, photon scatter and electron drift affects [BRN85, RBN85]. According to the theorem of Fano [Fan54], the electron fluence in the chamber in the vicinity of the cavity is almost unperturbed by the presence of the cavity in this situation. (Strictly speaking, Fano's theorem only applies to density changes in one medium. However, carbon and air are not too dissimilar except for their densities and Fano's theorem may be applied with negligible error.) The electrons in this simulation are almost in complete equilibrium. Non-equilibrium effects requiring corrections to Spencer-Attix theory amount to only a few percent of the total response. Why then, should the electron step-size play such a critical role in a simulation where electron transport does not matter a great deal to the physics? We observe, in fig. 14.7 a step-size variation of about 40% when ESTEPE is changed from 1% to 20%! To answer this question requires some closer examination of the various elements of electron transport.

14.1.2 Path-length correction

Figure 14.8: A 10 MeV electron being transported through a 1 cm slab of water as simulated by EGS in its default configuration (no ESTEPE or SMAX control, note that the electron takes only one step to cross the water slab) and with an ESTEPE of 10, 5, 2, and 1%.

To illustrate the concept of path-length correction, we consider the example of 10 MeV electrons incident normally upon a 1 cm slab of water. The top curve in fig. 14.8 depicts a typical EGS electron transport step through this slab with the EGS system used in its default configuration (no ESTEPE or SMAX control). Note that the electron went through in only one step. The other curves in fig. 14.8 depict similar histories except that ESTEPE has been adjusted to 10, 5, 2, or 1%. As ESTEPE gets smaller and smaller, the electron tracks begin to "look" like real electron tracks, similar to those that one would observe, for example, in bubble chamber photographs. We know that electron steps are curved, as depicted in the previous figures. Must we use exceedingly small electron steps to calculate accurately in Monte Carlo simulations? The answer depends upon the application. If one is interested in accurate physical "pictures" of electron tracks, then short step-sizes, consistent with the resolution desired, must be used. However, imagine that we are only interested in calculating the energy deposited in this slab. Then, considering fig. 14.8 with the realization that the energy deposited is proportional to the total curved path, it would be possible to simulate passage through this slab using only one step *if one could accurately correct for the actual path-length the electron would have travelled if one had used very small steps.*

Figure 14.9 depicts the relationship between the total curved path of a step and its straight-line path in the direction of motion at the start of the step. For a given value of the curved

Figure 14.9: A pictorial representation of the total curved path-length of an electron step, t, and the straight-line path-length, s, in the direction of motion at the beginning of the step. The average lateral deflection of a transport step, ρ, is related to t. The displacements s and ρ are mutually orthogonal.

path-length, t, the average straight-line path in the starting direction of motion, s, are related by eq. 17.1 which has been attributed to Lewis [Lew50],

$$s = \int_0^t dt' \langle cos\Theta(t')\rangle, \tag{14.1}$$

where $\Theta(t')$ is the multiple scattering angle as a function of the actual curved path-length along the path, t', and the average value, $\langle\rangle$, is to be computed using the probability distribution of any multiple scattering theory. Several strategies have been developed for calculating s using the Lewis equation. Yang [Yan51] advocated an expansion of eq. 17.1 to second order in Θ and the use of a small-angle multiple scattering theory to compute the average value. This is the strategy employed in the EGS code where the Fermi-Eyges multiple scattering theory [Eyg48] is used to compute the average value. (As mentioned previously, the multiple scattering in EGS is performed using Bethe's formulation of the Molière theory.) Unfortunately, this approach has been shown to produce path-length corrections, $(t-s)/s$, a factor of 2 too high [HW55, BR87]. Berger [Ber63] advocated the relation,

$$s = \frac{1}{2}t[1+\cos(\Theta(t))], \tag{14.2}$$

and he showed that s calculated using this equation agrees with that calculated using eq. 17.1 in the limit of small angle if the multiple scattering theory of Goudsmit and Saunderson [GS40a, GS40b] is used. Bielajew and Rogers [BR87] expanded eq. 17.1 to 4^{th} order in Θ and evaluated the average value using Bethe's version of Molière's multiple scattering theory. They showed that this approach and s calculated using eq. 17.3 agree, even for large average scattering angles of the order of a radian. The proof that this approach is valid is given later in the chapter.

The path-length correction can be quite large, as seen in fig. 14.10, where the path-length correction, $(t-s)/s$, in water is plotted versus electron kinetic energy for various step-sizes as measured by ESTEPE. If one wishes to reduce computing time by using large electron

Figure 14.10: The path-length correction in water versus kinetic energy for various step-sizes as measured by ESTEPE. The line $t = t_{max}$ shows the maximum step-size allowed by the Molière theory.

steps, then one must correct for path-length curvature. The larger the step, the greater the correction. Greater corrections are needed for lower energies as well. Recall that the path-length correction used by EGS is about a factor of 2 too high. This fact is almost entirely responsible for the step-size artefact seen in fig. 16.1. Too much curvature correction resulted in too much energy being deposited in the first $r_0/2$ slab and, by conservation of energy, too little in the second. In the "no PLC" case, the opposite prevailed. The *failure* to account for electron path-length curvature leads to less energy deposition in the upstream slab and too much in the downstream one. As the step-size is reduced, however, the electron tracks are modelled more and more correctly, relying on multiple scattering selected for each of the steps for the development of curvature of the tracks. If one uses a correct path-length correction, such as that proposed by Berger [Ber63] or Bielajew and Rogers [BR87], then most of the step-size artefact vanishes, as exhibited in fig. 14.11. The residual step-size dependence has to do with other neglected features of electron transport that we have yet to discuss.

Figure 14.11: The deep energy deposition problem described for fig. 16.1. In this case, a "proper" (*i.e.* demonstrably correct) path-length correction is used to eliminate most of the step-size dependence.

We now return to the ion chamber simulation and see what effect the use of a correct path-length correction has. Figure 14.12 shows the improvement of the ion chamber calculation, a reduction in the step-size dependence with use of a proper path-length correction. Yet, there still remains a considerable dependence on ESTEPE. Some physics must be missing

Figure 14.12: The ion chamber calculation described for fig. 14.7 with a reduction of the step-size dependence with the use of a “proper” path-length correction. A significant step-size dependence remains.

from this simulation!

14.1.3 Lateral deflection

Returning to fig. 14.8, we see that as the step-size is made smaller and the electron histories are simulated with increasing accuracy, not only does the electron path acquire curvature, it is also deflected laterally. If one is faced with a simulation in which lateral transport is important (for example, the air tube of figs. 16.2 and 16.3), but one wishes to use as few electron steps as possible, one ought to account for the the lateral deflection *during the course of each electron step.* If we use a sufficiently small step-size, this lateral deflection will occur naturally as the multiple scattering angle selected for each electron step deflects the electron, accomplishing the lateral transport. We saw that in the example of the air tube, if the step-size was restricted to be of the order of the diameter of the tube, the effects of lateral transport were incorporated properly in the simulation. Therefore, if one wishes to use fewer transport steps in a simulation of this nature, a more sophisticated approach is needed.

Figure 14.9 illustrates the basic concept of the lateral deflection. An average lateral deflection, ρ, is associated with an electron transport step characterized by the total curved path of the step, t. Berger [Ber63] has provided a method that correlates ρ with t and the multiple scattering angle, Θ, for the electron step,

$$\rho = \frac{1}{2} t \sin \Theta(t). \tag{14.3}$$

This is called the "lateral correlation" because the displacement, ρ, is correlated to the multiple scattering angle [Ber63]. The proof that this prescription is valid will be given later. Figure 14.13 shows that this correction is large for large step-sizes and small energies. We shall show, in another section, evidence of the reduction of step-size artefacts through the use of Berger's lateral correlation algorithm.

14.1.4 Boundary crossing

A general Monte Carlo method should be able to simulate electron trajectories in complex geometries. The condensed-history technique, whether the multiple scattering is performed through the use of the theories of Fermi-Eyges, Molière, or Goudsmit-Saunderson, is limited by the fundamental constraints of these theories. These theories are strictly applicable in only infinite or semi-infinite geometries. Some theories (*e.g.* Fermi-Eyges, Molière) are applicable only for small average scattering angles as well. It would be far too complex to construct a multiple scattering theory that applies for all useful geometries. In particular, how should a Monte Carlo electron transport algorithm treat the approach and retreat from

Figure 14.13: The average angular correlation versus electron kinetic energy for various step-sizes as measured by ESTEPE. The reader is referred to ref. [BR87] for the calculational details used in the construction of this figure.

arbitrarily shaped boundaries yet still not violate the basic constraints of the underlying theories? Unless multiple scattering theories become much more sophisticated, there is only one solution — *shorten the electron steps in the vicinity of boundaries so that for a majority of electron steps in the simulation, any part of the total curved path is restricted to a single medium.* In other words, the underlying theories rely upon the particle transport taking place in an infinite, or semi-infinite medium. Therefore, in the vicinity of a boundary, the electron step should be shortened enough so that the underlying theory is not violated, at least for most of the transport steps. The details of how this boundary crossing is accomplished is very much code-dependent. However, the above "law" should apply for all condensed-history Monte Carlo methods. The details of how this can be accomplished with the EGS code will be given later.

14.2 PRESTA

14.2.1 The elements of PRESTA

So far we have discussed electron step-size artefacts and how they can be circumvented by shortening the electron transport step-size. The occurrences of artefacts were related to a shortcoming in, or the lack of, a path-length correction, the lack of lateral transport during the course of an electron step, or the abuse of the basic constraints of the multiple scattering theory in the vicinity of boundaries describing the geometry of the simulation. PRESTA, the Parameter Reduced Electron-Step Transport Algorithm [BR87], attempts to address these shortcomings with the EGS code. The general features of PRESTA are applicable to all condensed-history codes. The fine details, only a few of which we shall discuss, are not. Before plunging ourselves into the features of PRESTA, we return to the examples dealt with earlier in the chapter and show how PRESTA handles the difficulties.

We have discussed the energy deposition in the middle of three $r_0/2$ slabs due to 1 MeV electrons. Recall that in fig. 16.1 we saw large step-size artefacts produced by the EGS code that could be "healed" by using short step-sizes. Later in fig. 14.11 we cured most of the problem by using a correct path-length correction. In fig. 14.14, we also include lateral deflections and a careful boundary crossing algorithm, (the remaining components of PRESTA), and all residual variation with step-size disappear. In this example, it was the correct path-length correction which was responsible for most of the improvement. The correct path-length correction method is one of the major components of PRESTA.

In the ion chamber simulation of figs. 14.7 and 14.12, the improvement of ion chamber response was quite dramatic but still incomplete. The evidence of step-size dependence was still quite strong. Once PRESTA is used for the simulation, however, the step-size artefact vanishes, as evidenced in fig. 14.15. It is the inclusion of lateral transport that is responsible for the remaining improvement in this case.

Figure 14.14: The energy deposition to the middle of three $r_0/2$ water slabs due to 1 MeV electrons. This simulation was discussed previously in figs. 16.1 and 14.11. When PRESTA is used, all evidence of step-size dependence vanishes.

Figure 14.15: The ion chamber response calculation visited already in figs. 14.7 and 14.12. The use of PRESTA virtually eliminates any step-size dependence in this calculation.

Finally, the improvement in the air tube calculation of fig. 16.3 is shown in fig. 14.16. In this case, it is the boundary crossing algorithm that is almost entirely responsible for the improvement.

Therefore, path-length correction, lateral deflection and a careful boundary crossing algorithm are essential elements of a general purpose, accurate electron transport algorithm. It remains to be proven in a more rigorous fashion that these components are physically valid in a more general context other than the examples given. Otherwise the improvements may be fortuitous. To do this requires a brief introduction to the Molière theory, specifically on the limits on electron step-size demanded by this multiple scattering formalism.

14.2.2 Constraints of the Molière Theory

In this section we briefly discuss the physical constraints of the Molière multiple scattering theory. Rather than present many mathematical formulae, we concentrate on graphical representations of the various limits. For further detail, the reader is encouraged to examine refs. [NHR85, BR87] for the implementation of the Molière theory in the EGS code. The original papers are enlightening [Mol47, Mol48], and the exposition of Molière's theory by Bethe [Bet53] is a true classic of scientific literature.

Figure 14.16: The air tube calculation of fig. 16.3 is dramatically improved by the use of PRESTA. The label blcmin= 1.989 refers to a parameter that controls the boundary crossing algorithm. This point is discussed later.

The Molière theory is constrained by the following limits:

- The angular deflection is "small". (The Molière theory is couched in a small angle approximation.) Effectively, this constraint provides the upper limit on step-size.
- The theory is a *multiple* scattering theory, that is, many atomic collision participate to cause the electron to be deflected. Effectively, this constraint provides the lower limit on step-size.
- The theory applies only in infinite or semi-infinite homogeneous media. This constraint provides the motivation for treating the electron transport very carefully in the vicinity of interfaces.
- Energy loss is not built into the theory.

Bethe [Bet53] carefully compared the multiple scattering theories of Molière [Mol47, Mol48] and Goudsmit-Saunderson [GS40a, GS40b]. The latter theory does not resort to any small-angle approximation. Bethe showed that the small angle constraint of the Molière theory can be expressed as an equation that yields the maximum step-size [NHR85, BR87]. Below this limit, the two theories are fundamentally the same. This upper limit is used by PRESTA. (The default EGS upper limit is actually about 0.8 of the PRESTA limit.) Bethe's upper limit is plotted in fig. 14.17 as the curve labelled t_{max}. Also plotted in this figure is the CSDA range [BS83]. We note that at larger energies, greater than about 3 MeV in water, the CSDA range is a more stringent restriction on electron step-size. This means that for large energies, step-sizes can be quite large, up to the range of the electron. However, one must recall that the Molière theory does not incorporate energy loss directly. Therefore, if we wish to approach the upper limit on step-size, we must treat the energy loss part of the problem carefully. This topic will be discussed in a later section.

There is a critical parameter in the Molière theory, Ω_0, that can be interpreted as the number of atoms that participate in the multiple scattering. Molière considered his development to be valid for $\Omega_0 \geq 20$. It has been found that sensible results can be obtained for $\Omega_0 \geq e$ [BR87]. The lower limit, $\Omega_0 = e$, represents the "mathematical" limit below which Molière's formalism breaks down mathematically. It is interesting that Molière's theory can be "pushed" into the "few-scattering" regime and still produce reliable answers. We shall return to this point later. The minimum step-size, t_{min}, obeying $\Omega_0 = e$ is plotted versus electron kinetic energy in fig. 14.17 for water. We see in this figure, that the minimum and maximum step-sizes are the same at about 230 eV in water. Therefore, this represents the absolute minimum energy for which multiple scattering can be modelled using the Molière theory. (In this energy region, atomic binding effects begin to play an increasingly important role requiring the use of more sophisticated low-energy theories.) As the energy increases, so does the range over which the Molière theory is valid. The lower limit reaches an asymptotic bound at about 4×10^{-4} cm, while the upper limit continues upwards monotonically with

Figure 14.17: The minimum and maximum step-size limits of the Molière theory, t_{min} and t_{max} respectively. These limits are for water and the behavior for other materials can be obtained elsewhere [NHR85, BR87]. The dashed curve is the CSDA range [BS83].

increasing energy. Thus, for high energy, the applicable range in water extends from about 4 microns to the electron CSDA range.

In a previous section we discussed a type of artefact that can be problematic with the EGS code. That is, if one demands a step-size that is too short, EGS "turns off" the simulation of multiple scattering. We saw a dramatic example of this in fig. 16.3. Figure 14.18 compares t_{min} with step-sizes measured by various values of ESTEPE as calculated for water. Note that if one demands a step-size of 1% ESTEPE, then multiple scattering will not be simulated for electrons with energies less than about 40 keV. To circumvent this problem, PRESTA does not allow the ESTEPE restriction to reduce step-size below t_{min}.

Figure 14.18: Electron step-size is plotted versus kinetic energy for various values of ESTEPE and t_{min}. These curves apply for water. For other media, consult ref. [NHR85, BR87]. If one demands a 0.1% ESTEPE in water, then multiple scattering cannot be modelled using the Molière theory for electrons below about 500 keV.

The answer to the question, "Is the Molière theory valid between these upper and lower limits?", is a complicated one. The benchmarking of PRESTA can be construed as a verification of the consistency of the Molière theory. If the Molière theory contained any intrinsic step-size dependence, then so would the results calculated using PRESTA, barring some highly fortuitous coincidences. In the next few subsections, we examine all the components of PRESTA, trying the utmost to omit unnecessary complications.

14.2.3 PRESTA's path-length correction

In section 14.1.2 we discussed a new path-length correction. This method used the Lewis formula, eq. 17.1, expanded it to 4^{th} order in Θ, and evaluated the mean values using the Molière distribution functions [BR87]. We have seen impressive reductions in step-size dependences exhibited in figs. 14.11 and 14.12. It now remains to prove that this path-length correction is valid in more general applications. To this end, we modify our electron transport algorithm in the following fashion to conform with all the constraints of the Molière theory:

- Energy loss mechanisms are "switched off", including losses to "continuous" and "discrete" processes.

- Bounding surfaces of all kinds are eliminated from the simulations. The transport takes place in an infinite medium.

- The step-size constraints of the Molière theory are obeyed.

We performed the following simulations: An electron was set in motion in a given direction, which defines the z-axis for the problem. A history was defined by having the total curved path, summed over all electron steps, exactly equal to the Molière upper limit. This was achieved by choosing the step to be a divisor of t_{max}. That is, one simulation was done with $t = t_{max}$, another with $t = t_{max}/2$, another with $t = t_{max}/3$, $\ldots t_{max}/N$, where N is an integer. The quantity "scored" was the average displacement along the z-axis, $\langle z \rangle_N$, at the end of the history. The sum of the curved paths of the N steps always equals t_{max}. We note that lateral displacements play no role in this simulation because they would average out to zero. We argue that *if the path-length correction and the Molière theory are both consistent, then the $\langle z \rangle_N$'s should be independent of N, or equivalently, step-size independent.*

We show two extreme cases in figs. 14.19 and 14.20. The former, for 10 MeV electrons in water, plots $\langle z \rangle_N$ versus the inverse number of steps, $1/N$. For contrast, the default path-length correction algorithm of EGS and simulations performed without a path-length correction are shown. Recall that there is no energy loss in these simulations. As an indicator of scale, we have included a line indicating the step-size (measured in $1/N$) equal to the CSDA range in water. We have seen before that at high energies,above 3 MeV in water, the Molière upper limit exceeds the CSDA range. We have also included the ESTEPE=20% line, approximately the default EGS step-size in water. If one used the default EGS simulation, one would make path-length related errors of only a few percent. The new path-length correction would allow the default upper limit on step-size in EGS to be extended upwards, allowing steps approaching the full CSDA range, without introducing artefacts! The new path-length correction thus shows a potential of speeding up high energy simulations! Benchmarks have yet to be performed in this area.

Figure 14.19: A test of the step-size dependence of the Molière theory with the new path-length correction and with other path-length correction methods. This case is for 10 MeV electrons in water.

Figure 14.20: A test of the step-size dependence of the Molière theory with the new path-length correction and with other path-length correction methods. This case is for 10 keV electrons in water.

Figure 14.20 depicts a similar set of simulations at 10 keV, three orders of magnitude less than the previous example. The ESTEPE=20% line, near the default EGS step-size, is close to the Molière upper limit. Path-length corrections are very important here. We also show Molière's lower limit, the $\Omega_0 = 20$ line. It was mentioned previously that Molière's lower limit was found to be too conservative and that sensible results could be expected for $\Omega_0 \geq e$. This is shown in fig. 14.20. The new path-length correction (or the Molière theory) show evidence of breakdown only in the vicinity of $\Omega_0 = e$. It is more likely, however, that this is a numerical problem as various functions, which become singular near this limit, are difficult to express numerically. Similar tests have been performed for other energies and materials. In all cases the step-size independence of the path-length correction and the Molière theory is demonstrated.

14.2.4 PRESTA's lateral correlation algorithm

In section 14.1.3, we discussed the importance of lateral transport for each electron step in certain calculations. Berger's algorithm [Ber63], eq. 17.4, is used by PRESTA. To test this algorithm, we used a test very similar to that used to prove the viability of the path-length correction of the previous section. Again, we modify our electron transport algorithm to conform with all the constraints of the Molière theory. Energy loss mechanisms were “switched off”, all bounding surfaces were eliminated from the simulations to make it seem as if the transport took place in an infinite medium, and the step-size constraints of the Molière theory were obeyed. We performed the following simulations: An electron was set in motion in a given direction, which defines the z-axis for the problem. As before, a history was defined by having the total curved path, summed over all electron steps, exactly equal to the Molière upper limit. The quantity “scored” was the average displacement perpendicular to the z-axis, $\langle r \rangle_N$, at the end of the history. The sum of the curved paths of the N steps always equals t_{max}. Path-length corrections play a minor role in these simulations because the geometric straight-line transport distances are somewhat dependent upon the amount of curvature correction applied to the electron steps. However, as shown in the previous section, the path-length correction and the Molière theory are both consistent. If the lateral correlation algorithm is also consistent, then the $\langle r \rangle_N$'s should also be independent of N, or equivalently, step-size independent.

We show one representative case in fig. 14.21 for 100 keV electrons in water, which depicts $\langle r \rangle_N$ versus the inverse number of steps, $1/N$. We also show two other calculations of r_N which do not include the lateral correlation algorithm. One is the default EGS calculation with its default path-length correction and the other has no path-length correction. The relatively small difference between these two curves indicates that this test depends only weakly upon the path-length correction used. (If the new path-length correction was used without a lateral correlation algorithm, it would lie somewhere between these two curves.) A great reduction of step-size dependence in this calculation is demonstrated. Only for

Figure 14.21: Step-size independence test of the lateral correlation algorithm. Also shown are two calculations without lateral displacements, with and without the default EGS path-length correction. This test depends only weakly upon the path-length correction used. This case is for 100 keV electrons in water.

the large step-sizes is there any evidence of deviation. This feature has been observed at all energies [BR87]. However, we shall see in the next section that the remaining dependence is eliminated when energy loss is incorporated. The "ESTEPE=20%" line shows the approximate step-size used by EGS in its default configuration.

14.2.5 Accounting for energy loss

The underlying Molière theory does not treat energy loss directly. Actually, it is not too difficult to use the Molière theory in a more general fashion and incorporate energy loss. One merely has to convert integral equations in the following fashion:

$$\int_0^t dt' f(t', E(t')) \Longrightarrow \int_{E_f}^{E_0} dE' f\left(t'(E'), E'\right)/|s(E')|, \tag{14.4}$$

where $f()$ is any function of the curved path-length, t, and the energy, E. The function, s(), is the stopping power. The familiar equation relating E and t directly is obtained by making the substitution, $f() \rightarrow 1$ in the above equation. However, such equations prove to be difficult to handle numerically and it is not really necessary. In all the formulae used in regards to multiple scattering and the various elements of PRESTA, an integration over t' is involved. It is then sufficiently accurate to make the approximation that the energy is constant if it is evaluated at the midpoint of the step. In more concrete terms, we approximate,

$$\int_0^t dt' f\left(t', E(t')\right) \approx \int_0^t dt' f(t', \tilde{E}), \tag{14.5}$$

where $\tilde{E} = \frac{1}{2}[E_0 + ts(\tilde{E})]$. Note that this latter equation for $\tilde{E}$ is really an iterative equation and it has been found that it is sufficient to evaluate it only to first order. That is, we make the approximation that $\tilde{E} \approx \frac{1}{2}\{E_0 + ts(\frac{1}{2}[E_0 + ts(E_0)])\}$. Some justification for this treatment can be obtained from the following relation,

$$I = \int_{E_f}^{E_0} dE\ f(E) = \Delta E \left\{ f(\tilde{E}) + \frac{1}{24}(\Delta E)^2 f''(\tilde{E}) \ldots \right\}, \tag{14.6}$$

where $\tilde{E} = (E_0 + E_f)/2$, $\Delta E = E_0 - E_f$, and $f''(E)$ is the second derivative of with respect to E. Thus, if ΔE is not large with respect to E, and $f''()$ is not too large, the approximation, $I \approx \Delta E f(\tilde{E})$ is valid.

Further justification may be obtained by viewing the step-size independence of $\langle z \rangle_N$ and $\langle r \rangle_N$ with energy loss incorporated by the above method *i.e.* evaluating all energy-related expressions at the mid-point of the step. The results are shown in figs. 14.22 and 14.23, respectively. In each case, the step-size was chosen to be a fixed value of Molière's upper limit. However, as the particle loses energy this step-size changes owing to it's inherent energy dependence. In each case, the electron's endpoint energy, at which point the transport was

Figure 14.22: A similar test of the path-length correction done for figs. 14.19 and 14.20 but with energy loss incorporated. Electron histories were terminated when the kinetic energy, EKCUT, reached 1% of the starting energy, except in the 10 keV case where it was 10%.

Figure 14.23: A similar test of the lateral correlation done for fig. 14.21 but with energy loss incorporated. Electron histories were terminated when the kinetic energy, EKCUT, reached 1% of the starting energy, except in the 10 keV case where it was 10%.

terminated, was chosen to be 1% of the starting energy. The only exception was 10 keV, where the endpoint energy was 1 keV. We note that both $\langle z \rangle_N$ and $\langle r \rangle_N$ exhibit step-size independence. Even more remarkable is the fact that the minor step-size dependence exhibited by $\langle r \rangle_N$, shown in fig. 14.21, has vanished. This improvement appears to be fortuitous resulting form cancellations of second-order effects. More research is needed to study the theories concerning lateral displacements.

14.2.6 PRESTA's boundary crossing algorithm

The final element of PRESTA is the boundary crossing algorithm. This part of the algorithm tries to resolve two irreconcilable facts: that electron transport must take place across boundaries of arbitrary shape and orientation, and that the Molière multiple scattering theory is invalid in this context.

If computing speed did not matter, the solution would be obvious—use as small a step-size as possible within the constraints of the theory. With this method, a great majority of the transport steps would take place far removed from boundaries and the underlying theory would only be "abused" for that small minority of steps when the transport takes place in the direct vicinity of boundaries. This would also solve any problems associated with the omission of lateral translation and path-length correction. However, with the inclusion of a reliable path-length correction and lateral correlation algorithm, we have seen that we may simulate electron transport with very large steps in infinite media. For computing efficiency, we wish to use these large steps as often as possible.

Consider what happens as a particle approaches a boundary in the PRESTA algorithm. First we interrogate the geometry routines of the transport code and find out the closest distance to any boundary. As well as any other restrictions on electron step-size, we restrict the electron step-size, (total, including path-length curvature) to the closest distance to any boundary. We choose to restrict the *total* step-size so that no part of the electron path could occur across any boundaries. We then transport the particle, apply path-length corrections, the lateral correlation algorithm, and perform any "scoring" we wish to do. We then repeat the process.

At some point this process must stop, else we encounter a form of Xeno's paradox. We will never reach the boundary! We choose a minimum step-size which stops this sort of step-size truncation. We call this minimum step-size t'_{min}. If a particle's step-size is restricted to t'_{min}, we are in the vicinity of a boundary. The particle may or *may not* cross it. At this point, to avoid ambiguities, the lateral correlation algorithm is switched off, whether or not the particle actually crosses the boundary. If we eventually cross the boundary, we transport the particle with the same sort of algorithm. We *start* with a step t'_{min}. We then let the above algorithm take over. This process is illustrated in fig. 14.24. This example is for a 10 MeV electron incident normally upon a 1 cm slab of water. The first step is t'_{min} in length.

Figure 14.24: Boundary crossing algorithm example: A 10 MeV electron enters a 1 cm slab of water from the left in the normal direction. The first step is t'_{min} in length. Since the position here is less than t'_{min} away from the boundary, the next step is length t'_{min} as well. The next 4 steps are approximately $2t'_{min}$, $4t'_{min}$, $8t'_{min}$, and $16t'_{min}$ in length, respectively. Finally, the transport begins to be influenced by the other boundary, and the steps are shortened accordingly. The electron leaves the slab in 3 more steps.

Since the position at this point is less than t'_{min} away from the boundary (owing to path curvature), the next step is length t'_{min} as well. The next 4 steps are approximately $2t'_{min}$, $4t'_{min}$, $8t'_{min}$, and $16t'_{min}$ in length, respectively. Finally, the electron begins to "see" the other boundary, shortens its steps accordingly. For example, the total curved path "a" in the figure is associated with the transport step "b". The distance "a" is the distance to the closest boundary.

Finally, what choice should be made for t'_{min}? One could choose t'_{min}= t_{min}, the minimum step-size constraint of the Molière theory. Although this option is available to the PRESTA user, practice has shown it to be too conservative. Larger transport steps may be used in the vicinity of boundaries. The following choice, the default setting for t'_{min}, has been found to be be a good practical choice, allowing both accurate calculation and computing efficiency: Choose t'_{min} to equal t_{max} for the minimum energy electron in the problem (as set by transport cut-off limits). Then scale the energy-dependent parts of the equation for t'_{min} accordingly, for higher energy electrons. The reader is referred to ref. [BR87] for the mathematical details. As an example, we return to the "air tube" calculation of fig. 14.16. In that figure, the choice of "blcmin", the variable in PRESTA which controls the boundary crossing algorithm and which is closely related to t'_{min}, was set to 1.989. This causes t'_{min} to be equal to t_{max} for 2 keV electrons. A transport cut-off of 2 keV is appropriate in this simulation because electrons with this energy have a range which is a fraction of the diameter of the tube. In most practical problems, if one chooses the transport cut-off realistically, PRESTA's default selection for t'_{min} produces accurate results. Again, the reader is referred to the PRESTA documentation [BR87] for further discussion.

PRESTA, as the name implies, was designed to calculate quickly as well as accurately, since it wastes little time taking small transport steps in regions where it has no need to. There is no space to go into further discussion about this although there is a brief discussion in Chapter 17. Again, the reader is referred elsewhere [BR87]. Typical timing studies have shown that PRESTA, in its standard configuration, executes as quickly, and sometimes much more quickly, then EGS with ESTEPE set so as to produce accurate results. For problems with a fine mesh of boundaries, for example a depth-dose curve with a $r_0/40$ mesh, the timing is about the same. For other problems, with few boundaries, the gain in speed is about a factor of 5.

14.2.7 Caveat Emptor

It would leave the reader with a mistaken impression if the chapter was terminated at this point. PRESTA has demonstrated that step-size dependence of calculated results has been eliminated in many cases and that computing time can be economized as well. By understanding the elements of condensed-history electron transport, some problems have been solved. Calculational techniques that isolate the effects of various constituents of the electron transport algorithm have been developed and used to prove their step-size independence. However, PRESTA is not the final answer because it does not solve all step-size dependence problems, in particular, backscattering. This is demonstrated by the example shown in fig. 14.25. In this example, 1.0 MeV electrons were incident normally on a semi-infinite slab of water. The electron transport was performed in the CSDA approximation. That is, no δ-rays or bremsstrahlung γ's were set in motion and the unrestricted collision stopping power was used. The ratio of backscattered kinetic energy to incident kinetic energy was calculated. The default EGS calculation (with ESTEPE control) is shown to have a large step-size dependence. The PRESTA calculation is much improved but still exhibits some residual dependence on step-size.

In general, problems that depend strongly on backscatter will exhibit a step-size dependence, although the severity is much reduced when one uses PRESTA. We may speculate on the reason for the existence of the remaining step-size dependence. Recall that the path-length correction, which relates the straight-line path length, s, and t, the curved path-length of the transport step, really calculates only an average value. That is, given t, the value of s is predetermined and unique. It is really a distributed quantity and should be correlated to the multiple scattering angle of the step. In other words, we expect the distribution to be peaked in the backward direction if $\Theta = \pi$ and peaked in the forward direction if $\Theta = 0$. To this date, distributions of this sort which are accurate for large angle scattering are unknown. If they are discovered they may cure PRESTA's remaining step-size dependence.

Figure 14.25: Fractional energy backscattered from a semi-infinite slab of water with 1.0 MeV electrons incident normally. The electron transport was performed in the CSDA approximation. (No δ-rays or γ's were set in motion). The default EGS and PRESTA calculations are contrasted. There is still evidence of step-size dependence in the PRESTA calculation.

Bibliography

[Ber63] M. J. Berger. Monte Carlo Calculation of the penetration and diffusion of fast charged particles. *Methods in Comput. Phys.*, 1:135 – 215, 1963.

[Bet53] H. A. Bethe. Molière's theory of multiple scattering. *Phys. Rev.*, 89:1256 – 1266, 1953.

[BR87] A. F. Bielajew and D. W. O. Rogers. PRESTA: The Parameter Reduced Electron-Step Transport Algorithm for electron Monte Carlo transport. *Nuclear Instruments and Methods*, B18:165 – 181, 1987.

[BRN85] A. F. Bielajew, D. W. O. Rogers, and A. E. Nahum. Monte Carlo simulation of ion chamber response to ^{60}Co – Resolution of anomalies associated with interfaces,. *Phys. Med. Biol.*, 30:419 – 428, 1985.

[BS83] M. J. Berger and S. M. Seltzer. Stopping power and ranges of electrons and positrons. *NBS Report NBSIR 82-2550-A (second edition)*, 1983.

[Eyg48] L. Eyges. Multiple scattering with energy loss. *Phys. Rev.*, 74:1534, 1948.

[Fan54] U. Fano. Note on the Bragg-Gray cavity principle for measuring energy dissipation. *Radiat. Res.*, 1:237 – 240, 1954.

[GS40a] S. A. Goudsmit and J. L. Saunderson. Multiple scattering of electrons. *Phys. Rev.*, 57:24 – 29, 1940.

[GS40b] S. A. Goudsmit and J. L. Saunderson. Multiple scattering of electrons. II. *Phys. Rev.*, 58:36 – 42, 1940.

[HW55] D. F. Hebbard and R. P. Wilson. The effect of multiple scattering on electron energy loss distribution. *Aust. J. Phys.*, 8:90 –, 1955.

[Lew50] H. W. Lewis. Multiple scattering in an infinite medium. *Phys. Rev.*, 78:526 – 529, 1950.

[Mol47] G. Z. Molière. Theorie der Streuung schneller geladener Teilchen. I. Einzelstreuung am abgeschirmten Coulomb-Field. *Z. Naturforsch*, 2a:133 – 145, 1947.

[Mol48] G. Z. Molière. Theorie der Streuung schneller geladener Teilchen. II. Mehrfach- und Vielfachstreuung. *Z. Naturforsch*, 3a:78 – 97, 1948.

[NHR85] W. R. Nelson, H. Hirayama, and D. W. O. Rogers. The EGS4 Code System. Report SLAC–265, Stanford Linear Accelerator Center, Stanford, Calif, 1985.

[RBN85] D. W. O. Rogers, A. F. Bielajew, and A. E. Nahum. Ion chamber response and $\mathrm{A_{wall}}$ correction factors in a ^{60}Co beam by Monte Carlo simulation. *Phys. Med. Biol.*, 30:429 – 443, 1985.

[Rog84] D. W. O. Rogers. Low energy electron transport with EGS. *Nucl. Inst. Meth.*, 227:535 – 548, 1984.

[SA55] L. V. Spencer and F. H. Attix. A theory of cavity ionization. *Radiat. Res.*, 3:239 – 254, 1955.

[Yan51] C. N. Yang. Actual path length of electrons in foils. *Phys. Rev*, 84:599 – 600, 1951.

Problems

1.

Chapter 15

Advanced electron transport algorithms

In this chapter we consider the transport of electrons in a condensed history Class II scheme [Ber63]. That is to say, the bremsstrahlung processes that result in the creation of photons above an energy threshold E_γ, and Møller knock-on electrons set in motion above an energy threshold E_δ, are treated discretely by creation and transport. Sub-threshold processes are accounted for in a continuous slowing down approximation (CSDA) model. For further description of the Class II scheme the reader is encouraged to read Berger's article [Ber63] who coined the terminology and gave a full description and motivation for the classification scheme. Figure 15.1 gives a graphical description of the transport.

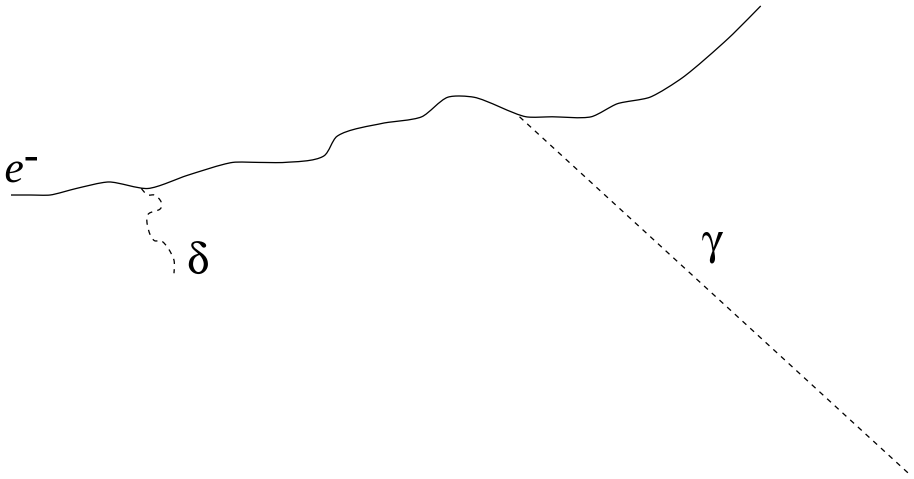

Figure 15.1: This is a depiction of a complete electron history showing elastic scattering, creation of bremsstrahlung above the E_γ threshold, the setting in motion of a knock-on electron above the E_δ threshold and absorption of the primary and knock-on electrons.

The electron transport processes between the particle creation, absorption vertices is governed by the Boltzmann transport equation as formulated by Larsen [Lar92]:

$$\left[\frac{1}{v}\frac{\partial}{\partial t} + \vec{\Omega}\cdot\vec{\nabla} + \sigma_{\rm s}(E) - \frac{\partial}{\partial E}L(E)\right]\psi(\vec{x},\vec{\Omega},E,t) = \int_{4\pi} {\rm d}\Omega'\,\sigma_{\rm s}(\vec{\Omega}\cdot\vec{\Omega}',E)\psi(\vec{x},\vec{\Omega}',E,t)\ , \quad (15.1)$$

where $\vec{x}$ is the position, $\vec{\Omega}$ is a unit vector indicating the direction of the electron, E is the energy of the electron and t is time. $\sigma_{\rm s}(\vec{\Omega}\cdot\vec{\Omega}',E)$ is the macroscopic differential scattering cross section,

$$\sigma_{\rm s}(E) = \int_{4\pi} {\rm d}\Omega'\,\sigma_{\rm s}(\vec{\Omega}\cdot\vec{\Omega}',E) \quad (15.2)$$

is the total macroscopic cross section (probability per unit length), $L(E)$ is the restricted stopping power appropriate for bremsstrahlung photon creation and Møller electrons beneath their respective thresholds E_γ and E_δ, v is the electron speed and $\psi(\vec{x},\vec{\Omega},E,t)\,{\rm d}\vec{x}\,{\rm d}\Omega\,{\rm d}E$ is the probability of there being an electron in ${\rm d}\vec{x}$ about $\vec{x}$, in ${\rm d}\Omega$ about $\vec{\Omega}$ and in ${\rm d}E$ about E at time t. The boundary condition to be applied to each segment in Figure 15.1 is:

$$\psi(\vec{x},\vec{\Omega},E,0) = \delta(\vec{x})\delta(\hat{z}-\vec{\Omega})\delta(E_n - E)\ , \quad (15.3)$$

where the start of each segment is translated to the origin and rotated to point in the z-direction. ($\hat{z}$ is a unit vector pointing along the z-axis.) The energy at the start of the n-th segment is E_n.

For our considerations within the CSDA model, we note that E and t can be related since the pathlength, s,

$$s = vt = \int_E^{E_n} \frac{{\rm d}E'}{L(E')}, \quad (15.4)$$

permitting a slight simplification of Eq. 15.1:

$$\left[\frac{\partial}{\partial s} + \vec{\Omega}\cdot\vec{\nabla} + \sigma_{\rm s}(E)\right]\psi(\vec{x},\vec{\Omega},s) = \int_{4\pi} {\rm d}\Omega'\,\sigma_{\rm s}(\vec{\Omega}\cdot\vec{\Omega}',E)\psi(\vec{x},\vec{\Omega}',s)\ . \quad (15.5)$$

The cross section still depends on E which may be calculated from Eq. 15.4.

Lewis [Lew50] has presented a "formal" solution to Eq.15.5. By assuming that ψ can be written in an expansion in spherical harmonics,

$$\psi(\vec{x},\vec{\Omega},s) = \sum_{lm} \psi_{lm}(\vec{x},s)Y_{lm}(\vec{\Omega})\ , \quad (15.6)$$

one finds that

$$\left[\frac{\partial}{\partial s} + \kappa_l\right]\psi_{lm}(\vec{x},s) = -\sum_{\lambda\mu}\vec{\nabla}\psi_{\lambda\mu}(\vec{x},s)\cdot\vec{Q}_{lm}^{\lambda\mu}\ , \quad (15.7)$$

where

$$\kappa_l(E) = \int_{4\pi} {\rm d}\Omega'\,\sigma_{\rm s}(\vec{\Omega}\cdot\vec{\Omega}',E)[1 - P_l(\vec{\Omega}\cdot\vec{\Omega}')]\ , \quad (15.8)$$

and

$$\vec{Q}^{\lambda\mu}_{lm} = \int_{4\pi} \mathrm{d}\Omega\, Y^*_{lm}(\vec{\Omega})\, \vec{\Omega}\, Y_{\lambda\mu}(\vec{\Omega}) \ . \tag{15.9}$$

If one considers angular distribution only, then one may integrate over all $\vec{x}$ in Eq. 15.7 giving:

$$\left[\frac{\partial}{\partial s} + \kappa_l\right] \psi_l(s) = 0 \ , \tag{15.10}$$

resulting in the solution derived by Goudsmit and Saunderson [GS40a, GS40b]:

$$\psi(\vec{\Omega}, s) = \frac{1}{4\pi} \sum_l (2l+1) P_l(\hat{z}\cdot\vec{\Omega}) \exp\left(-\int_0^s \mathrm{d}s'\, \kappa_l(E)\right) \ . \tag{15.11}$$

Eq. 15.7 represents a complete formal solution of the Class II CSDA electron transport problem but it has never been solved exactly. However, Eq. 15.7 may be employed to extract important information regarding the moments of the distributions. Employing the definition,

$$k_l(s) = \exp\left(-\int_0^s \mathrm{d}s'\, \kappa_l(E)\right) \ , \tag{15.12}$$

Lewis [Lew50] has shown the moments $\langle z\rangle$, $\langle z\cos\Theta\rangle$, and $\langle x^2+y^2\rangle$ to be:

$$\langle z\rangle = \int_0^s \mathrm{d}s'\, k_1(s') \ , \tag{15.13}$$

$$\langle z\cos\Theta\rangle = \frac{k_1(s)}{3} \int_0^s \mathrm{d}s'\, \frac{1+2k_2(s')}{k_1(s')} \ , \tag{15.14}$$

and

$$\langle x^2+y^2\rangle = \frac{4}{3} \int_0^s \mathrm{d}s'\, k_1(s') \int_0^{s'} \mathrm{d}s''\, \frac{1-k_2(s'')}{k_1(s'')} \ . \tag{15.15}$$

It can also be shown using Lewis's methods that

$$\langle z^2\rangle = \frac{2}{3} \int_0^s \mathrm{d}s'\, k_1(s') \int_0^{s'} \mathrm{d}s''\, \frac{1+2k_2(s'')}{k_1(s'')} \ , \tag{15.16}$$

and

$$\langle x^2+y^2+z^2\rangle = 2 \int_0^s \mathrm{d}s'\, k_1(s') \int_0^{s'} \mathrm{d}s''\, \frac{1}{k_1(s'')} \ , \tag{15.17}$$

which gives the radial coordinate after the total transport distance, s. Note that there was an error[1] in Lewis's paper where the factor 1/3 was missing from his version of $\langle z\cos\Theta\rangle$. In the

[1] The correction of Lewis's Eq. 26 is:

$$H_{l1} = \sqrt{\frac{1}{4\pi(2l+1)}} k_l(s) \int_0^s \mathrm{d}s'\, \frac{l k_{l-1}(s') + (l+1) k_{l+1}(s')}{k_1(s')}$$

The reader should consult Lewis's paper [Lew50] for the definition of the H-functions.

limit that $s \longrightarrow 0$, one recovers from Eqs. 15.14 and 15.16 the results $\lim_{s \longrightarrow 0} \langle z \cos\Theta \rangle = s$ and $\lim_{s \longrightarrow 0} \langle z^2 \rangle = s^2$ which are not obtained without correcting the error as described in the footnote. Similar results for the moments have been derived recently by Kawrakow [Kaw96a] using a statistical approach.

Before leaving this introductory section it warrants repeating that these equations are all "exact" within the CSDA model and are independent of the form of the elastic scattering cross section. It should also be emphasized that Larsen analysis [Lar92] proves that the condensed history always gets the correct answer (consistent with the validity of the elastic scattering cross section) in the limit of small step-size providing that the "exact" Goudsmit-Saunderson multiple-scattering formalism is employed (and that its numerical stability problems at small step-size can be solved). Larsen analysis also draws some conclusions about the underlying Monte Carlo transport mechanisms and how they relate to convergence of results to the correct answer. Some Monte Carlo techniques can be expected to be less step-size dependent than others and converge to the correct answer more efficiently, using larger steps.

The ultimate goal of a Monte Carlo transport algorithm should be to make electron condensed history calculations as stable as possible with respect to step-size. That is, for a broad range of applications there should be step-size *independence* of the result. Hence, it would be most efficient to use steps as large as possible and not be subject to calculation errors. While we have not yet achieved this goal, we have made much progress towards it and describe some of this progress in a later section.

15.1 What does condensed history Monte Carlo do?

Monte Carlo calculations attempt to solve Eq. 15.5 iteratively by breaking up the transport between discrete interaction vertices, as depicted in Figure 15.2. The first factor determining the electron step-size distance is the distance to a discrete interaction. These distances are stochastic and characterized by an exponential distribution. Further subdivision schemes may be employed and these can be classified as numeric, physics' or boundary step-size constraints.

15.1.1 Numerics' step-size constraints

A geometric restriction, say $s \leq s_{\max}$ may be used.A geometric restriction of this form was introduced by Rogers [Rog84b] in the EGS Monte Carlo code [NHR85, BHNR94]. This has application in graphical displays of Monte Carlo histories. One wants the electron tracks to have smooth lines and so the individual pathlengths should be of the order of the resolution size of the graphics display. Otherwise, the tracks look artificially jagged, as they do in Figure 15.2. Of course, there are some real sharp bends in the electron tracks associated with large angle elastic scattering, but these are usually infrequent. One can predict the

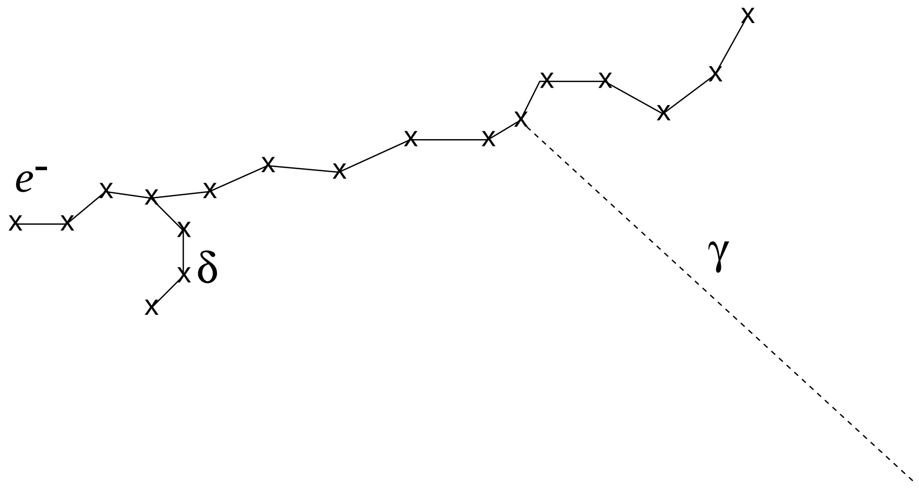

Figure 15.2: This is how a Monte Carlo calculates of the complete electron history as depicted in Figure 15.1. The transport takes place in steps, the vertices of which are marked with the symbol "×".

number of steps required to follow a particle to termination. For this case $N = r(E_{\rm i})/s_{\rm max}$, where $r(E_{\rm i})$ is the range of an electron with starting energy $E_{\rm i}$.

Another popular choice is a constant fractional energy loss per electron step, *i.e.* $\Delta E/E =$ constant. This has a slight disadvantage that the electron steps get shorter and shorter as the energy of the electrons in the simulation gets smaller and smaller. In terms of the dynamic range of the energies of the particles in the simulation, generally the lower ones play a lesser important role (there are exceptions to this of course!) and so, despite its popularity, it is probably wasteful in many applications. One can predict the number of steps required to follow a particle to termination in this case as well. For this case $N = log(1 - \Delta E/E)/log(E\text{min}/E_{\rm i})$ where Emin is the minimum electron energy for which transport takes place. One sees that as Emin is pushed downwards by the requirements of some applications, that the number of steps acquires a slow logarithmic growth, unlike the geometric restriction. The constant fractional energy loss per electron step is built into the ETRAN Monte Carlo code [Ber63, Sel89, Sel91] and its well-known progeny, the ITS system [HKM$^+$92] and the electron component of MCNP [Bri86, Bri93]. In these code systems, the value of the constant is kept in internal tables and its value determined through trial and error. In the EGS system [NHR85, BHNR94], it is available to the user as a "tuning" parameter, to be adjusted (lowered) until the answer converges to the (presumably correct) result.

There are other schemes of step-size restriction that will not be discussed. However, we see that the two discussed thus far play the role of an "integration mesh-density". To get better results one must increase the resolution. To be practical, the mesh density ought to be as large as possible, consistent with target accuracy of the application. Larsen [Lar92] has made an interesting analysis of Monte Carlo algorithms and how they should be expected to converge. He showed that the ETRAN scheme proposed by Berger [Ber63] contains O(Δs) errors. Thus, one expects that accurate calculation with ETRAN methods would converge slowly and require small step-sizes to get the answer correct. Indeed, this was the case with the EGS code as well, motivating the step-size restrictions introduced into EGS by Rogers [Rog84b].

Larsen [Lar92] also proposed an alternative Monte Carlo method (calling it "Method 3") to enable faster convergence since it contains O(Δs^2) errors. The algorithm for each sub-step, Δs, is that first it must be broken into two parts. The first part is a drift of length $\Delta s/2$ in the initial direction of motion, and a deduction in energy due to continuous energy losses over the first part of this step. The multiple-scattering angle is sampled at this new energy but for a deflection angle assuming that the particle as gone the full sub-step distance, Δs, and deflection by this angle. The sub-step is then completed by executing a drift of distance $\Delta s/2$ in the new direction. Although this method may seem as is if is doubling the number of steps, this is actually not the case since the most computer-intensive part of the process, namely sampling the multiple-scattering angle and rotation, is performed only once per sub-step.

The result of the "Method 3" procedure is to impart longitudinal and lateral distributions to the sub-step, both correlated to the multiple-scattering angle, Θ. Assuming the particle starts at the origin and is directed along the z-axis, after a total sub-step pathlength of Δs, the final resting place will be:

$$\begin{aligned}
\Delta x &= (\Delta s/2)\sin\Theta\cos\Phi \\
\Delta y &= (\Delta s/2)\sin\Theta\sin\Phi \\
\Delta z &= (\Delta s/2)(1+\cos\Theta)\ , \qquad (15.18)
\end{aligned}$$

where Φ is a randomly selected azimuthal angle and it is understood that Θ is sampled from the Goudsmit-Saunderson [GS40a, GS40b] multiple-scattering theory at the mid-point energy.

Only two previously published Monte Carlo methods have followed this prescription. Berger's method [Ber63] is similar except that he proposed a straggling term for the lateral components and the energy dependence was taken account for directly. However, the longitudinal and lateral distributions have only been recently been implemented into ETRAN [Sel91].

The other method is the PRESTA algorithm [BR86, BR87] that has been incorporated into EGS. This algorithm is different from Method-3 in that the final longitudinal position was determined by its average rather than the distribution implied by Eq. 15.18. However, this only contributes to the O(Δs^2) error. The other very important distinction is that the PRESTA algorithm employs the Molière multiple-scattering method method [Mol47, Mol48] with corrections and limitations discussed by Bethe [Bet53].

There has also been a recently-published method called the Longitudinal and Lateral Correlation Algorithm (LLCA) proposed by Kawrakow [Kaw96a]. It incorporates the Method-3 transport scheme except that the multiple-scattering theory, while representing an improvement over Molière's method is still an approximation [Kaw96b] to the Goudsmit-Saunderson method. However, there is one important improvement over Method-3 in that the lateral position of the electron at the end of the sub-step is connected to the multiple-scattering angle by means of a distribution, rather than direct correlation as implied by Eq. 15.18. For the present, this distribution function has only been calculated using single-scattering methods (analog Monte Carlo or event-by-event elastic scattering). Presumably, this is an $O(\Delta s^2)$ or higher correction.

15.1.2 Physics' step-size constraints

There are also step-size restrictions related to keeping the step-sizes within range of the validity of the theories underlying the condensed history method. This is important, for example, when using the Molière multiple-scattering theory [Mol47, Mol48]. Bethe [Bet53] analyzed Molière multiple-scattering theory, comparing it to the "exact" theory of Goudsmit and Saunderson [GS40a, GS40b] and provided a correction that improves the large-angle behaviour of Molière theory for large angles as Molière theory is couched in the small-angle formalism of Bothe [Bot21] and Wentzel [Wen22]. The electron step-size constraint arises from not allowing the multiple-scattering angle to attain values greater than 1 radian.

Small-angle multiple scattering theories still play an important role in electron Monte Carlo calculations of the Class II variety since Class II condensed history techniques sample the multiple-scattering distributions "on-the-fly" as the pathlength can, in principle be anything within the constraints already discussed. Class I algorithms, as defined in Berger's work [Ber63], demand that the electrons follow a predetermined energy grid, allowing the multiple-scattering distributions to be pre-calculated. While Class I and Class II have their attributes and shortcomings, the use of an approximate multiple-scattering theory in Class II calculations, considered with the conclusion of the previous section, forces the realization that one can not necessarily expect that the limit of small step-size will produce the correct answer for Class-II/approximate multiple-scattering algorithms! It should also be remarked that Class-I/exact multiple-scattering schemes are subject to numerical instabilities as smaller step-sizes require an increasing number of terms in the Legendre serious of Eq. 15.11 to be summed. There have been studies demonstrating that one can converge to the incorrect answer in a Class-II/approximate multiple-scattering algorithm [Rog93, Bie96].

Step-size instability of the Molière theory [Mol48] has been studied extensively [AMB93, Bie94] and comparisons with Goudsmit-Saunderson theory [GS40a, GS40b] have been performed [Bet53, Win87] as well as comparisons with single-elastic scattering Monte Carlo [Bie94]. This has motivated the development of a new multiple-scattering theory based on Goudsmit-Saunderson theory [GS40a, GS40b] but formulated in such a way as to allow sampling "on-the-fly" as required by Class II algorithms and eliminating the small step-size numerical instability of the Goudsmit-Saunderson Legendre summation that arises from the form expressed in Eq. 15.11. This recent work [KB98] will be discussed in a later section. However, this new multiple-scattering theory will guarantee that condensed history Monte Carlo will always converge to the correct answer in the limit of small electron step-size.

15.1.3 Boundary step-size constraints

The final category of electron step-size constraint we consider relates to the geometry, specifically interfaces and material boundaries. Although the transport theory expressed in Eq. 15.5 and the various solutions to it describe electron transport in infinite, unbounded uniform media, practical applications contain boundaries and interfaces between media. Except for the stopping of electrons at interfaces and the updating the material-dependent transport data, this problem was not considered until the EGS/PRESTA algorithm was developed [BR86, BR87]. This algorithm requires knowledge of the nearest distance to any interface and shortens the electron step-size accordingly, by setting $s = s^{\perp}_{\rm min}$ where $s^{\perp}_{\rm min}$ is the nearest distance to any interface. This is always a perpendicular distance (as suggested by the notation) unless the closest distance happens to be along an intersection of two surfaces. This procedure requires more information from the geometry[2] but it is necessary to avoid potential misuse of the underlying transport theory. Of course this shortening can not continue indefinitely as the electron would never reach the surface, a transport equivalent of Xeno's paradox. PRESTA continues the procedure until the transport steps approach the lower limit of validity of Molière theory, usually from about 3 to 20 mean-free-path distances, and then allows the electron to reach the surface, does not model the lateral components of sub-step transport given in Eq. 15.18 (This is a necessary part of the transport logic, otherwise the lateral transport takes the electron away from the surface, in either medium), updates material-dependent data and carries on in the next medium. The initial distance is again related to the lower limit of validity of Molière theory and thereafter the algorithm adjusts step-sizes according to $s^{\perp}_{\rm min}$. As a particle recedes from a boundary, its steps grow and grow, allowing for efficient, rapid transport away from interfaced. This behaviour is depicted in Figure 15.3.

However, this technique is not without its difficulties. Because lateral transport is not modeled for the steps that touch the boundary, the multiple-scattering deflection is performed at the end of the sub-step. Electron can thus backscatter from a surface, requiring careful handling of the transport logic in the vicinity of interfaces [Bie95]. This "boundary-crossing algorithm" as implemented in PRESTA also pushes the Molière theory towards the edge of its region of validity. Granted, the misuse of Molière theory is minimized but it still exists.

[2] The general requirements for electron transport in a geometry composed entirely of planar and quadric surfaces (*i.e.* spheroids, cones, hyperboloids, paraboloids) has recently been developed [Bie95]. Although the distance of intersection to any quadric surface along the particle trajectory requires finding the root of a quadratic equation, the nearest distance to a quadric surface is the root of an $n^{\rm th}$-order equation where $n \leq 6$!

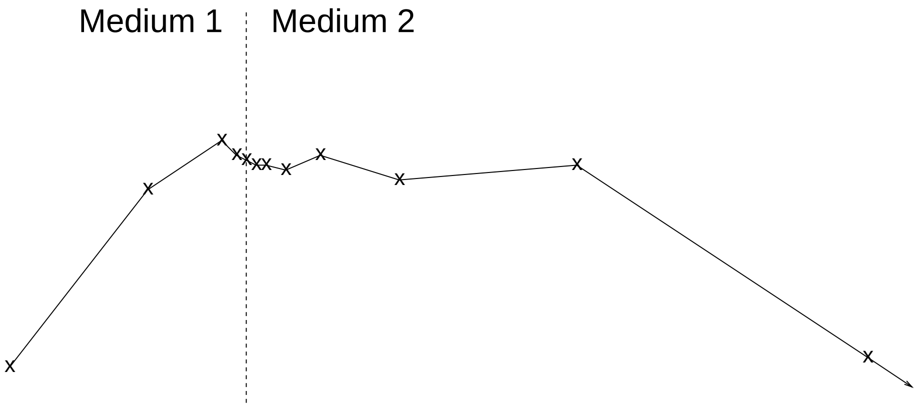

Figure 15.3: This is a depiction of operation of the PRESTA algorithm, which adjusts electron step-sizes in the vicinity of boundaries.

A more fundamental shortcoming was pointed out by Foote and Smyth [FS95] who pointed out that the deflection at the interface can cause a spurious events whereby an electron, having crossed a boundary can assume a trajectory that is parallel, or nearly so, to the surface at this point. The artefact shows up interfaces between condensed materials and gases. An electron penetrating the gas may be scattered into a near-parallel trajectory with the boundary. Even step-sizes of the order of several mean-free-path distances may be too large in the gas.

This artefact can be eliminated through use of a condensed history method that "evaporates" to a single-scattering method in the vicinity of interfaces [Bie96]. The algorithm is sketched in Figure 15.4. Using the new method, the only way that an electron can cross the interface is through a "no scatter drift" across the interface which involves no approximation. This technique, coupled with the new multiple-scattering theory will allow for error-free Monte Carlo calculations in the limit of small step-size in applications with arbitrarily complex geometries, interfaces and media.

15.2 The new multiple-scattering theory

The "exact" multiple-scattering angular distribution of Eq. 15.11 may be integrated easily over azimuthal angles (assuming that the cross section does not depend on polarisation) and written:

$$\psi(\cos\Theta, s) = \sum_l (l+1/2) P_l(\cos\Theta) \exp\left(-\int_0^s \mathrm{d}s'\, \kappa_l(E)\right) , \tag{15.19}$$

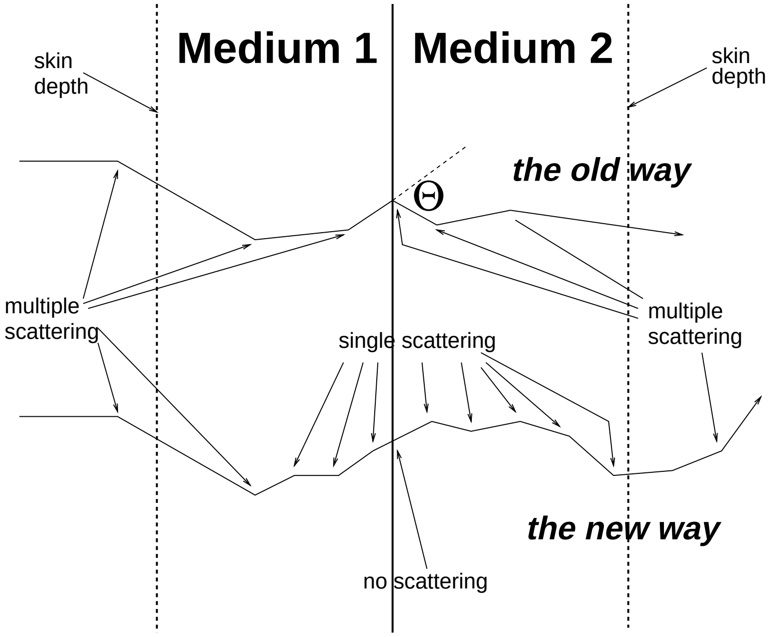

Figure 15.4: This is a depiction of the new boundary-cross algorithm which eliminates boundary-related artefacts.

and then reorganized in the following form [KB98]:

$$\psi(\cos\Theta, s) = e^{-\lambda}\delta(1-\cos\Theta)+\lambda e^{-\lambda}\tilde{\sigma}+(1-e^{-\lambda}-\lambda e^{-\lambda})\sum_l (l+1/2)P_l(\cos\Theta)\frac{e^{-\lambda g_l}-1-\lambda g_l}{e^{-\lambda}-1-\lambda} \ , \tag{15.20}$$

where

$$\lambda = \int_0^s \mathrm{d}s'\,\sigma_{\mathrm{s}}(E) \tag{15.21}$$

is the distance measured in mean-free-path taking into account the change in energy of the scattering cross section, and $e^{-\lambda}$ is the probability that the electron can go a distance λ without scattering even once,

$$\tilde{\sigma} = \frac{1}{\lambda}\int_0^s \mathrm{d}s'\,\sigma_{\mathrm{s}}(\cos\Theta, E) \tag{15.22}$$

is the angular distribution of a single-scattering event with probability $\lambda e^{-\lambda}$ taking into account energy loss, and

$$g_l = \frac{1}{\lambda}\int_0^s \mathrm{d}s' \int_0^\pi \mathrm{d}(\cos\Theta)\,\sigma_{\mathrm{s}}(\cos\Theta, E)P_l(\cos\Theta) \ , \tag{15.23}$$

which is related to the κ_l defined in Eq. 15.8.

The general from of Eq. 15.21 was suggested by Berger and Wang [BW89] as a way of reducing some of the singularity in Eq. 15.11 to make the summation for large-l tractable.

This approach has some moderate success for Class I pre-calculations but Class II algorithms must still sample "on-the-fly". Therefore, we have adopted an alternative approach.

This approach is based on a similar analysis of the small-angle multiple-scattering problem [Bie94]. Consider the part of Eq. 15.20 that describes two or more scatterings. Defining the notation

$$\psi^{(2+)}(\mu, s) = \sum_l (l + 1/2) P_l(\mu) \frac{e^{-\lambda g_l} - 1 - \lambda g_l}{e^{-\lambda} - 1 - \lambda} \ , \tag{15.24}$$

where $\mu = \cos\Theta$. The change of variables,

$$u = (1 + a)\left(1 - \frac{2a}{1 - \mu + 2a}\right) \tag{15.25}$$

allows us to write an alternate form of $\psi^{(2+)}(\mu, s)$, namely

$$q^{(2+)}(u, s)\mathrm{d}u = \psi^{(2+)}(\mu, s)\mathrm{d}\mu \ , \tag{15.26}$$

where for the moment, a is an arbitrary parameter.

The motivation for this transformation is quite subtle. The magnitude of the derivative of u with respect to μ is:

$$\mathrm{d}u = (1 + a)\left(1 - \frac{2a}{1 - \mu + 2a}\right) \ , \tag{15.27}$$

which resembles a screened Rutherford cross section with an arbitrary screening angle, a. As discovered in the small-angle study, most of the shape of the multiple-scattering distribution, which is peaked strongly in the forward direction for the usual case of small screening angles, resembles a screened Rutherford cross section with some effective width. The "effective screening" angle a can then be fixed by th requirement that $q^{(2+)}(u, s)$ be as flat as possible for all angles and all transport distances. The procedure is described elsewhere. It suffices to say that the $q^{(2+)}$-surfaces produced, starting with a screened Rutherford cross section employing the Molière screening angle [Mol47] along with Mott [Mot29, Mot32] that includes spin and relativistic corrections [Mot29, Mot32], are flat enough so that a linear interpolation table that is accurate to within 0.2% can be represented in a few hundred kB of data for 100 atomic elements suitable for applications from 1 keV upwards[3].

15.3 Longitudinal and lateral distributions

In this section we consider longitudinal and lateral transport components of Monte Carlo sub-step. Although the transport scheme represented by Eq. 15.18 has been shown to yield results correct to O(Δs), it can be shown that all the moments represented by Eqs. 15.13–15.17 are not correct. Thus, even average penetration distances and lateral diffusion are not

[3] We are grateful to Dr Stephen Seltzer for providing the Mott cross section data.

accounted for correctly. For most applications, electrons scatter for many elastic and inelastic scatterings before tallying some result. After many scatterings, the only information that really matters are the first few moments. There are, of course exceptions, the most important one being low energy electron backscatter. In the application, single and plural events can lead to backscatter from a foil. The "boundary-crossing algorithm" discussed previously may come to the rescue, however. This is because electrons must penetrate using single-scattering methods to a skin-depth of several mean-free-path distances before the condensed history algorithm is allowed to take over. If single and plural scattering from within the skin-depth is contributing in a significant way to the backscatter events, then this will automatically be accounted for.

We now describe a transport algorithm that gives exactly the Lewis moments, $\langle z\rangle$, $\langle z\cos\Theta\rangle$, $\langle z^2\rangle$, and $\langle x^2+y^2\rangle$ with only a little more computational effort[4].

We create the ansatz:

$$\begin{aligned}
\Delta x/s &= [\beta(s)-\delta_{\|}(s,\xi_x)]\sin\Theta\cos[\Phi-\delta_\phi(s,\xi_\phi)]+\delta_\perp(s,\xi_x) \\
\Delta y/s &= [\beta(s)-\delta_{\|}(s,\xi_x)]\sin\Theta\sin[\Phi-\delta_\phi(s,\xi_\phi)]+\delta_\perp(s,\xi_y) \\
\Delta z/s &= [\beta(s)-\delta_{\|}(s,\xi_z)]\cos\Theta+[\alpha(s)-\delta_{\|}(s,\xi_x)]\ ,
\end{aligned} \tag{15.28}$$

where $\delta_\perp$, $\delta_\|$ and δ_ϕ are transverse, longitudinal and azimuthal straggling functions and ξ_i is a uniform random variable between 0 and 1. The lateral and longitudinal straggling functions have the property that their average is exactly zero, *i.e.* $\int_0^1 \mathrm{d}\xi\,\delta_\perp(s,\xi)=0$ and $\int_0^1 \mathrm{d}\xi\,\delta_\|(s,\xi)=0$ while the azimuthal straggling function's average value represents the average angle between the direction of motion and the azimuthal component of the straggling function [Kaw96a]. It also has a straggling component. This function has been determined by single-scattering calculations [Kaw96a]. The functions $\alpha(s)$ and $\beta(s)$ can be found by insisting that $\langle z\rangle$ and $\langle z\cos\Theta\rangle$ comply with $\langle z\rangle$ in Eq. 15.13 and with $\langle z\cos\Theta\rangle$ in Eq. 15.14. The values of $\int_0^1 \mathrm{d}\xi\,\delta_\perp^2(s,\xi)$ and $\int_0^1 \mathrm{d}\xi\,\delta_\|^2(s,\xi)$ can be determined by forcing agreement with $\langle z^2\rangle$, and $\langle x^2+y^2\rangle$. It should be remarked that the shape of the straggling functions is not determined by this approach. We can derive more information about them by calculating higher Lewis moments. This will lead to information about $\int_0^1 \mathrm{d}\xi\,\delta_\perp^n(s,\xi)$ and $\int_0^1 \mathrm{d}\xi\,\delta_\|^n(s,\xi)$, where $n>2$. Since we do not know the exact shape of the straggling functions, we have to guess. Small-angle theory suggests Gaussian's for the lateral straggling functions. This work remains to be done. The use of the previously-computed azimuthal straggling function, should guarantee compliance with $\langle x\sin\Theta\cos\Phi\rangle$ and $\langle y\sin\Theta\sin\Phi\rangle$.

Before ending this section, we make a few remarks on the computational efficiency of this new method. The most computationally intensive part of Eq. 15.28 is sampling the multiple-scattering distribution, which is required by any method. The straggling functions can be

[4] It is very important remark that more advanced methods have to be computationally efficient. It is pointless to develop complicated calculational schemes that cost more to execute than simply turning down the step-size to obtain the same degree of accuracy!

pre-calculated and put into interpolation tables, a task no more difficult than the multiple-scattering table described in the previous section. Alternatively, since the shape of the distributions is arbitrary, simple forms may be used and sampling these distributions may be very rapid.

One possible criticism of this approach is that it is bound to produce the occasional unphysical result of the form $x^2 + y^2 + z^s > s^2$. It is anticipated that this type of event will be rare, and some indication of this has been given by Berger [Ber63], who first suggest Gaussian straggling terms for the lateral component. Another criticism is that the ansatz in Eq. 15.28 is not general enough. Indeed, although moments of the type $\langle z^n \rangle$ can be used to determine $\int_0^1 \mathrm{d}\xi \, \delta_{\parallel}^n(s, \xi)$, they will likely be in conflict with higher order moments of the sort $\langle z^n \cos^m \Theta \rangle$. Actually, this criticism is coupled directly to the previous one and results from our incomplete understanding of the solution to the complete transport problem. Further research along these lines, such as the Fokker-Planck solution to this problem, while approximate will shed more insight on the general transport solution. However, it is also likely that Eq. 15.28 represents a significant advance in condensed history methods and may provide true step-size independence for a large class of electron transport problems.

15.4 The future of condensed history algorithms

We conclude with some comments on the future of condensed history algorithms to place its research in some sort of larger perspective. We investigate briefly two scenarios that are pointed to by present computer hardware developments. *Will condensed history continue to play a role when computers get much faster?* and *Will analog-based condensation techniques ever replace our analytic-based ones?*.

Will condensed history continue to play a role when computers get much faster?

The other was of asking this question is: *Will analog Monte Carlo techniques replace condensed history methods for most future applications?*

Depending on the application, condensed history techniques "outrun" single-scattering calculations by a factor 10^3—10^5. Computing power per unit cost increases by approximately a factor of 2 every year. This means that an application that runs today with condensed history calculations can be done in the same amount of time by analog methods in about 10–17 years!

The answer to this is that the problems usually expand in complexity as the technology to address them advances. In the next decade or two we will not be asking the same questions! The questions will be more complex and the simpler, cruder method of condensed history, whatever it evolves to in that time, will still have an important role to play.

A perfect example of this is radiotherapy treatment planning calculations. Presently, condensed history techniques are not used because it takes a few hours to perform on a workstation-class computer. Software and hardware technology may reduce this to seconds in about 5 years, making it feasible for routine use. In a few more years, calculation times will be microseconds and Monte Carlo will be used in all phases of treatment planning, even the most sophisticated such as inverse planning and scan-plan-treat single pass tomotherapy machines.

Condensed history gets this answer to sufficient accuracy and medical physics will not resort to single-scattering methods that take 10^3—10^5 longer to execute for marginal (and and largely unnecessary) gain in accuracy. Once calculation error has been reduced to about 2% or so, its contribution to the overall error of treatment delivery will be negligible.

Will analog-based condensation techniques ever replace our analytic-based ones?

One approach to addressing the problem of slow execution for single-scattering Monte Carlo is to pre-compute electron single-scattering histories and tally the emergence of particles from macroscopic objects of various shapes, depending on the application. Then one transports these objects in the application rather than electrons! Ballinger *et. al.* [BRM92] used hemispheres as his intended application was primarily low-energy backscatter from foils. Ballinger *et. al.* did their calculations within the hemispheres almost completely in analog mode, for both elastic and inelastic events.

Neuenschwander and Born [NB92] and later Neuenschwander *et. al.* [Nel95] used EGS4 [NHR85, BHNR94] condensed history methods for pre-calculation in spheres for the intended application of transport within radiotherapy targets (CT-based images) and realized a speed increase of about 11 over condensed history. Svatos *et. al.* [SBN$^+$95] is following up on this work by using analog methods.

Since these "analog-based condensation" techniques play a role in specialized applications it begs the question whether or not these techniques can play a more general role. To answer this, consider that we are seeking the general solution to the problem: given an electron starting at the origin directed along the z-axis for a set of energies E_n, what is the complete description of the "phase space" of particles emerging from a set spheres[5] of radii r_n? That is, what is $\psi(\vec{x}, \vec{\Omega}, E, s, q; E_n, r_n)$, where $\vec{x}$ is the final position on the sphere, $\vec{\Omega}$ is the direction at the exit point of the sphere, E is the exit energy, s is the total pathlength, q is the charge (3 possible values in our model, electrons, positrons or photons), E_n is the starting energy, and r_n is the radius of the sphere. Now, imagine that we require n-points to fit some input or output phase-space variable (*e.g.* 100 different values of E) and that we must provide storage for N decades of input energy. (The input and output energies would likely be tabulated on a logarithmic mesh.) The result is that one would require $3Nn^8$ real words of data to store the results of the general problem!

To make the example more concrete, imagine that we wish to store 9 decades in input energy (from, say, 1 keV to 100 TeV) and set $n = 100$. This would require 1.2 exabytes (1.2×10^{18})

[5] We will use this geometry as an example. A set of spheres is necessary so that geometry-adaptive techniques may be employed.

bytes of information. Estimating current on-line storage capability at about 10 terabytes/m^3, the required storage would be 1.2×10^5 m^3, or a cube about 50 m on a side. This class of solution would require storage densities of the order 10^2–10^3 greater than current technology, something for the distant future, perhaps in 50 years or so?

However, this solution really reflects a paucity of clever ideas. In a previous section we have already seen how multiple-scattering angles can be represented compactly. It is likely that further research may give us more insight into how to represent the data to the entire problem in a compact way. It may turn out that the future of this class of Monte Carlo calculations may be with pre-computed distributions. However, condensed history research will provide the most sensible way to *interpolate* the data. The better the interpolation scheme, the more compact the data will be. This may be the surviving contribution of condensed history research in the distant future.

Bibliography

[AMB93] P. Andreo, J. Medin, and A. F. Bielajew. Constraints on the multiple scattering theory of Molière in Monte Carlo simulations of the transport of charged particles. *Med. Phys.*, 20:1315 – 1325, 1993.

[Ber63] M. J. Berger. Monte Carlo Calculation of the penetration and diffusion of fast charged particles. *Methods in Comput. Phys.*, 1:135 – 215, 1963.

[Bet53] H. A. Bethe. Molière's theory of multiple scattering. *Phys. Rev.*, 89:1256 – 1266, 1953.

[BHNR94] A. F. Bielajew, H. Hirayama, W. R. Nelson, and D. W. O. Rogers. History, overview and recent improvements of EGS4. *National Research Council of Canada Report PIRS-0436*, 1994.

[Bie94] A. F. Bielajew. Plural and multiple small-angle scattering from a screened Rutherford cross section. *Nucl. Inst. and Meth.*, B86:257 – 269, 1994.

[Bie95] A. F. Bielajew. HOWFAR and HOWNEAR: Geometry Modeling for Monte Carlo Particle Transport. *National Research Council of Canada Report PIRS-0341*, 1995.

[Bie96] A. F. Bielajew. A hybrid multiple-scattering theory for electron-transport Monte Carlo calculations. *Nucl. Inst. and Meth.*, B111:195 – 208, 1996.

[Bot21] W. Bothe. Das allgemeine Fehlergesetz, die Schwankungen der Feldstärke in einem Dielektrikum und die Zerstreuung der α-Strahlen. *Z. für Physik*, 5:63 – 69, 1921.

[BR86] A. F. Bielajew and D. W. O. Rogers. PRESTA: The Parameter Reduced Electron-Step Transport Algorithm for electron Monte Carlo transport. *National Research Council of Canada Report PIRS-0042*, 1986.

[BR87] A. F. Bielajew and D. W. O. Rogers. PRESTA: The Parameter Reduced Electron-Step Transport Algorithm for electron Monte Carlo transport. *Nuclear Instruments and Methods*, B18:165 – 181, 1987.

[Bri86] J. Briesmeister. MCNP—A general purpose Monte Carlo code for neutron and photon transport, Version 3A. *Los Alamos National Laboratory Report LA-7396-M (Los Alamos, NM)*, 1986.

[Bri93] J. F. Briesmeister. MCNP—A general Monte Carlo N-particle transport code. *Los Alamos National Laboratory Report LA-12625-M (Los Alamos, NM)*, 1993.

[BRM92] C. T. Ballinger, J. A. Rathkopf, and W. R. Martin. The response history Monte Carlo method for electron transport. *Nucl. Sci. Eng.*, 112:283 – 295, 1992.

[BW89] M. J. Berger and R. Wang. Multiple-scattering angular deflections and energy-loss straggling. In T.M. Jenkins, W.R. Nelson, A. Rindi, A.E. Nahum, and D.W.O. Rogers, editors, *Monte Carlo Transport of Electrons and Photons*, pages 21 – 56. Plenum Press, New York, 1989.

[FS95] B. J. Foote and V. G. Smyth. The modeling of electron multiple-scattering in EGS4/PRESTA and its effect on ionization-chamber response. *Nucl. Inst. and Meth.*, B100:22 – 30, 1995.

[GS40a] S. A. Goudsmit and J. L. Saunderson. Multiple scattering of electrons. *Phys. Rev.*, 57:24 – 29, 1940.

[GS40b] S. A. Goudsmit and J. L. Saunderson. Multiple scattering of electrons. II. *Phys. Rev.*, 58:36 – 42, 1940.

[HKM$^+$92] J. A. Halbleib, R. P. Kensek, T. A. Mehlhorn, G. D. Valdez, S. M. Seltzer, and M. J. Berger. ITS Version 3.0: The Integrated TIGER Series of coupled electron/photon Monte Carlo transport codes. *Sandia report SAND91-1634*, 1992.

[Kaw96a] I. Kawrakow. Electron transport: longitudinal and lateral correlation algorithm. *Nuclear Instruments and Methods*, B114:307 – 326, 1996.

[Kaw96b] I. Kawrakow. Electron transport: multiple and plural scattering. *Nuclear Instruments and Methods*, B108:23 – 34, 1996.

[KB98] I. Kawrakow and A. F. Bielajew. On the representation of electron multiple elastic-scattering distributions for Monte Carlo calculations. *Nuclear Instruments and Methods*, B134:325 – 336, 1998.

[Lar92] E. W. Larsen. A theoretical derivation of the condensed history algorithm. *Ann. Nucl. Energy*, 19:701 – 714, 1992.

[Lew50] H. W. Lewis. Multiple scattering in an infinite medium. *Phys. Rev.*, 78:526 – 529, 1950.

[Mol47] G. Z. Molière. Theorie der Streuung schneller geladener Teilchen. I. Einzelstreuung am abgeschirmten Coulomb-Field. *Z. Naturforsch*, 2a:133 – 145, 1947.

[Mol48] G. Z. Molière. Theorie der Streuung schneller geladener Teilchen. II. Mehrfach- und Vielfachstreuung. *Z. Naturforsch*, 3a:78 – 97, 1948.

[Mot29] N. F. Mott. . *Proc. Royal Society London*, A124:425, 1929.

[Mot32] N. F. Mott. . *Proc. Royal Society London*, A135:429, 1932.

[NB92] H. Neuenschwander and E. J. Born. A Macro Monte Carlo method for electron beam dose calculations. *Phys. Med. Biol.*, 37:107 – 125, 1992.

[Nel95] W. R. Nelson. private communication. *(conversation with A.F. Bielajew)*, 1995.

[NHR85] W. R. Nelson, H. Hirayama, and D. W. O. Rogers. The EGS4 Code System. Report SLAC–265, Stanford Linear Accelerator Center, Stanford, Calif, 1985.

[Rog84] D. W. O. Rogers. Low energy electron transport with EGS. *Nucl. Inst. Meth.*, 227:535 – 548, 1984.

[Rog93] D. W. O. Rogers. How accurately can EGS4/PRESTA calculate ion chamber response? *Medical Physics*, 20:319 – 323, 1993.

[SBN$^+$95] M. M. Svatos, C. T. Ballinger, H. Neuenschwander, T. R. Mackie, W. P. Chandler, C. L. Hartmann-Siantar, J. A. Rathkopf, and P. J. Reckwerdt. Electron transport in radiotherapy using local-to-global Monte Carlo. *In "Proceedings of the International Conference on Mathematics and Computations, Reactor Physics, and Environmental Analyses" (American Nuclear Society Press, La Grange Park, Illinois, U.S.A.)*, pages 154 – 161, 1995.

[Sel89] S. M. Seltzer. An overview of ETRAN Monte Carlo methods. In T.M. Jenkins, W.R. Nelson, A. Rindi, A.E. Nahum, and D.W.O. Rogers, editors, *Monte Carlo Transport of Electrons and Photons*, pages 153 – 182. Plenum Press, New York, 1989.

[Sel91] S. M. Seltzer. Electron-photon Monte Carlo calculations: the ETRAN code. *Int'l J of Appl. Radiation and Isotopes*, 42:917 – 941, 1991.

[Wen22] G. Wentzel. Zur theorie der streuung von β-strahlen. *Ann. Physik*, 69:335 – 368, 1922.

[Win87] K. B. Winterbon. Finite-angle multiple scattering. *Nuclear Instruments and Methods*, B21:1 – 7, 1987.

Problems

1.

Chapter 16

Electron Transport in Electric and Magnetic Fields

In this lecture we discuss the fundamentals of electron transport in static external electric and magnetic fields in vacuum and dense media. By "static" and "external" is meant that macroscopic $\vec{\mathbf{E}}$ and/or $\vec{\mathbf{B}}$ fields are set up in the region where the electron transport is taking place. For example, a high-energy particle detector may be placed in a constant magnetic field so that the momentum of charged particles may be analyzed. The external fields are considered to be static in the sense that they do not change with time during the course of the simulations. This is not a fundamental constraint but it is imposed for simplicity. The bulk of the discussion concerns the theoretical viability of performing electron transport in dense media in the presence of external fields. The trajectories of particles in this case can be quite complicated. The particles can be subjected to a myriad of forces—de-accelerations due to inelastic processes with orbital electrons, elastic deflections due to attraction or repulsion in the nuclear electric field, acceleration or de-acceleration by the external electric field, and deflections by the external electric and magnetic fields.

In comparison to the effects of the internal processes of multiple scattering and inelastic collisions, the effect of the external fields can be quite dramatic. Electric field strengths can be as high as 2 MV/(g/cm^2). The rate of a charged particle's change in energy due to this field can be equal in magnitude to the rate of energy loss of high energy electrons in matter. We wish to establish a method, even if it is a "brute force" one, that will allow us to do charged particle transport under these circumstances. We do not wish to treat the effects of the external fields as perturbations on the field-free transport in media. Yet, we do not wish to discard all the theoretical work that has been achieved in field-free transport. Rather, we shall retain what we know about inelastic energy-loss mechanisms and multiple scattering and attempt to include the effect of the external fields, albeit in a simple-minded fashion.

We commence the lecture with a "review" discussion of charged particle transport in external fields. We set up the equations and then solve them in vacuum. The vacuum solutions will play a role in the benchmarking of the differential equations as modelled in the Monte Carlo code. We then prove formally under what circumstances the vacuum transport equations can be "tacked on" to the field-free transport with little error. In general, it will be shown that the vacuum external field transport can be superimposed upon the field-free transport as long as the charged particle steps are short enough.

In the last part of the lecture we discuss the practical application of the external field transport equations to a Monte Carlo code. As a specific example, we shall use the EGS code [NHR85] with which the author is most familiar. The method of application discussed will be very general and may be applied equally well to any condensed-history electron transport code. The ITS codes [HM84] have also been adapted for transport in external fields [Hal74, HSV75, HSV77].

16.1 Equations of motion in a vacuum

In this section we derive the differential equations of motion for charged particles moving in arbitrary electric and magnetic fields. In this section we consider transport in a vacuum.

The change in momentum with time due to electric, $\vec{\mathbf{E}}$, and magnetic, $\vec{\mathbf{B}}$, fields is given by the Lorentz force equation [Jac75],

$$\frac{d\vec{\mathbf{p}}}{dt} = e(\vec{\mathbf{E}} + \vec{\mathbf{v}} \times \vec{\mathbf{B}}), \tag{16.1}$$

where $\vec{\mathbf{p}}$ is the momentum, t is the time, e is the charge of the particle (negative for negatrons and positive for positrons), and $\vec{\mathbf{v}}$ is the velocity. This equation may be cast in the following form:

$$\frac{d(\gamma\vec{\boldsymbol{\beta}})}{ds} = \frac{e}{m_0c^2\beta}(\vec{\mathbf{E}} + c\vec{\boldsymbol{\beta}} \times \vec{\mathbf{B}}), \tag{16.2}$$

where c is the speed of light, $\vec{\boldsymbol{\beta}}$ is the velocity in units of the speed of light, $\vec{\boldsymbol{\beta}} = \vec{\mathbf{v}}/c$, γ is the familiar relativistic factor, $(1 - \vec{\boldsymbol{\beta}} \cdot \vec{\boldsymbol{\beta}})^{-1/2}$, ds is the differential path-length that can be related to the time, t, through the relation, $ds = (ds/dt)dt = (c\beta)dt$, and m_0 is the rest mass of the charged particle.

If one applies the differential operator to the left hand side of eq. 17.3, expresses $d\gamma$ as $\gamma^3\vec{\boldsymbol{\beta}}(\vec{\boldsymbol{\beta}} \cdot \boldsymbol{d}\vec{\boldsymbol{\beta}})$, and takes the inner product of both sides of the equation with $\vec{\boldsymbol{\beta}}$, then one can obtain the equation,

$$\frac{d\vec{\boldsymbol{\beta}}}{ds} = \frac{e}{m_0c^2\beta\gamma}(\vec{\mathbf{E}} - \vec{\boldsymbol{\beta}}(\vec{\mathbf{E}}{\cdot}\vec{\boldsymbol{\beta}}) + c\vec{\boldsymbol{\beta}}{\times}\vec{\mathbf{B}}). \tag{16.3}$$

One can perform similar manipulations to derive a particularly useful form of this equation,

$$\frac{d\vec{\mathbf{u}}}{ds} = \frac{e}{m_0c^2\beta^2\gamma}(\vec{\mathbf{E}} - \vec{\mathbf{u}}(\vec{\mathbf{E}} \cdot \vec{\mathbf{u}}) + c\beta\vec{\mathbf{u}} \times \vec{\mathbf{B}}), \tag{16.4}$$

where $\vec{\mathbf{u}}$ is the unit direction vector of the charged particle[1]. We note, from eq. 17.5, that $\vec{\mathbf{u}} \cdot (d\vec{\mathbf{u}}/ds) = 0$. This means that the change in the direction vector is transverse to it's direction. This also follows directly from the fact that the magnitude is, by fiat, unity, *i.e.* $\vec{\mathbf{u}} \cdot \vec{\mathbf{u}} = 1$, from which it follows that $d(\vec{\mathbf{u}} \cdot \vec{\mathbf{u}}) = 2\vec{\mathbf{u}}{\cdot}d\vec{\mathbf{u}} = 0$. We shall make use of eq. 17.5 when applying external field transport to a Monte Carlo code.

16.1.1 Special cases: $\vec{\mathbf{E}}$ =constant, $\vec{\mathbf{B}} = 0$; $\vec{\mathbf{B}}$ =constant, $\vec{\mathbf{E}} = 0$

In this section we present the solutions for the special cases of uniform electric and magnetic fields in vacuum. These solutions may be used by Monte Carlo codes for transport in vacuum and they may be used for "benchmarking" the transport equations. Rather than repeat derivations which demand much careful arithmetic manipulation, we simply present the solutions and encourage the reader to verify them at his leisure.

It can be shown that the general solution for the equation of motion in the constant electric field case takes the form,

$$x_{\|} = \frac{m_0c^2\gamma_0}{eE}\left(\cosh\left(\frac{eEx_\perp}{m_0c^2\gamma_0\beta_{\perp 0}}\right) - 1 + \beta_{\|0}\sinh\left(\frac{eEx_\perp}{m_0c^2\gamma_0\beta_{\perp 0}}\right)\right), \tag{16.5}$$

where $x_{(\|,\perp)}$ is the distance from the origin in the direction (parallel, perpendicular) to $\vec{\mathbf{E}}$, γ_0 is the value of γ at $x_\| = 0$ and $x_\perp = 0$, $E = |\vec{\mathbf{E}}|$, and $\beta_{(\|,\perp)0}$ is the initial component of $\vec{\boldsymbol{\beta}}$ in the direction (parallel, perpendicular) to $\vec{\mathbf{E}}$.

This equation contains two interesting limiting cases. In the "weak-field" limit (WFL), $\frac{eEx_\perp}{m_0c^2\gamma_0\beta_{\perp 0}} \to 0$, eq. 17.6 becomes,

$$\lim_{\text{WFL}} x_\| \longrightarrow \frac{\beta_{\|0}x_\perp}{\beta_{\perp 0}} + \frac{eEx_\perp^2}{2m_0c^2\gamma_0\beta_{\perp 0}^2}. \tag{16.6}$$

The non-relativistic limit (NRL) may be obtained from eq. 17.7 by taking the limit, $\gamma_0 \to 1$, with the result,

$$\lim_{\text{NRL}} x_\| \longrightarrow v_{\|0}t + \frac{eEt^2}{2m_0}, \tag{16.7}$$

where $v_{(\|,\perp)0} = c\beta_{(\|,\perp)0}$, and t, the time, is expressed as $t = x_{\perp 0}/v_{\perp 0}$. The NRL solution, a parabola, should be familiar from elementary mechanics.

Note that the previous three equations, eqs. 17.6–17.8, are valid for positions ($e = |e|$) and negatrons ($e = -|e|$). As well, if the particle starts out in a direction opposing the electric

[1] This equation, with $\vec{\mathbf{B}} = 0$, was stated erroneously in a previous reference [RBMG84]. The error, $\vec{\mathbf{u}}$ was mistakenly set to $\vec{\beta}$ in eq. 17.5, was inconsequential since $\beta \approx 1$ in the calculations associated with that report.

force, then $x_\parallel$ can be a double-valued function of $x_\perp$ or t. Additionally, as a consequence of the decomposition into parallel and perpendicular components, the motion always takes place in the plane defined by $\beta_{\parallel 0}$ and $\beta_{\perp 0}$.

The equation of motion in a constant magnetic field should be familiar from elementary mechanics for either relativistic or non-relativistic charged particles. The development of the solution is considerably simplified owing to the fact that the magnetic field does not alter the particle's energy. This results directly from the force and the particle's velocity being always perpendicular. The equations of motion are,

$$x_{\perp 1} = \frac{p^0_{\perp 2}}{eB}\left((1 - \cos(\frac{eBx_\parallel}{p^0_\parallel})\right) + \frac{p^0_{\perp 1}}{eB}\sin(\frac{eBx_\parallel}{p^0_\parallel}), \tag{16.8}$$

and

$$x_{\perp 2} = -\frac{p^0_{\perp 1}}{eB}\left(1 - \cos(\frac{eBx_\parallel}{p^0_\parallel})\right) + \frac{p^0_{\perp 2}}{eB}\sin(\frac{eBx_\parallel}{p^0_\parallel}). \tag{16.9}$$

In these equations, $B = |\vec{\mathbf{B}}|$, and the direction of motion has been resolved into 3 components, $(x_{\perp 1},\, x_{\perp 2},\, x_\parallel)$, forming an orthogonal triad with $x_\parallel$ aligned with the $\vec{\mathbf{B}}$-field. The momentum is also resolved into the three components, $(p_{\perp 1},\, p_{\perp 2},\, p_\parallel)$, with initial values, $(p^0_{\perp 1},\, p^0_{\perp 2},\, p^0_\parallel)$ when $(x^0_{\perp 1},\, x^0_{\perp 2},\, x^0_\parallel)$=(0, 0, 0). The component of momentum parallel to $\vec{\mathbf{B}}$ is a constant of the motion, *i.e.* $p_\parallel$=$p^0_\parallel$, as is the energy and the velocity in the direction of the $\vec{\mathbf{B}}$-field.

As seen from eqs. 17.9 and 17.10, the motion in the $x_{\perp 1}$–$x_{\perp 2}$ plane is a circle centered at $(p^0_{\perp 2}/eB,\, -p^0_{\perp 1}/eB)$ with radius $|p^0_\perp/(eB)|$ where $|p^0_\perp| = \sqrt{(p^0_{\perp 1})^2 + (p^0_{\perp 2})^2}$. For relativistic particles one should substitute $p^0 = m_0\gamma_0 c\beta$ and $p^0 = m_0 v_0$ may be used for non-relativistic particles. In other words, the particle "spirals" along the $\vec{\mathbf{B}}$-axis with a constant speed and radius. If one "sights" along the $\vec{\mathbf{B}}$-axis, positrons move in the counter-clockwise direction while negatrons move in the clockwise direction.

16.2 Transport in a medium

The transport of charged particles in media with external electric and magnetic fields can be quite complicated and difficult to handle theoretically. However, we still can accomplish the simulation of charged particle transport in media with certain approximations. In this section we state these approximations using a very general development.

Consider the motion of a charged particle in a medium with external electric and magnetic fields. Besides the coupling to the external fields, the particle is acted upon by elastic and inelastic forces. Assuming the medium is isotropic and homogeneous, the equation of motion takes the general form,

$$\frac{d\vec{\mathbf{p}}}{dt} = \vec{\mathbf{F}}_{ret}(E(t)) + \vec{\mathbf{F}}_{ms}(E(t)) + \vec{\mathbf{F}}_{em}(\vec{\mathbf{x}}(t), E(t), \vec{\mathbf{u}}(t)), \tag{16.10}$$

where $\vec{\mathbf{p}}$ is the momentum, t is the time, $\vec{\mathbf{F}}_{ret}$ is the force due to inelastic (retarding) forces, $\vec{\mathbf{F}}_{ms}$ is the force due to elastic (multiple scattering) forces, and $\vec{\mathbf{F}}_{em}$ is the force due to external (electric and magnetic) forces. We may integrate eq. 16.10 implicitly to obtain,

$$\vec{\mathbf{v}} = \vec{\mathbf{v}}_0 + \frac{1}{m_0\gamma(E)} \int_0^t dt' \; \{\vec{\mathbf{F}}_{ret}(E(t')) + \vec{\mathbf{F}}_{ms}(E(t')) + \vec{\mathbf{F}}_{em}(\vec{\mathbf{x}}(t'), E(t), \vec{\mathbf{u}}(t'))\}. \tag{16.11}$$

$$\vec{\mathbf{x}} = \vec{\mathbf{x}}_0 + \vec{\mathbf{v}}_0 t + \int_0^t dt'' \vec{\mathbf{v}}(t'') \tag{16.12}$$

These are very complex equations of motion with much interplay among all the constituents. $\vec{\mathbf{F}}_{ret}$ accounts for inelastic processes having to do mostly with electron-electron interactions and bremsstrahlung photon creation in the nuclear field. This force affects mostly E, the energy, and consequently v, the magnitude of the velocity, $\vec{\mathbf{v}}$. There is some deflection as well but angular deflection is dominated by multiple scattering. $\vec{\mathbf{F}}_{ret}$ couples to $\vec{\mathbf{F}}_{ms}$ and $\vec{\mathbf{F}}_{em}$ because they all depend on E. $\vec{\mathbf{F}}_{ms}$ accounts for elastic processes having to do mostly with deflections caused by the nuclei of the medium. It changes the direction of the velocity. Consequently, $\vec{\mathbf{F}}_{ms}$ couples to $\vec{\mathbf{F}}_{em}$ since the latter depends on $\vec{\mathbf{u}}$, the direction of motion of the particle. (By definition, $\vec{\mathbf{u}}$ is a unit vector.) The energy lost to the nuclear recoil can usually be ignored and $\vec{\mathbf{F}}_{ms}$ does not couple to $\vec{\mathbf{F}}_{ret}$. $\vec{\mathbf{F}}_{em}$ accounts for the interaction with the external electric and magnetic fields. It depends on E, $\vec{\mathbf{u}}$ and also $\vec{\mathbf{x}}$ if the external fields are spatially dependent. $\vec{\mathbf{F}}_{em}$ can alter both the magnitude and direction of $\vec{\mathbf{v}}$ thereby coupling directly to both $\vec{\mathbf{F}}_{ms}$ and $\vec{\mathbf{F}}_{ret}$. Moreover, outside the integral in eq. 16.11 is an overall factor of $1/(m_0\gamma(E))$, owing to the fact that the mass changes when the energy changes.

To complicate matters even further, we do not know the exact nature of $\vec{\mathbf{F}}_{ret}$ and $\vec{\mathbf{F}}_{ms}$. For "microscopic" Monte Carlo methods, where we model every electron interaction, we can only say something about the momentum before and after the interaction. For complete rigour, one would have to solve the quantum mechanical equations of motion incorporating external fields. However, unless the external fields are very strong, they may be treated in a perturbation formalism for microscopic Monte Carlo methods.

In this lecture we restrict ourselves to "condensed-history" Monte Carlo, for which the arguments are even more subtle. For complete rigour, one should incorporate the external fields into the Boltzmann transport equation and solve directly. To our knowledge, this has not been attempted. Instead, we attempt to superimpose the transport in the external fields upon "field-free" charged particle transport and discuss what approximations need to be made.

A major difficulty arises from the fact that during a condensed-history transport step the trajectory of the particle is not known nor are the exact forms of $\vec{\mathbf{F}}_{ret}$ and $\vec{\mathbf{F}}_{ms}$ known. Instead we wish to make use of the already existing statistical treatments of $\vec{\mathbf{F}}_{ret}$ and $\vec{\mathbf{F}}_{ms}$, for example, Bethe-Bloch slowing down theory [Bet30, Bet32, Blo33] and Molière [Mol47, Mol48] multiple scattering theory. If the external fields are different for the different possible particle trajectories, we are faced an unresolvable ambiguity. Therefore, we must demand that the *transport steps be small enough so that the fields do not change very much over the course of the step.* With this approximation, eq. 16.11 becomes,

$$\vec{\mathbf{v}} = \vec{\mathbf{v}}_0 + \frac{1}{m_0\gamma(E)} \int_0^t dt' \; \{\langle \vec{\mathbf{F}}_{ret}(E(t'))\rangle + \langle \vec{\mathbf{F}}_{ms}(E(t'))\rangle + \vec{\mathbf{F}}_{em}(\vec{\mathbf{x}}_0, E(t), \vec{\mathbf{u}}(t'))\}, \qquad (16.13)$$

where $\vec{\mathbf{x}}_0$ is the position at the beginning of the particle's step and the $\langle\vec{\mathbf{F}}\rangle$'s denote that we are now employing the statistical formulations of the physical effects produced by $\langle\vec{\mathbf{F}}\rangle$.

We make the further approximation that the energy does not change very much during the course of a particle step. With the approximation of small energy losses, eq. 16.13 becomes,

$$\vec{\mathbf{v}} = \vec{\mathbf{v}}_0 + \frac{1}{m_0\gamma(E_0)} \left\{ \langle \vec{\mathbf{F}}_{ret}(E_0)\rangle t + \langle \vec{\mathbf{F}}_{ms}(E_0)\rangle t + \int_0^t dt' \; \vec{\mathbf{F}}_{em}(\vec{\mathbf{x}}_0, E_0, \vec{\mathbf{u}}(t'))\right\}, \qquad (16.14)$$

where E_0 is the energy evaluated at the beginning of the particle's step.

Finally, we make the approximation that the direction angle $\vec{\mathbf{u}}$ does not change much over the course of the step. While this can be accomplished by reducing the size of the step for most of the charged particle steps, occasionally large angle scatterings associated with single nucleus-electron interactions will occur. Therefore, this approximation must break down at least some of the time. Fortunately, multiple scattering is dominated by small angle events with only relatively few large angle ones [Ber63]. With this approximation,

$$\vec{\mathbf{v}} = \vec{\mathbf{v}}_0 + \frac{t}{m_0\gamma(E_0)}\{\langle \vec{\mathbf{F}}_{ret}(E_0)\rangle + \langle \vec{\mathbf{F}}_{ms}(E_0)\rangle + \vec{\mathbf{F}}_{em}(\vec{\mathbf{x}}_0, E_0, \vec{\mathbf{u}}_0)\}. \qquad (16.15)$$

where $\vec{\mathbf{u}}_0$ is the direction vector evaluated at the beginning of the particle's step. At this point, by virtue of the approximations we have made, the $\vec{\mathbf{F}}$'s are decoupled. Note that nothing has been said about the sizes of the $\vec{\mathbf{F}}$'s relative to one another except that they do not perturb the "force-free" trajectory too much.

To make a closer connection to applying the equations to Monte Carlo simulations, we recast the equations so that they are dependent upon the total path-length of the step, s, rather than the time. To this end we write,

$$t = \int_0^s \frac{ds}{v}, \qquad (16.16)$$

which, to 1^{st}-order may be written,

$$t = \frac{s}{v_0}\left(1 + \frac{\Delta v(E_0)}{v_0}\right), \qquad (16.17)$$

where $\Delta v(E_0) = v_0 - v(E_0)$ accounts for energy losses in the relationship expressed by eq. 16.16. From eq. 16.15, we find that to 1^{st}-order in the $\vec{\mathbf{F}}$'s,

$$\Delta v(E_0) = -\frac{s}{m_0\gamma(E_0)v_0}\vec{\mathbf{u}}_0 \cdot \{\langle\vec{\mathbf{F}}_{ret}(E_0)\rangle + \langle\vec{\mathbf{F}}_{ms}(E_0)\rangle + \vec{\mathbf{F}}_{em}(\vec{\mathbf{x}}_0, E_0, \vec{\mathbf{u}}_0)\}, \tag{16.18}$$

where $\vec{\mathbf{u}}_0$ is the direction vector evaluated at the start of the electron step. Using eqs. 16.12, 16.15–16.18 we find that the new direction vector, $\vec{\mathbf{u}} = \vec{\mathbf{v}}/|\vec{\mathbf{v}}|$ takes the form,

$$\vec{\mathbf{u}} = \vec{\mathbf{u}}_0 + \Delta\vec{\mathbf{u}}, \tag{16.19}$$

where,

$$\Delta\vec{\mathbf{u}} = \frac{s}{m_0\gamma(E_0)v_0^2}\{\langle\vec{\mathbf{F}}_{\perp,ret}(E_0)\rangle + \langle\vec{\mathbf{F}}_{\perp,ms}(E_0)\rangle + \vec{\mathbf{F}}_{\perp,em}(\vec{\mathbf{x}}_0, E_0, \vec{\mathbf{u}}_0)\}, \tag{16.20}$$

The $\vec{\mathbf{F}}_\perp$'s are the components of the $\vec{\mathbf{F}}$'s in the direction perpendicular to $\vec{\mathbf{u}}_0$. In similar fashion we find that,

$$\vec{\mathbf{x}} = \vec{\mathbf{x}}_0 + \vec{\mathbf{u}}_0 s + \frac{s}{2}\Delta\vec{\mathbf{u}} \tag{16.21}$$

We may write eq. 16.21 slightly differently as,

$$\vec{\mathbf{x}} = \vec{\mathbf{x}}_0 + \vec{\mathbf{u}}_0 s + \vec{\boldsymbol{\epsilon}}^{(1)}_{\perp,ret+ms} + \vec{\boldsymbol{\epsilon}}^{(1)}_{\perp,em}, \tag{16.22}$$

where the $\vec{\boldsymbol{\epsilon}}^{(1)}$'s are the 1^{st}-order perturbations of the trajectory due to inelastic slowing down plus multiple scattering and the deflection and energy change in the external electric and magnetic fields. Since the $\vec{\boldsymbol{\epsilon}}^{(1)}$'s are decoupled, we may calculate $\vec{\boldsymbol{\epsilon}}^{(1)}_{\perp,ret+ms}$ as if there were no external fields and $\vec{\boldsymbol{\epsilon}}^{(1)}_{\perp,em}$ as if the transport had occurred in a vacuum. We note that, to 1^{st}-order, the deflections are perpendicular to the initial trajectory. If we had carried out the analysis to 2^{nd}-order in the $\vec{\mathbf{F}}$'s, we would obtain an equation of the form,

$$\vec{\mathbf{x}} = \vec{\mathbf{x}}_0 + \vec{\mathbf{u}}_0 s(1 - \epsilon^{(2)}_{\parallel}) + \vec{\boldsymbol{\epsilon}}^{(1)}_{\perp,ret+ms} + \vec{\boldsymbol{\epsilon}}^{(1)}_{\perp,em} + \vec{\boldsymbol{\epsilon}}^{(2)}_{\perp}, \tag{16.23}$$

where $\vec{\boldsymbol{\epsilon}}^{(2)}_{\perp}$ is the 2^{nd}-order perpendicular deflection, and $1-\epsilon^{(2)}_{\parallel}$ can be identified as the *path-length correction*, a quantity that accounts for the curvature of the charged particle step. Rather than transporting the charged particle the full path-length s in the initial direction, this distance must be shortened owing to the deflection of the particle during the step. This is seen to be a 2^{nd}-order effect. We have seen in Lecture 16 that these corrections can be quite large if one attempts to use large step-sizes in field-free simulations. Clearly, if the 1^{st}-order effects of multiple scattering, slowing down or speeding up, and deflection due to the external field are to be considered small in a perturbation sense, then one should ensure the the 2^{nd}-order affects should be commensurately smaller. A reliable method for calculating path-length corrections in field-free transport is discussed in Lecture 16. The terms $\vec{\boldsymbol{\epsilon}}_\perp^{(2)}$ and $\epsilon_\parallel^{(2)}$ contain "mixing terms" between the field-free and external field components. To our knowledge, no theory has yet been developed to encompass all of these effects. Until such time that such a theory is developed, we restrict ourselves to charged particle steps that are small enough so that the 1^{st}-order equations discussed herein are valid. However, if we make this restriction, we may transport particles in any medium with any external field configuration, no matter how strong.

16.3 Application to Monte Carlo, Benchmarks

At the start of a transport step in a medium with external fields present, we know $\vec{\mathbf{x}}_0$, the position, $\vec{\mathbf{u}}_0$, the unit direction vector, E_0, the energy (which one may easily use to find v_0, the magnitude of the velocity, or $\beta_0 = v_0/c$), $\vec{\mathbf{D}}_0$, the macroscopic electric field strength at $\vec{\mathbf{x}}_0$, and $\vec{\mathbf{H}}_0$, the macroscopic magnetic field strength at $\vec{\mathbf{x}}_0$. In the condensed-history approach to electron Monte Carlo transport, aside from those discrete interactions which we consider explicitly, the interactions are grouped together and treated by theories which consider the medium in which the transport takes place to be a homogeneous, bulk medium. Therefore, using the macroscopic fields, $\vec{\mathbf{D}}$ and $\vec{\mathbf{H}}$, rather than the microscopic fields, $\vec{\mathbf{E}}$ and $\vec{\mathbf{B}}$, is consistent with the condensed-history approach. After the transport step, we wish to know $\vec{\mathbf{x}}_f$, $\vec{\mathbf{u}}_f$, and E_f, the final position, unit direction vector, and energy.

If we make the identification,

$$\vec{\mathbf{F}}_{\perp,em}(\vec{\mathbf{x}}_0, E_0, \vec{\mathbf{u}}_0) = e(\vec{\mathbf{D}}_0 - \vec{\mathbf{u}}_0(\vec{\mathbf{u}}_0 \cdot \vec{\mathbf{D}}_0) + \vec{\mathbf{v}}_0 \times \vec{\mathbf{H}}_0), \tag{16.24}$$

we may use the results of the previous section and write directly the equation, for $\vec{\mathbf{u}}_f$,

$$\vec{\mathbf{u}}_f = \vec{\mathbf{u}}_0 + \Delta\vec{\mathbf{u}}_{ms,ret} + \Delta\vec{\mathbf{u}}_{em}, \tag{16.25}$$

where the deflection due to multiple scattering and inelastic collisions is:

$$\Delta\vec{\mathbf{u}}_{ms,ret} = \frac{s}{m_0\gamma(E_0)v_0^2}\{\langle\vec{\mathbf{F}}_{\perp,ret}(E_0)\rangle + \langle\vec{\mathbf{F}}_{\perp,ms}(E_0)\rangle\}, \tag{16.26}$$

and the deflection due to the external electromagnetic field is:

$$\Delta\vec{\mathbf{u}}_{em} = \frac{es}{m_0\gamma(E_0)v_0^2}(\vec{\mathbf{D}}_0 - \vec{\mathbf{u}}_0(\vec{\mathbf{u}}_0 \cdot \vec{\mathbf{D}}_0) + \vec{\mathbf{v}}_0 \times \vec{\mathbf{H}}_0). \tag{16.27}$$

Using the results of the previous section, we may also write directly the equation for $\vec{\mathbf{x}}_f$,

$$\vec{\mathbf{x}}_f = \vec{\mathbf{x}}_0 + \vec{\mathbf{u}}_0 s + \frac{s}{2}(\Delta\vec{\mathbf{u}}_{ms,ret} + \Delta\vec{\mathbf{u}}_{em}). \tag{16.28}$$

We may also have written these equations from the results of the vacuum transport section, sec. 16.1, by superimposing the inelastic and multiple scattering forces on the transport in a vacuum but with external electric and magnetic fields.

We wish to ensure that new direction vector remains properly normalized, *i.e.* $|\vec{\mathbf{u}}_f| = 1$. Therefore, after the transport has taken place according to eqs. 16.24–16.28, we normalize $\vec{\mathbf{u}}_f$ using,

$$\vec{\mathbf{u}}'_f = \frac{\vec{\mathbf{u}}_f}{\sqrt{1 + |\Delta\vec{\mathbf{u}}_{ms,ret}|^2 + 2\Delta\vec{\mathbf{u}}_{ms,ret} \cdot \Delta\vec{\mathbf{u}}_{em} + |\Delta\vec{\mathbf{u}}_{em}|^2}}. \tag{16.29}$$

Note that there is no inherent contradiction introduced by eq. 16.29 because the normalization uses terms that are higher than 1^{st}-order in the $\Delta\vec{\mathbf{u}}$'s. We remark that $\vec{\mathbf{u}}_f$ is already normalized to 1^{st}-order in the $\Delta\vec{\mathbf{u}}$'s, consistent with our development. For repeated transport steps, the $\vec{\mathbf{u}}_f$ we calculate becomes the $\vec{\mathbf{u}}_0$ of the next step. We have assumed that $\vec{\mathbf{u}}_0$ is properly normalized in our development. Unless we obey this normalization condition to all orders, the 2^{nd} and higher-order terms will eventually accumulate and cause substantial error.

Finally, we calculate the energy loss from the equation,

$$E_f = E_0 - \Delta E_{ret} + e\vec{\mathbf{D}}_0 \cdot (\vec{\mathbf{x}}_f - \vec{\mathbf{x}}_0), \tag{16.30}$$

where ΔE_{ret} is the energy loss due to inelastic collisions. It takes the form,

$$\Delta E_{ret} = \int_0^s ds'|dE/ds'|, \tag{16.31}$$

where $|dE/ds'|$ is the stopping power.

At this point we should re-state the constraints we must impose so that our method is valid. The condition that the fields not change very much over the transport step takes the form,

$$\frac{|\vec{\mathbf{D}}(\vec{\mathbf{x}}_{f)} - \vec{\mathbf{D}}(\vec{\mathbf{x}}_0)|}{|\vec{\mathbf{D}}(\vec{\mathbf{x}}_0|)} \equiv \delta_D \ll 1, \tag{16.32}$$

and,

$$\frac{|\vec{\mathbf{H}}(\vec{\mathbf{x}}_f) - \vec{\mathbf{H}}(\vec{\mathbf{x}}_0)|}{|\vec{\mathbf{H}}(\vec{\mathbf{x}}_0)|} \equiv \delta_H \ll 1. \tag{16.33}$$

The constraint that the energy does not change very much over the transport step takes the form,

$$\frac{\Delta E_{ret}}{E_{0,kinetic}} \equiv \delta_{ret} \ll 1, \tag{16.34}$$

and,

$$\frac{|\Delta E_{em}|}{E_{0,kinetic}} \equiv \delta_{em} \ll 1. \tag{16.35}$$

Finally, the constraint that the direction does not change very much over the transport step takes the form,

$$|\Delta\vec{\mathbf{u}}_{ms,ret}| \ll 1, \tag{16.36}$$

and,

$$|\Delta\vec{\mathbf{u}}_{em}| \ll 1. \tag{16.37}$$

Recall that all of the constraints expressed by eqs. 16.32–16.37 can be satisfied by using a small electron step-size except $|\Delta\vec{\mathbf{u}}_{ms,ret}| \ll 1$, which must be violated some of the time if the multiple scattering model includes large-angle single-event Rutherford scattering. We have argued that this occurs infrequently enough so that the error introduced is negligible. The error, if any, would show up in high-Z media subjected to strong magnetic fields, since high-Z media produce more large-angle events and the magnetic force is direction dependent.

An algorithm for electron transport in media subjected to external electric and magnetic fields would be of the following form:

- Choose a total path-length for the step, s, consistent with the constraints expressed in eqs. 16.32–16.37.
- Calculate $\vec{\mathbf{u}}_f$ using eqs. 16.25–16.27. The inelastic and elastic scattering portions, $\langle\vec{\mathbf{F}}_{\perp,ret}(E_0)\rangle$ and $\langle\vec{\mathbf{F}}_{\perp,ms}(E_0)\rangle$, can be calculated using any slowing-down and multiple scattering theory.
- Transport the particle according to eq. 16.28[2].
- Normalize the direction vector, $\vec{\mathbf{u}}_f$, using eq. 16.29[3].
- Calculate the new energy according to eqs. 16.30 and 16.31.
- Repeat.

To proceed further would entail detailed knowledge of a specific Monte Carlo code. This would take us beyond the scope of this lecture. Rather, we assert that this has been done

[2] The EGS code simplifies the procedure even more [RBMG84] and ignores the lateral transport caused by $\langle\vec{\mathbf{F}}_{\perp,ret}\rangle$, $\langle\vec{\mathbf{F}}_{\perp,ms}\rangle$, and $\vec{\mathbf{F}}_{\perp,em}$ at this stage. If the steps are short enough, accurate electron transport will be accomplished through the deflection of the direction vector, $\vec{\mathbf{u}}$. (For a discussion of the importance of lateral transport during an electron step caused by multiple scattering, see Lecture 16.)

[3] Although this guarantees that the direction vector remains normalized, a different procedure has been adopted for the EGS code. This normalization was applied for the electric field deflection only. For deflection by the magnetic field, only the components transverse to the magnetic field are normalized. Within the structure of the EGS code, this guarantees that the parallel component of $\vec{\mathbf{u}}$ does not change thereby imposing conservation of momentum in the direction parallel to the magnetic field.

both for the EGS code [RBMG84] and the ITS codes [Hal74, HSV75, HSV77]. We simply give examples of some benchmark calculations.

Figure 16.1: Electron trajectories influenced by a constant electric field in vacuum calculated using the EGS4 Monte Carlo code compared with an exact analytical solution.

First we present a test of transport in a vacuum in a constant electric field. In fig. 16.1 we compare electron trajectories calculated using the EGS Monte Carlo code and the exact analytical solution, eq. 17.6 discussed in sec. 16.1.1. In this example, the electric field is aligned with the ordinate with a field strength of 511 kV/cm. Three positron trajectories are depicted with energies 0.1, 1.0, and 10 MeV initially directed at a 45° angle against the electric field. We also show two negatron (e^-) trajectories with energies 2.0 and 20 MeV initially directed perpendicular to the electric field. The Monte Carlo calculated trajectories are depicted by the solid lines while the exact analytical solutions are shown with dotted lines. The electron step-size limits were obtained by letting the change in kinetic energy be at most two percent (the constraint expressed by eq. 16.35) and the deflection be at most 0.02 (the constraint expressed by eq. 16.37). We see only minor evidence that the neglect of lateral transport during the course of a step causes error. This error can be further reduced by shortening the step-sizes even more.

Next we present a test of transport in a vacuum in a constant magnetic field. In fig. 16.2 we compare 0.1 MeV negatron and 1.0 MeV positron trajectories calculated using the EGS Monte Carlo code and the exact analytical solution, eqs. 17.9 and 17.10 also discussed in

Figure 16.2: Electron trajectories influenced by a constant magnetic field in vacuum calculated using the EGS4 Monte Carlo code compared with exact analytical solutions.

sec. 16.1.1. In this case the magnetic field is aligned with the abscissa with a field strength of 0.17 Tesla. The initial velocities are directed 45° away from the $\vec{\mathbf{B}}$-axis. The analytic solutions are represented by the solid lines. The agreement is excellent in both cases.

The "vacuum" tests were designed to verify that the differential equations of electron transport in external fields are being correctly integrated during the course of the electron simulation. We have verified that the relatively crude "first-order" method presented in sec. 16.2 works acceptably well in vacuum. If more accuracy is desired, then one merely has to shorten the electron step-size. We may have been much more sophisticated if we were interested in rapid and accurate transport in vacuum. For constant, or nearly constant fields, we may have employed the exact analytic solutions expressed in sec. 16.1 into the Monte Carlo code directly. If the simulations take place in finite geometries, then one would have to find solutions of the intersection points of the electron trajectories with the surfaces enclosing the geometry in which the simulation takes place. This is a straightforward but tedious problem for most classes of bounding surfaces, for example, cones or cylinders.

For electron transport in spatially varying fields or simulations in tenuous media where the multiple scattering and retardation forces may be considered to be perturbations upon the deflections and energy losses due to the external fields, one may use more sophisticated integration techniques to obtain electron trajectories more quickly. For example, the ITS series of codes uses a fifth-order Runge-Kutta technique to integrate the external field transport equations [Hal74, HSV75, HSV77].

When one is interested in electron transport in dense media, unless more theoretical work is done to elucidate the complex interplay of multiple scattering, retardation and external field forces, one must resort to a method similar to that presented at the start of this section. We have verified that this method works well and now wish to perform benchmarks in dense media. Unfortunately, the authors are not aware of clear-cut benchmark experiments of this sort. Instead, we present three examples which verify at most semi-quantitatively, that external field transport in dense media is possible.

The first example is that of a 10 MeV negatron through 10 m of air with a magnetic field with strength 0.17 T aligned with the electron's initial direction of motion. An "end-on" view is shown in fig. 16.3. Because the initial direction and the magnetic field are aligned, there is no deflection by the magnetic field until the particle is deflected by the medium. The initial deflection, a creation of a low energy δ-ray in the left-hand side of the figure, caused the initial deflection in this example. Afterwards, the negatron spirals in the counter-clockwise direction. The electron transport cut-off energy in this example is 10 keV at which point the simulation of δ-ray transport is seen to come to an abrupt end. An interesting feature of this example is that a relatively high-energy δ-ray was produced spiraling many times before losing all its energy. The δ-ray actually drifted about 20 cm in the direction of the magnetic field.

Figure 16.3: An "end-on" view of a 10 MeV negatron being transported through air. The particle's initial direction and the magnetic field direction are normal to and out of the plane of view. The vertical axis is 10 cm long.

Figure 16.4: 10 MeV electrons incident on water containing a uniformly charged cylinder. The charge on the cylinder strongly focusses the electrons. The horizontal axis is 5 cm long.

A second example is a representation of electric field focusing of electrons. In this case, a uniformly charged cylinder with radius 0.5 cm, is placed 2.5 cm deep in a water target. The target is irradiated by a broad, parallel beam of 10 MeV negatrons. The electric potential is $\Phi(\rho)[\mathrm{MV}] = -\ln(2\rho)$, which is zero on the surface of the cylinder, and is sufficient to "bind" any electron which strikes the surface of the water. The variable ρ is the distance away from the cylinder's axis measured in centimeters. The electric field outside the water is zero as well as inside the cylinder. We note the strong focusing of the electrons although in a few cases, the multiple scattering turns particles away from the cylinder.

Finally, we present an example that provides some quantitative support for the viability of electron transport in dense media in the presence of strong electric fields. It is somewhat related to the previous example, having to do with focusing of electrons into a cylinder held at high potential [RBMG84]. Some explanation is required beforehand, however.

When electrons bombard insulating materials and come to rest as a result of slowing down in the medium, they become trapped and can set up large electric fields in the medium. Subsequent bombardments may be greatly perturbed by the presence of these fields, and this may result in a deleterious effect on the accuracy of dosimetry measurements in these insulating materials [RBMG84]. It was common practice to place a radiation detection device in a cylindrical hole drilled in a plastic phantom and subject it to electron bombardment. The air in the cylinder becomes ionized and conducting during bombardment and large amounts of positive charge may be induced to the cylinder to bring its surface to constant potential. Therefore, electrons are focussed to the cylinder causing one to measure an artificially high response in the detector. These effects were calculated by Monte Carlo methods [RBMG84] but the benchmark of the Monte Carlo code is not “clear-cut”. The electric field strength had to be estimated and it was further assumed that the electric field strength at the surface of the cylinder was proportional to the total amount of charge bombarding the insulating material.

Figure 16.5: Measured and calculated response enhancement due to the presence of large electric fields versus total electron dose responsible for the stored charge causing the focusing effect. The incident electron energy was 5.7 MeV and the 0.7 cm diameter cylinder was located 1.5 cm deep in a polymethylmethacrylate (PMMA) medium.

Given these approximations, a comparison of the calculated and measured enhancement of detector response is presented in fig. 16.5. In this comparison, the incident electron energy was 5.7 MeV and the cylinder in which the detector was placed was 0.7 cm in diameter placed at a depth of 1.5 cm in a polymethylmethacrylate (PMMA) medium. In this case, the maximum field strength exceeded 1MV/cm! We note that the experimental and calculated results agree to within the 1% accuracy of the simulations.

Bibliography

[Ber63] M. J. Berger. Monte Carlo Calculation of the penetration and diffusion of fast charged particles. *Methods in Comput. Phys.*, 1:135 – 215, 1963.

[Bet30] H. A. Bethe. Theory of passage of swift corpuscular rays through matter. *Ann. Physik*, 5:325, 1930.

[Bet32] H. A. Bethe. Scattering of electrons. *Z. für Physik*, 76:293, 1932.

[Blo33] F. Bloch. Stopping power of atoms with several electrons. *Z. für Physik*, 81:363, 1933.

[Hal74] J. A. Halbleib, Sr. . *J. Appl. Phys.*, 45:4103 –, 1974.

[HM84] J. A. Halbleib and T. A. Mehlhorn. ITS: The integrated TIGER series of coupled electron/photon Monte Carlo transport codes. *Sandia Report SAND84-0573*, 1984.

[HSV75] J. A. Halbleib, Sr., and W. H. Vandevender. Coupled electron/photon transport in static external magnetic fields. *IEEE Transactions on Nuclear Science*, NS-22:2356 – 2361, 1975.

[HSV77] J. A. Halbleib, Sr., and W. H. Vandevender. Coupled electron photon collisional transport in externally applied electromagnetic fields. *J. Appl. Phys.*, 48:2312 – 2319, 1977.

[Jac75] J. D. Jackson. *Classical Electrodynamics.* Wiley, New York, 1975.

[Mol47] G. Z. Molière. Theorie der Streuung schneller geladener Teilchen. I. Einzelstreuung am abgeschirmten Coulomb-Field. *Z. Naturforsch*, 2a:133 – 145, 1947.

[Mol48] G. Z. Molière. Theorie der Streuung schneller geladener Teilchen. II. Mehrfach- und Vielfachstreuung. *Z. Naturforsch*, 3a:78 – 97, 1948.

[NHR85] W. R. Nelson, H. Hirayama, and D. W. O. Rogers. The EGS4 Code System. Report SLAC–265, Stanford Linear Accelerator Center, Stanford, Calif, 1985.

[RBMG84] J. A. Rawlinson, A. F. Bielajew, P. Munro, and D. M. Galbraith. Theoretical and experimental investigation of dose enhancement due to charge storage in electron-irradiated phantoms. *Med. Phys.*, 11:814 – 821, 1984.

Problems

1.

Chapter 17

Variance reduction techniques

In this chapter we discuss various techniques which may be used to make calculations more efficient. In some cases, these techniques require that no further approximations be made to the transport physics. In other cases, the gains in computing speed come at the cost of computing results that may be less accurate since approximations are introduced. The techniques may be divided into 3 categories: those that concern electron transport only, those that concern photon transport only, and other more general methods. The set of techniques we discuss does not represent an exhaustive list. There is much reference material available and we only cite a few of them (refs. [Kah56], [HH64], [MI75], [Car81], [Lun81]). An especially rich source of references is McGrath's book [MI75], which contains an annotated bibliography. Instead we shall concentrate on techniques that have been of considerable use to the authors and their close colleagues. However, it is appropriate to discuss briefly what we are trying to accomplish by employing variance reduction techniques.

17.0.1 Variance reduction or efficiency increase?

What we really mean to do when we employ variance reduction techniques is to reduce the time it takes to calculate a result with a given variance. Analogue Monte Carlo calculations attempt to simulate the full stochastic development of the electromagnetic cascade. Hence, with the calculated result is associated an estimated variance, s^2. The method by which s^2 is estimated was discussed in Chapter 5 *Error estimation.* Let us assume that it is calculated by some consistent method as discussed previously. If the estimated variance is too large for our purposes we run more histories until our criterion is satisfied. How do we estimate how many more histories are needed? How do we know that the estimated variance is meaningful? Assuming we can do this, what do we do if it is too expensive to simulate the requisite number of histories? We may need a more subtle approach than reducing variance by "grinding out" more histories.

Let us say we devise some "tricks" that allow us to reduce the variance by, say, a factor of 10 using the same number of histories. Let's also imagine that this new subtle approach we have devised takes, say, 20 times longer on average to complete a particle history. (For example, our variance reduction technique may involve some detailed, expensive calculation executed every particle step.) Although we have reduced the variance by a factor of 10, we take 20 times longer to calculate each particle history. We have actually reduced the efficiency by a factor of two! To add to the insult, we have wasted our own time implementing a technique which reduces efficiency!

We require a measure that we may use to estimate gains in efficiency of a given "variance reduction" technique. It is common to use the efficiency, ϵ, defined by:

$$\epsilon = \frac{1}{s^2 T}, \tag{17.1}$$

where T is a measure of the computing time used (*e.g.* CPU seconds). The motivation for this choice comes from the following: We can safely assume that mean values of quantities calculated by Monte Carlo methods are distributed normally. This is a consequence of the the Central Limit Theorem[1] as discussed in Chapter 5 *Error estimation.* It follows then that for calculations performed using identical methods, the quantities s^2N and s^2T, where N is the number of histories, are constant, on average, as long as one is in the region of validity of the Central Limit Theorem. This is so because N should be directly proportional to T. By considering the efficiency to be constant, eq. 17.1 may be used to estimate the total computing time required to reach a given statistical accuracy if a preliminary result has already been obtained. For example, if one wishes to reduce the uncertainty, s, by a factor of 2, one needs 4 times as many histories. More importantly, eq. 17.1 allows us to make a quantitative estimate of the gain (or loss!) in efficiency resulting from the use of a given "variance reduction" technique since it accounts for not only the reduction in variance but also the increased computing time it may take to incorporate the technique. In the aforementioned example, using eq. 17.1 we would obtain ϵ(with subtlety)/ϵ(brute force) = 0.5, a *reduction* of 1/2. In the following sections we attempt to present more successful variance reduction techniques!

[1] Invoking the Central Limit Theorem requires that the population variance of a score or tally exists. While one can imagine score distributions where the population variance does not exist, they usually appear under contrived circumstances and rarely physical ones in radiation transport. If a population variance does not exist then one can always appeal to the *Strong Law of Large Numbers* [Fel67] which states that as long as a population mean exists, then repeated simulations should bring one closer to the true population mean. However, the way that this mean is approached may be much slower than that suggested by the Central Limit Theorem. If the population mean does not exist, then estimates of the sample mean are guaranteed not to be meaningful. An example of such a distribution would be

$$p(x)\mathrm{d}x = x^{-2}\mathrm{d}x \ , \ 1 \le x \le \infty \ , \tag{17.2}$$

that can be sampled straightforwardly but that does not have a mean!

A careful discussion of a score, especially one that has not been previously studied, ought to include a study of the underlying distribution or a crude but somehow representative analytic model that may be able to characterize the distribution.

17.1 Electron-specific methods

17.1.1 Geometry interrogation reduction

This section might also have been named "Code optimisation" or "Don't calculate what you don't really need", or something equivalent. We note that there is a fundamental difference between the transport of photons and electrons in a condensed-history transport code. Photons travel relatively long distances before interacting and their transport steps are often interrupted by boundary crossings (*i.e.* entering a new scoring region or element of the geometry). The transport of electrons is different, however. In addition to having its step interrupted by boundary crossings or the sampling of discrete interactions, the electron has other constraints on step-size. These constraints may have to do with ensuring that the underlying multiple scattering theories are not being violated in any way (See Chapter 14 *Electron step-size artefacts and PRESTA*), or the transport may have to be interrupted so that the equations of transport in an external electromagnetic field may be integrated [Bie89]. Therefore, it is often unnecessary to make repeated and expensive checks with the geometry routines of the transport code because the electron is being transported in an effectively infinite medium for most of the transport steps. The EGS4 code [NHR85], has an option that allows the user to avoid these redundant geometry subroutine calls. With this option switched on, whenever the geometry must be checked for whatever reason, the closest distance to any boundary is calculated and stored. This variable is then decremented by the length of each transport step. If this variable is greater than zero, the electron can not be close enough to a boundary to cross it and the geometry subroutines are not interrogated. If this variable drops to zero or less, the geometry subroutines are called because a boundary crossing may occur.

By way of example, consider the transport of an electron between two boundaries as depicted in figure 17.1. For the sake of argument, each transport step (between either the **x**'s and **o**'s has a distance $\sqrt{2}$ and the transport trajectory is directly diagonal between the two boundaries. For simplicity, we neglect scattering in this example. When the geometry routines are interrogated, the start of the step is delineated by an **x**. When the geometry routines can be skipped, the start of the step is delineated by an **o**.

This is the way the algorithm works. The closest distance to any boundary is usually calculated in EGS4 by `SUBROUTINE HOWFAR`. This distance is called `DNEAR` and it is calculated before the particle is transported. (In EGS4/PRESTA, it is calculated before the call to `SUBROUTINE HOWFAR`.) After transport by distance `VSTEP` in the case of electrons or `USTEP` in the case of photons, `DNEAR` is decremented by `VSTEP` or `USTEP`. On the subsequent step, if the next `USTEP` is less than `DNEAR` then is is known that the step will not cause the particle to escape the current region and so the call to `SUBROUTINE HOWFAR` is avoided. (The EGS4/PRESTA scheme is slightly more efficient since it employs a more current version of `DNEAR`. However, it also calculates `DNEAR` for every electron step.)

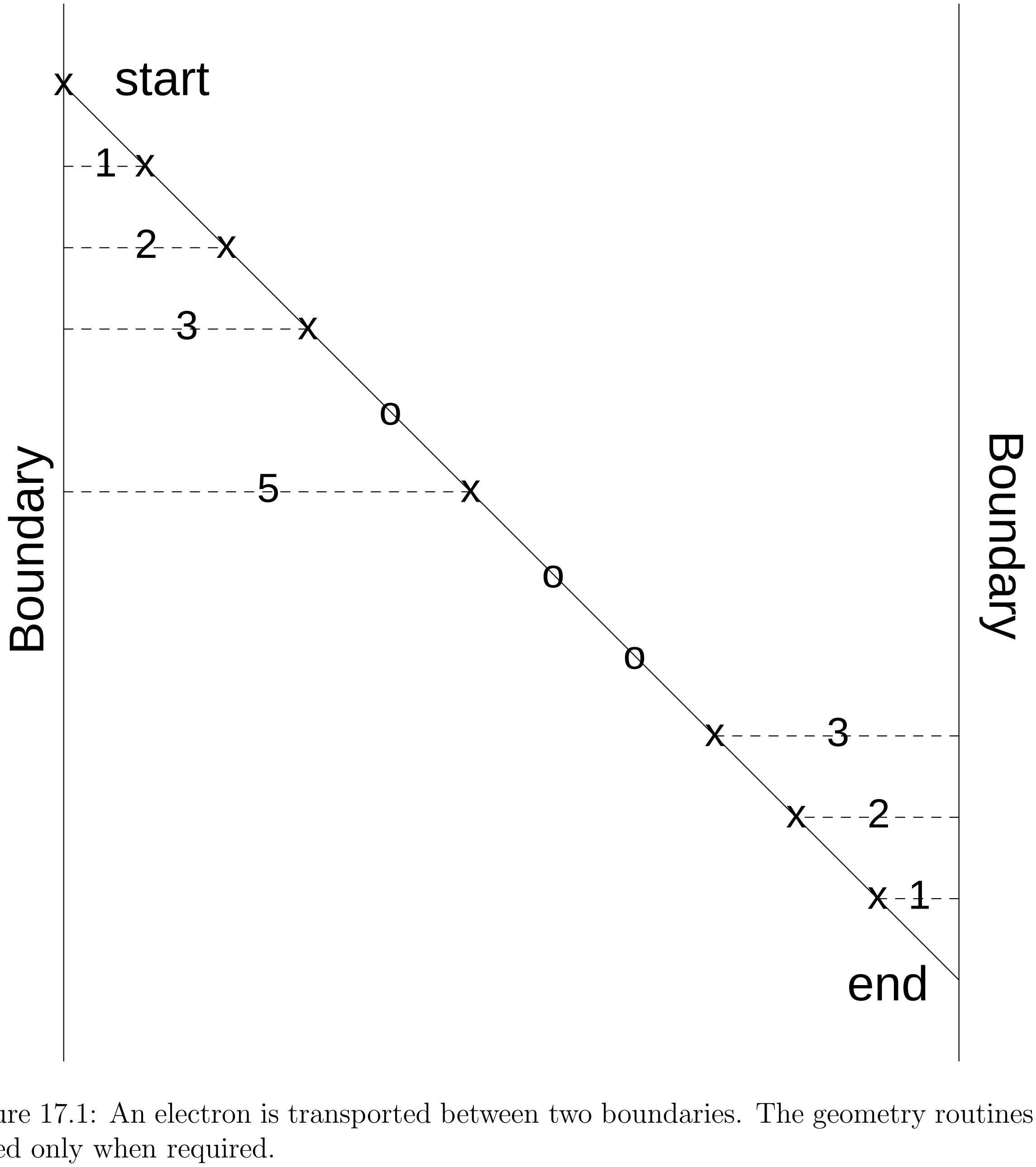

Figure 17.1: An electron is transported between two boundaries. The geometry routines are called only when required.

There is no approximation involved in this technique. The gain in transport efficiency is slightly offset by the extra calculation time that is spent calculating the distance to the closest boundary. (This parameter is not always needed for other aspects of the particle transport.) As an example, consider the case of a pencil beam of 1 MeV electrons incident normally on a 0.3 cm slab of carbon divided into twelve 0.025 cm slabs. For this set of simulations, transport and secondary particle creation thresholds were set at 10 keV kinetic energy and we used EGS4 [NHR85] setting the energy loss per electron step at 1% for accurate electron transport [Rog84b] at low energies. The case that interrogates the geometry routines on every step is called the "base case". We invoke the trick of interrogating the geometry routines only when needed and call this the "RIG" (reduced interrogation of geometry) case. The efficiency ratio, $\epsilon(\mathrm{RIG})/\epsilon(\mathrm{base})$, was found to be 1.34, a significant improvement. (This was done by calculating DNEAR in the HOWFAR routine of a planar geometry code. A discussion of DNEAR is given on pages 256–258 of the EGS4 manual [NHR85].)

Strictly speaking, this technique may be used for photons as well. For most practical problems, however, the mean free path for the photons in the problem is of the order, or greater than the distance between boundaries. For deep penetration problems or similar problems, this may not be true. However, this technique is usually more effective at speeding up the electron transport part of the simulation.

The extra time required to calculate the distance to the closest boundary may be considerable, especially for simulations involving curved surfaces. If this is so then the efficiency gain may be much less or efficiency may be lost. It is advisable to test this technique before employing it in "production" runs.

17.1.2 Discard within a zone

In the previous example, we may be just interested in the energy deposited in the planar zones of the carbon slab. We may, therefore, deposit the energy of an electron entirely within a zone if that electron's range is less than the distance to any bounding surface of the zone in which it is being transported. A depiction of this process can be seen in figure 17.2. We note that we make an approximation in doing this—we neglect the creation and transport of any bremsstrahlung γ's that may otherwise created. For the worst possible case in this particular example, we will be discarding electrons that have a range that is half of the zone thickness, *i.e.* having a kinetic energy of about 110 keV. The radiative yield of these electrons is only about 0.07%. Therefore, unless we are directly interested in the radiative component of the electron's slowing down process in this problem, the approximation is an excellent one. For the above example, we realise a gain in the efficiency ratio, $\epsilon(\text{zonal discard} + \mathrm{RIG})/\epsilon(\mathrm{base})$, of about 2.3. In this case, the transport cut-off, below which no electron was transported, was 10 keV. If we had used a higher cut-off the efficiency gain would have been less.

Before adopting this technique, the user should carefully analyze the consequences of the approximation—the neglect of bremsstrahlung from the low energy electron component.

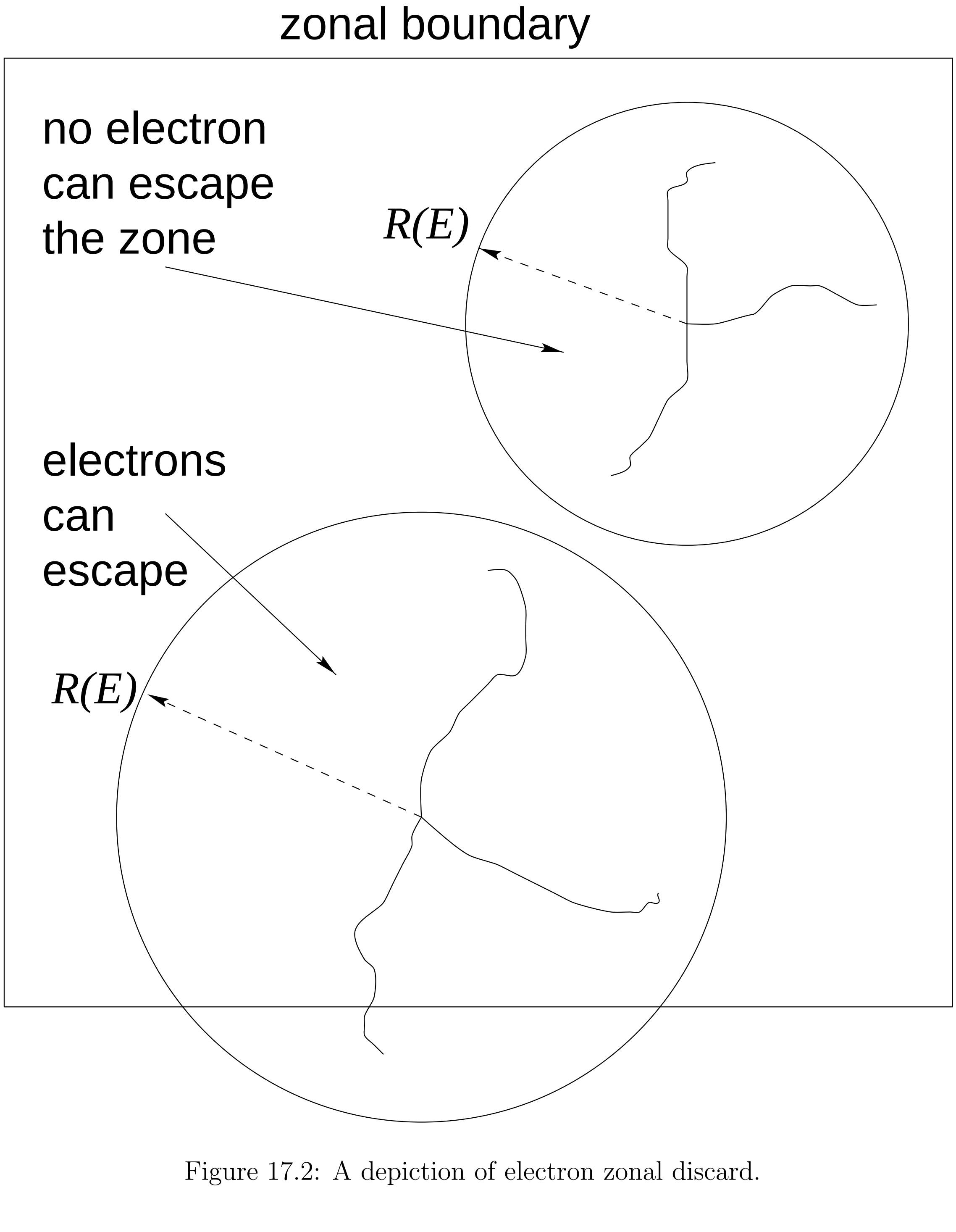

Figure 17.2: A depiction of electron zonal discard.

17.1.3 PRESTA!

In the previous Chapter 14, *Electron step-size dependencies and PRESTA*, we discussed an alternative electron transport algorithm, PRESTA. This algorithm, by making improvements to the physical modeling of electron transport, allows the use of large electron steps when one is far away from boundaries. This algorithm may, therefore, be considered to be a variance reduction technique, since it saves computing time by employing small steps only where needed—in the vicinity of boundaries and interfaces, as depicted in figure 17.3. Continuing with the present example, we calculate the gain in efficiency ratio, $\epsilon(\mathrm{PRESTA})/\epsilon(\mathrm{base})$, to be 6.1. RIG is always switched on with PRESTA, so it is actually fairer to calculate the efficiency ratio, $\epsilon(\mathrm{PRESTA})/\epsilon(\mathrm{RIG})$, which was found to be 4.6. If we allow zonal discard as well, we calculate the efficiency ratio, $\epsilon(\text{zonal discard} + \mathrm{PRESTA})/\epsilon(\text{zonal discard} + \mathrm{RIG})$, to be 3.1. There is a brief discussion in the previous chapter on when PRESTA is expected to run quickly. Basically, the fewer the boundaries and the higher the transport cutoffs, the faster PRESTA runs. A detailed discussion is given in the PRESTA documentation [BR87].

17.1.4 Range rejection

As a final example of electron variance reduction, we consider the technique called "range rejection". This is similar to the "discard within a zone" except for a few differences. Instead of discarding (*i.e.* stopping the transport and depositing the energy "on the spot") the electron because it can not reach the boundaries of the geometrical element it is in, the electron is discarded because it can not reach some region of interest. This is depicted in figure 17.4. For example, a particle detector may contain a sensitive volume where one wishes to calculate energy deposit, or some other quantity. Surrounding this sensitive volume may be shields, converters, walls *etc.* where one wishes accurate particle transport to be accomplished but where one does not wish to score quantities directly. Electrons that can not reach the sensitive volume may be discarded "on the spot", providing that the neglect of the bremsstrahlung γ's causes no great inaccuracy.

As an example of range rejection, we consider the case of an ion chamber [BRN85]. In this case, a cylindrical air cavity, 2 mm in depth and 1.0 cm in radius is surrounded by 0.5 g/cm^2 carbon walls. A flat circular end is irradiated by 1.25 MeV γ-rays incident normally. This approximates the irradiation from a distant source of ^{60}Co. This is a "thick-walled" ion chamber, so-called because it's thickness exceeds the range of the maximum energy electron that can be set in motion by the incident photons. This sets up a condition of "near charged particle equilibrium" in the vicinity of the cavity. The potential for significant saving in computer time is evident, for many electrons could never reach the cavity. We are interested in calculating the energy deposited to the air in the cavity and we are not concerned with scoring any quantities in the walls. The range rejection technique involved calculating the closest distance to the surface of the cavity on every transport step. If this distance exceeded the CSDA range of the electron, it was discarded. The omission of residual

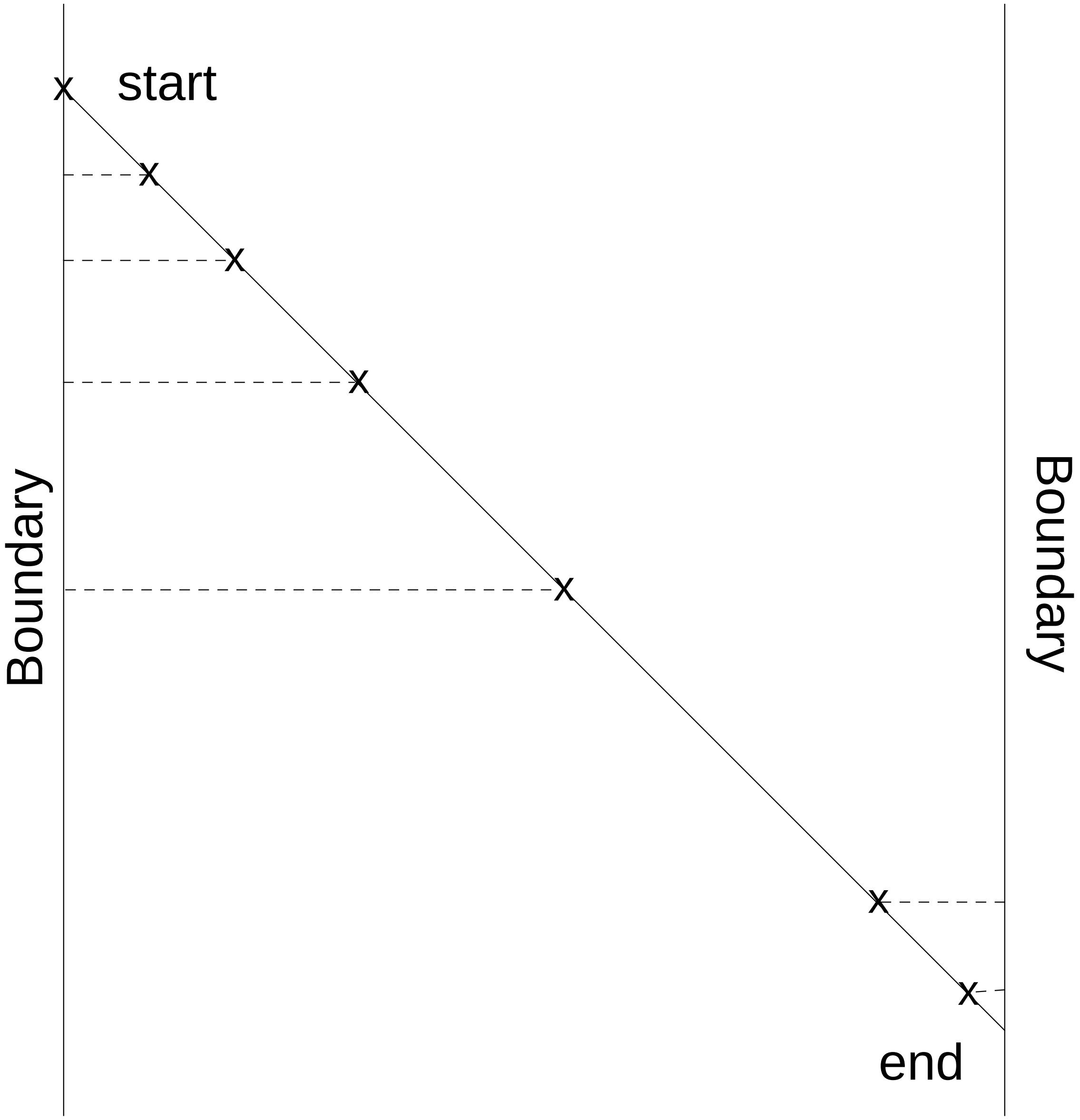

Figure 17.3: A depiction of PRESTA-like transport for the diagonal trajectory discussed previously.

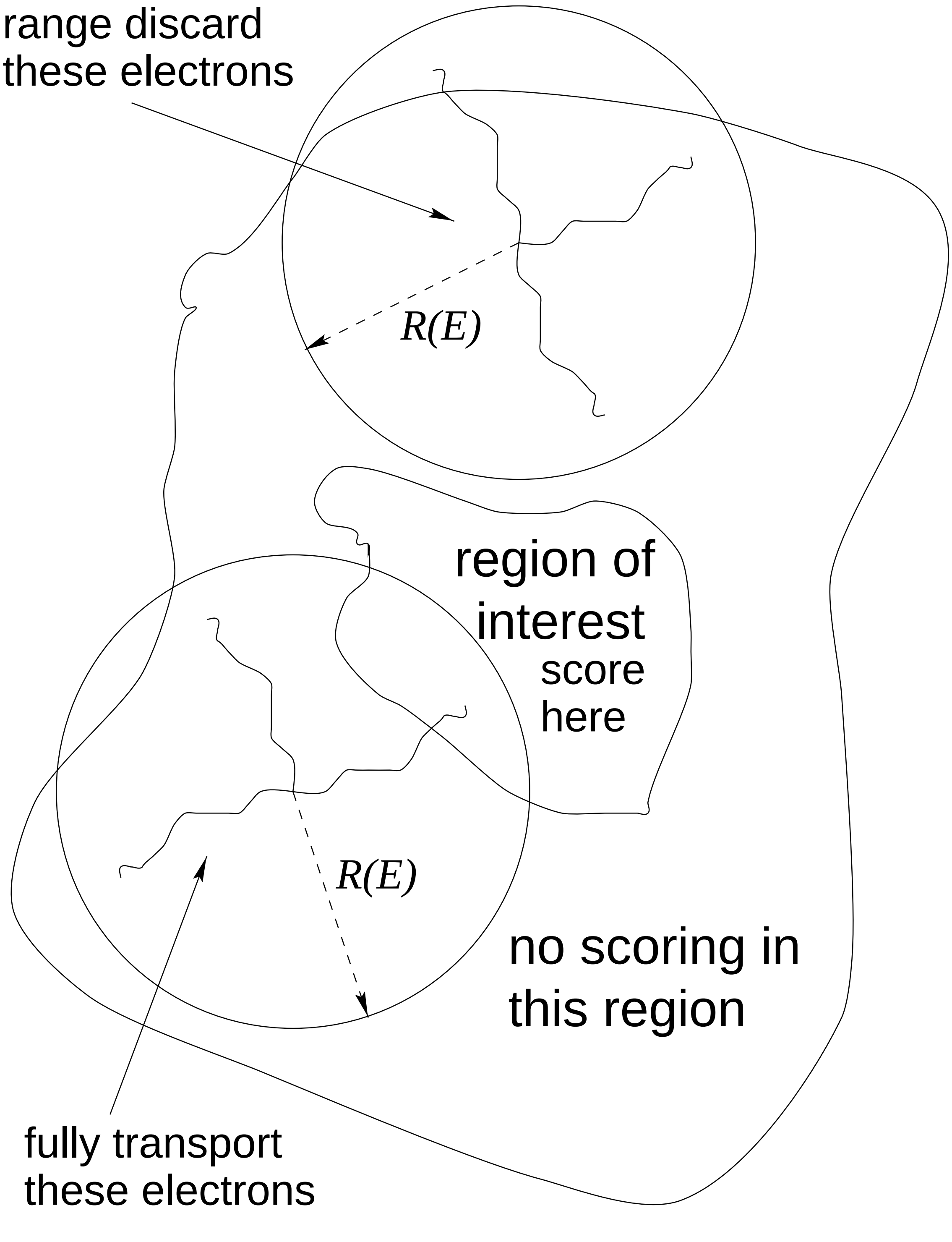

Figure 17.4: A depiction of range rejection.

bremsstrahlung photon creation and transport was negligible in this problem. The secondary particle creation thresholds were set at 10 keV kinetic energy as well as the transport cut-off energies. (ECUT=AE=0.521 MeV, PCUT=AP=0.01 MeV, and ESTEPE=0.01 for accurate low energy simulation.) A factor of 4 increase in efficiency was realised in this case.

Range rejection is a relatively crude but effective method. The version described above neglects residual bremsstrahlung and is applicable when the discard occurs in one medium. The bremsstrahlung problem could be solved by forcing at least some of the electrons to produce bremsstrahlung. The amount of energy eventually deposited from these photons would have to be weighted accordingly to keep the sampling game "fair". Alternatively, one could transport fully a fraction, say f, of the electrons and weight any resultant bremsstrahlung photons by $1/f$. The other problem, the one of multi-media discard, is difficult to treat in complete generality. The difficulty is primarily a geometrical one. The shortest distance to the scoring region is the shortest geometrical path only when the transport can occur in one medium. The shortest distance we need to calculate for range rejection is the path along which the energy loss is a minimum. It is not difficult to imagine that finding the "shortest" path for transport in more than one medium may be very difficult. For special cases this may be done or approximations may be made. The "payoff" is worth it as large gains in efficiency may be realised, as seen in the above example.

17.2 Photon-specific methods

17.2.1 Interaction forcing

In problems where the interaction of photons is of interest, efficiency may be lost because photons leave the geometry of the simulation without interacting. This is depicted in figure 17.5 where an optically thin region (one for which μt is small, where t is the thickness of the slab and μ is the interaction coefficient) allows many photons to penetrate and escape the region. Efficiency is lost because time is spent tracking photons through a geometry and they do not contribute to the score. This problem has a simple and elegant solution.

The probability distribution for a photon interaction is:

$$p(\lambda)d\lambda = e^{-\lambda}d\lambda, \tag{17.3}$$

where $0 \leq \lambda < \infty$ and λ is the distance measured in mean free paths. It can easily be shown that sampling λ from this distribution can be accomplished by the following formula[2]:

$$\lambda = -\ln(1-\xi), \tag{17.4}$$

[2] It is conventional to use the expression, $\lambda = -\ln(\xi)$, since both $1-\xi$ and ξ are distributed uniformly on (0,1) but the former expression executes more slowly. However, it has a closer connection to the following mathematical development.

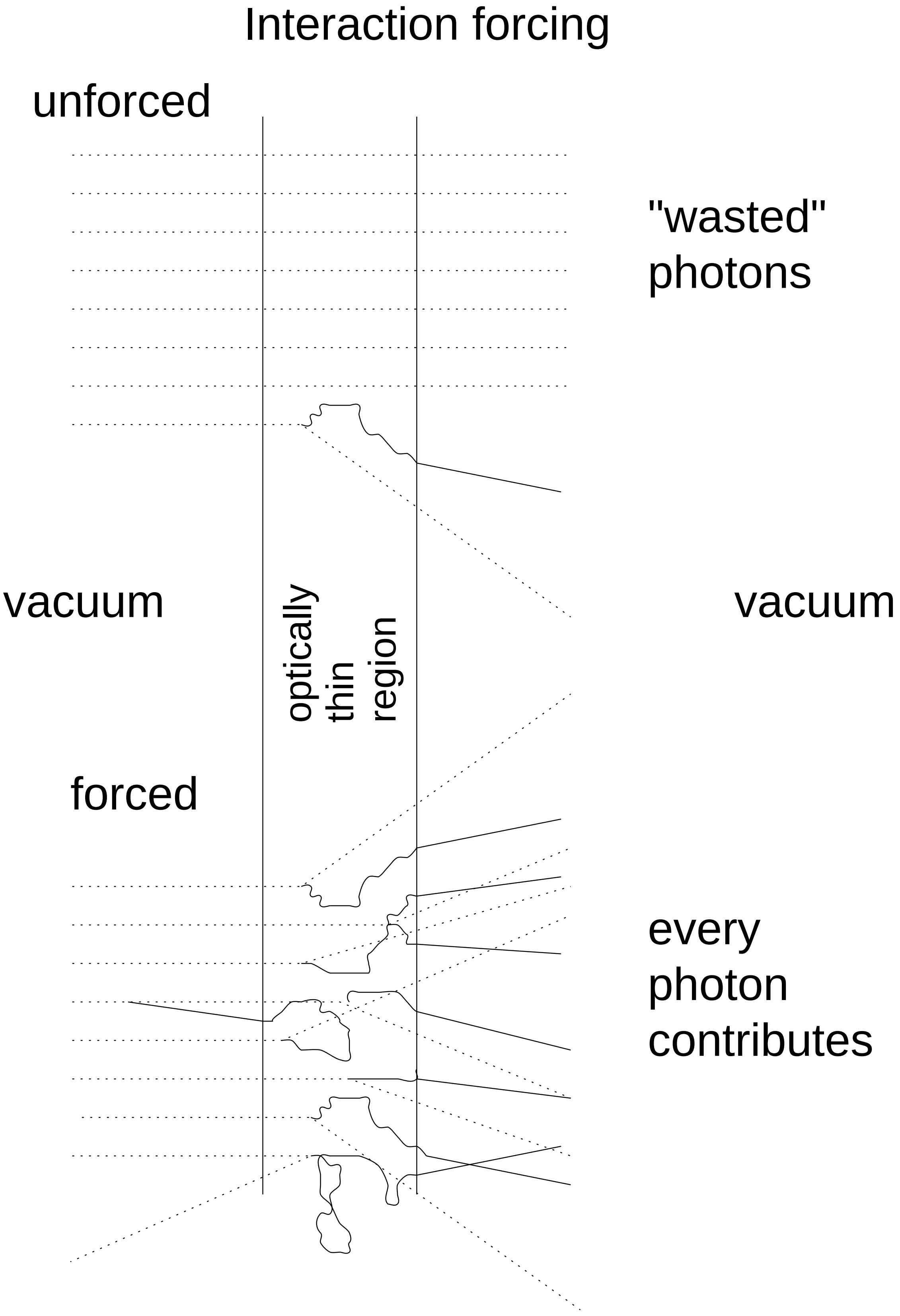

Figure 17.5: A depiction of photon interaction forcing.

where ξ is a random number uniform on the range, $0 \leq \xi < 1$. Since λ extends to infinity and the number of photon mean free paths across the geometry in any practical problem is finite, there is a non-zero and often large probability that photons leave the geometry of interest without interacting. If they don't interact, we waste computing time tracking these photons through the geometry.

Fortunately, this waste may be prevented. We can *force* these photons to interact. The method by which this can be achieved is remarkably simple. We construct the probability distribution,

$$p(\lambda)d\lambda = \frac{e^{-\lambda}d\lambda}{\int_0^\Lambda e^{-\lambda'}d\lambda'}, \tag{17.5}$$

where Λ is the total number of mean free paths along the direction of motion of the photon to the end of the geometry. (The geometry may be arbitrary.) This λ is restricted to the range, $0 \leq \lambda < \Lambda$, and λ is selected from the equation,

$$\lambda = -\ln(1 - \xi(1 - e^{-\Lambda})). \tag{17.6}$$

We see from eq. 17.6 that we recover eq. 17.4 in the limit $\Lambda \longrightarrow \infty$. Since we have forced the photon to interact within the geometry of the simulation we must *weight* the quantities scored resulting from this interaction. This weighting takes the form,

$$\omega' = \omega(1 - e^{-\Lambda}), \tag{17.7}$$

where ω' is the new "weighting" factor and ω is the old weighting factor. When interaction forcing is used, the weighting factor, $1 - e^{-\Lambda}$, simply multiplies the old one. This factor is the probability that the photon would have interacted before leaving the geometry of the simulation. This variance reduction technique may be used repeatedly to force the interaction of succeeding generations of scattered photons. It may also be used in conjunction with other variance reduction techniques. Interaction forcing may also be used in electron problems to force the interaction of bremsstrahlung photons.

On first inspection, one might be tempted to think that the calculation of Λ may be difficult in general. Indeed, this calculation is quite difficult and involves summing the contributions to Λ along the photon's direction through all the geometrical elements and materials along the way. Fortunately, most of this calculation is present in any Monte Carlo code because it must possess the capability of transporting the photons through this geometry! This interaction forcing capability can be included in the EGS code in a completely general, *geometry independent fashion* with only about 30 lines of code [RB84]!

The increase in efficiency can be dramatic if one forces the photons to interact. For example, for ion chamber calculations similar to those described in sec. 17.1.4 and discussed in detail elsewhere [BRN85], the efficiency improved by the factor 2.3. In this calculation, only about 6% of the photons would have interacted in the chamber. In calculating the dose to skin from contaminant electrons arising from the interaction of ^{60}Co (*i.e.* 1.25 MeV γ's) in 100 cm of air [RB84], the calculation executed 7 times more efficiently after forcing the photons to interact. In calculating the dose from ^{60}Co directly in the skin (a 0.001 cm slice of tissue) where normally only 6×10^{-5} of the photons interact, the efficiency improved by a factor of 2600 [RB84, RB85]!

17.2.2 Exponential transform, russian roulette, and particle splitting

The exponential transform is a variance reduction technique designed to enhance efficiency for either deep penetration problems (*e.g.* shielding calculations) or surface problems (*e.g.* build-up in photon beams). It is often used in neutron Monte Carlo work and is directly applicable to photons as well.

Consider the simple problem where we are interested in the surface or deep penetration in a simple slab geometry with the planes of the geometry normal to the z-axis. We then scale the interaction probability making use of the following formula:

$$\tilde{\lambda} = \lambda(1 - C\mu), \tag{17.8}$$

where λ is the distance measured in the number of mean free path's, $\tilde{\lambda}$ is the scaled distance, μ is the cosine of the angle the photon makes with the z-axis, and C is a parameter that adjusts the magnitude of the scaling. The interaction probability distribution is:

$$\tilde{p}(\lambda)d\lambda = (1 - C\mu)e^{-\lambda(1-C\mu)}d\lambda, \tag{17.9}$$

where the overall multiplier $1 - C\mu$ is introduced to ensure that the probability is correctly normalised, *i.e.* $\int_0^\infty \tilde{p}(\lambda)d\lambda = 1$. For $C = 0$, we have the unbiased probability distribution $e^{-\lambda}d\lambda$. One sees that for $0 < C < 1$, the average distance to an interaction is stretched[3]. For $C < 0$, the average distance to the next interaction is shortened. Examples of a stretched and shortened distribution are given in fig. 17.6. In order to play the game fairly, we must obtain the appropriate weighting function to apply to all subsequent scoring functions. This is obtained by requiring that the overall probability be unchanged. That is, we require:

$$\omega'\tilde{p}(\lambda)d\lambda = \omega p(\lambda)d\lambda, \tag{17.10}$$

where ω' is the new weighting factor and ω is the old weighting factor. Solving eq. 17.10 for ω' yields,

$$\omega' = \omega e^{-\lambda C\mu}/(1 - C\mu). \tag{17.11}$$

Finally, we require a technique to sample the stretched or shortened number of mean free paths to the next interaction point from a random number. It is easily shown that λ is selected using the formula:

$$\lambda = -\ln(\xi)/(1 - C\mu), \tag{17.12}$$

[3] Note that the average number of mean free paths to an interaction, $\langle\lambda\rangle$, is given by $\langle\lambda\rangle = \int_0^\infty \lambda\tilde{p}(\lambda)d\lambda = \frac{1}{1-C\mu}$.

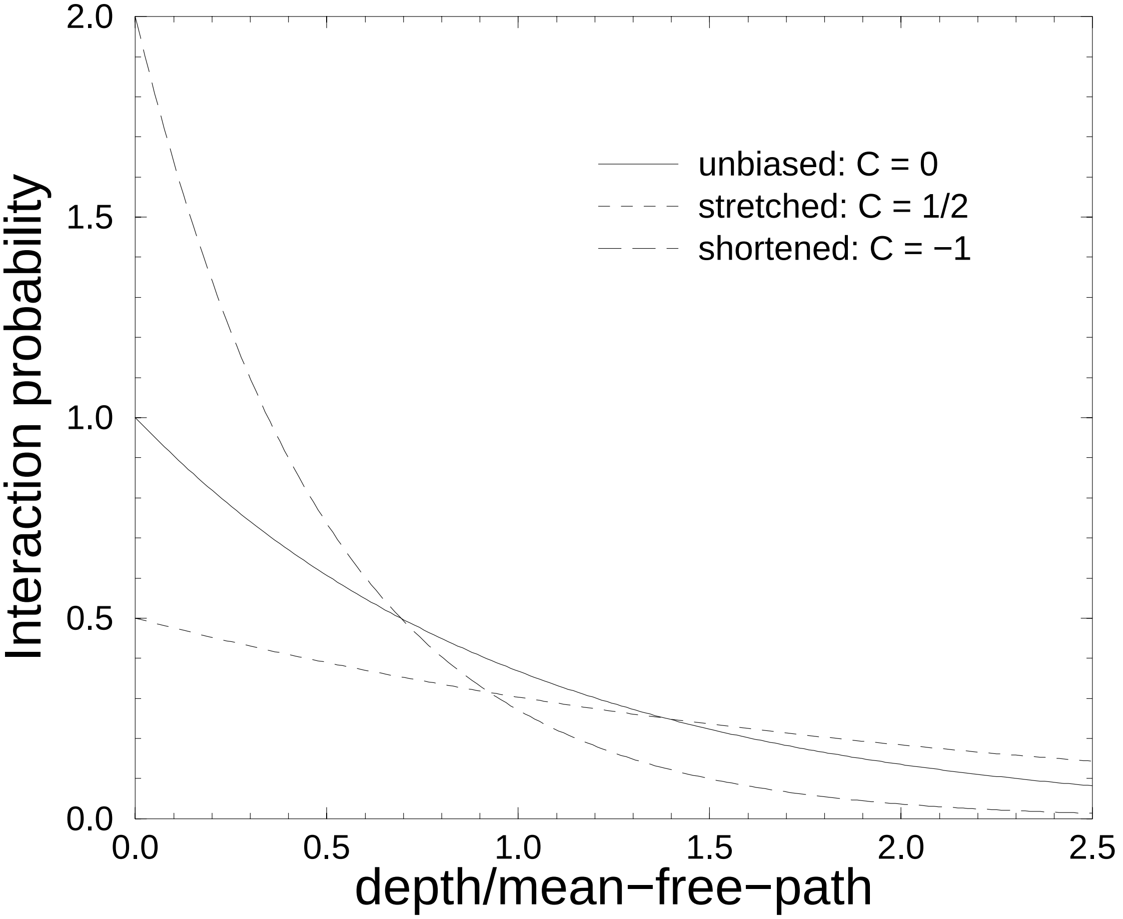

Figure 17.6: Examples of a stretched ($C = 1/2$) and shortened ($C = -1$) distribution compared to an unbiased one ($C = 0$). In all three cases, $\mu = 1$. For all three curves $\int_0^\infty \tilde{p}(\lambda) d\lambda$ is unity. The horizontal axis is in units of the number of mean free path's (mfp's).

where ξ is a random number chosen uniformly over the range, $0 < \xi \leq 1$.

For complete generality, one must obey the restriction, $|C| < 1$ since the photon's direction is arbitrary $(-1 \leq \mu \leq 1)$. "Path-length stretching" means that $0 < C < 1$, *i.e.* photons are made to penetrate deeper. "Path-length shortening" means that $-1 < C < 0$, *i.e.* photons are made to interact closer to the surface. For studies of surface regions, one may use a stronger biasing, *i.e.* $C \leq -1$. If one used $C \leq -1$ indiscriminately, then nonsense would result for particles going in the backward direction, *i.e.* $\mu < 0$. Sampled distances and weighting factors become negative. It is possible to use $C \leq -1$ for special, but important cases. (As we shall see in the next section, it is possible to remove all restrictions on C in finite geometries by combining exponential transforms and interaction forcing.) If one restricts the biasing to the incident photons which are directed along the axis of interest (*i.e.* $\mu > 0$) then $C \leq -1$ may be used. If one uses this severe biasing, then as seen in eq. 17.11, weighting factors for the occasional photon that penetrates very deeply can get very large. If this photon backscatters and interacts in the surface region where one is interested in gaining efficiency, the calculated variance can be undesirably increased. It is advisable to use a "splitting" technique [Kah56], dividing these large weight particles into a N smaller ones each with a new weight, $\omega' = \omega/N$ if they threaten to enter the region of interest. Thresholds for activating this splitting technique and splitting fractions are difficult to specify and choosing them is largely a matter of experience with a given type of application. The same comment applies when particle weights become vary small. If this happens and the photon is headed away from the region of interest it is advisable to play "russian roulette" [Kah56]. This technique works as follows: Select a random number. If this random number lies above a threshold, say α, the photon is discarded without scoring any quantity of interest. If the random number turns out to be below α the photon is allowed to "survive" but with a new weight, $\omega' = \omega/\alpha$, insuring the fairness of the Monte Carlo "game". This technique of "weight windowing" is recommended for use with the exponential transform [HB85] to save computing time and to avoid the unwanted increase in variance associated with large weight particles.

Russian roulette and splitting[4] can be used in conjunction with exponential transform, but they enjoy much use by themselves in applications where the region of interest of a given application comprises only a fraction of the geometry of the simulation. Photons are "split" as they approach a region of interest and made to play "russian roulette" as they recede. The three techniques, exponential transform, russian roulette and particle splitting are part of the "black art" of Monte Carlo. It is difficult to specify more than the most general guidelines on when they would be expected to work well. One should test them before employing them in large scale production runs.

Finally, we conclude this section with an example of severe exponential transform biasing with the aim to improve surface dose in the calculation of a photon depth dose curve [RB84].

[4] According to Kahn [Kah56], both the ideas and terminology for russian roulette and splitting are attributable to J. von Neumann and S. Ulam.

In this case, 7 MeV γ's were incident normally on a 30 cm slab of water. The results are summarised in Table 17.1. In each case the computing time was the same. Therefore, the relative efficiency reflects the relative values of $1/s^2$. As C decreases, the calculational efficiency for scoring dose at the surface increases while, in general, it decreases for the largest depth bin. The efficiency was defined to be unity for $C = 0$ at the for each bin. For the deepest bin there is an increase initially because the mean free path is 39 cm. At first the number of interactions in the 10 cm–30 cm bin increases! Note that as C is deceased the number of histories per given amount of computing time decreases. This is because more electrons are being set it motion, primarily at the surface. These electrons have smaller weights, however, to make the "game" fair.

Table 17.1: This series of calculations examines a case where a gain in the computational efficiency at the surface is desired. Each calculation took the same amount of computing time. In general, efficiency at the surface increases with decreased C while efficiency worsens at depth.

C	Histories	Relative efficiency on calculated dose		
	10^3	0–0.25 cm	6.0–7.0 cm	10–30 cm
0	100	≡1	≡1	≡1
-1	70	1	1.0	3.5
-3	55	1.5	1.2	0.6
-6	50	3.5	2.8	0.1

17.2.3 Exponential transform with interaction forcing

If the geometry in which the transport takes place is finite in extent, one may eliminate restrictions on the biasing parameter, C, by combining exponential transform with interaction forcing. By using the results of the previous two sections we find the interaction probability distribution to be:

$$p(\lambda)d\lambda = \frac{(1-C\mu)e^{-\lambda(1-C\mu)}}{1-e^{-\Lambda(1-C\mu)}}d\lambda. \tag{17.13}$$

The new weighting factor is:

$$\omega' = \omega\frac{(1-e^{-\Lambda(1-C\mu)})e^{-\lambda C\mu}}{1-C\mu}, \tag{17.14}$$

and the number of mean free paths is selected according to:

$$\lambda = -\frac{\ln(1-\xi(1-e^{-\Lambda(1-C\mu)}))}{1-C\mu}, \tag{17.15}$$

where ξ is a random number chosen uniformly over the range, $0 < \xi \leq 1$.

In the case $C \to 0$, eqs. 17.13–17.15 reduce to the equations of simple interaction forcing given in sec. 17.2.1. In the case $\Lambda \to \infty$, eqs. 17.13–17.15 reduce to the equations of exponential transform given in the previous section. However, the equations of this section permit any value of C to be used irrespective of the photon's direction as long as the geometry is finite, *i.e.* $0 < \Lambda < \infty$. In particular, the strong surface biasing, $C < -1$ need not be restricted to forward directed photons ($\mu > 0$), and penetration problems may use $C > 1$. This latter choice actually causes the interaction probability to *increase* with depth for forward directed photons! Again, as in the previous section, the same comments about particle splitting, russian roulette, and weight windowing apply.

17.3 General methods

17.3.1 Secondary particle enhancement

In some applications, one wishes to study the behaviour of secondary particles in an energy regime where they are highly suppressed. For example, X-rays from diagnostic X-ray tubes arise from bremsstrahlung radiation. The bremsstrahlung cross section is much smaller than the Møller cross section in the diagnostic regime (≈70 keV). So, calculating the bremsstrahlung characteristics by Monte Carlo method can be difficult since most of the effort is spent creating knock-on electrons. Another example would be the calculation of the effect of pair production in low-Z materials in the radiotherapy regime, below 50 MeV.

One approach is to enhance the number of these secondary particles by creating many of them, say N, once an interaction takes place and then giving them all a weight of $1/N$ to keep the game "fair". Once the interaction occurs, the secondary energy and directional probabilities can be sampled to produce distributions in energy and angle of the secondary particles emanating from a single interaction point. This method is more sophisticated than "splitting" where N *identical* particles are produced.

It is important that the stochastic nature of the primary particle be preserved. For this reason, the energy deducted from the primary particle is *not* the average of the secondary particles produced. The proper "straggling" is guaranteed by subtracting the entire energy of *one* of the secondary particles. This has the minor disadvantage that energy conservation is violated for the incident particle history that produces the "spray" of secondaries. However, over many histories and many interactions, energy conservation is preserved in an average sense.

The details of the implementation this method for the bremsstrahlung interaction in the EGS4 code is documented elsewhere [BMC89]. Examples of the use of this method in the radiotherapy regime [FRR90] and the diagnostic regime [NNS$^+$90] have been published.

17.3.2 Sectioned problems, use of pre-computed results

One approach to saving computer time is to split the problem into separate, manageable parts using the results of a previous Monte Carlo simulations as part of another simulation. These applications tend to be very specialised and unique problems demand unique approaches. For illustration, we shall present two related examples.

Fluence to dose conversion factors for monoenergetic, infinitely broad electron and photon beams incident normally on semi-infinite slabs of tissue and water have been calculated previously [RB85, Rog84a]. These factors, called $K_E(z)$, vary with depth, z, and on the energy of the photon beam, E, at the water surface. Dose due to an arbitrary incident spectrum as a function of depth, $D(z)$, is calculated from the following relation:

$$D(z) = \int_{E_{\mathrm{min}}}^{E_{\mathrm{max}}} \Phi(E) K_E(z) dE, \tag{17.16}$$

where $\Phi(E)$ is the electron or photon fluence spectrum and it is non-zero between the limits of E_{min} and E_{max}. Each K_E array represents a long calculation. If one uses these pre-calculated factors, one can expect orders of magnitude gains in efficiency. If one is interested in normally incident broad beams only, the calculated results should be quite accurate. The only approximations arise from the numerical integration represented by eq. 17.16 and associated interpolation errors. However, there are two important assumptions buried in the K_E's—the incident beams are *broad* and incident *normally.* For photons, using narrow beams in this method can cause 10% to 50% overestimates of the peak dose. For narrow electron beams this method is not recommended at all.

Another example is the study of the effects of scatter in a ^{60}Co therapy unit [REBvD88]. For the purpose of modeling the therapy unit in a reasonable amount of computing time, it was divided into two parts. First, the source capsule itself was modeled accurately and the phase space parameters (energy, direction, position) of those particles leaving the source capsule and entering the collimator system were stored. About 2×10^6 particles were stored in this fashion taking about 24 hrs of VAX 11/780 CPU time for executing the simulation. This data was then used repeatedly in modeling the transport of particles through the collimators and filters of the therapy head. The approximation inherent in this stage of the calculation is the interaction between the source capsule and the rest of the therapy head. However, since the capsule is small with respect to the therapy head and we are interested in calculating the effects of the radiation somewhat downstream from the therapy head, the approximation is an excellent one. Another aspect of this calculation was that the effect of the contaminant electrons downstream from the therapy head was studied. Again, this part of the calculation was "split off" and done by the method described previously. That is, eq. 17.16 was used to calculate the depth-dose profiles in tissue.

By splitting the problem into 3 parts, the total amount of CPU time used to simulate the ^{60}Co therapy head [REBvD88] required 5–16 hours of CPU time for each geometry. If we had attempted to simulate the problem entirely without "dividing and conquering", the amount of CPU time required would have been prohibitive.

17.3.3 Geometry equivalence theorem

A special but important subset of Monte Carlo calculations is normal beam incidence on semi-infinite geometries, with or without infinite planar inhomogeneities. The use of a simple theorem, called the “geometry equivalence” or “reciprocity” theorem, provides an elegant technique for calculating some results more quickly. First we prove the theorem.

Imagine that we have a zero radius beam coincident with the z-axis impinging on the geometry described above. We “measure” a response that must have the form $f(z,|\boldsymbol{\rho}|)$, where $\boldsymbol{\rho}$ is the cylindrical radius. This functional form holds true since there is no preferred azimuthal direction in the problem. If the beam is now shifted off the axis by an amount $\boldsymbol{\rho'}$, then the new functional form of the response must have the form, $f(z,|\boldsymbol{\rho}-\boldsymbol{\rho'}|)$, by translational symmetry. Finally, consider that we have a finite circular beam of radius $\rho_{\rm b}$ and we wish to integrate the response over a finite-size detection region with circular radius $\rho_{\rm d}$. This integrated response has the form,

$$F(z,\rho_{\rm b},\rho_{\rm d}) = \int^{|\boldsymbol{\rho'}|\le\rho_{\rm b}} d\rho' \int^{|\boldsymbol{\rho}|\le\rho_{\rm d}} d\rho \; f(z,|\boldsymbol{\rho}-\boldsymbol{\rho'}|), \qquad (17.17)$$

where $\int^{|\boldsymbol{\rho}|\le\rho_{\rm d}} d\rho$ is shorthand for $\int_0^{2\pi} d\phi \int_0^{\rho_{\rm d}} d\rho$. If we exchange integration indices in eq. 17.17, then we obtain the reciprocity relationship,

$$F(z,\rho_{\rm b},\rho_{\rm d}) = F(z,\rho_{\rm d},\rho_{\rm b}). \qquad (17.18)$$

What eq. 17.18 means is the following: If we have a circular beam of radius $\rho_{\rm b}$ and a circular detection region of radius $\rho_{\rm d}$, then the response we calculate is the same if we had a circular beam of radius $\rho_{\rm b}$ and a circular detection region of radius $\rho_{\rm d}$! The gain in efficiency comes when we wish to calculate the response of a small detector in a large area beam. If one does the calculation directly, then much computer time is squandered tracking particles that may never reach the detector. By using the reciprocity theorem one calculates the same quantity faster.

In an extreme form the reciprocity theorem takes the form [ICR84],

$$\lim_{\epsilon\longrightarrow 0} F(z,\rho_{\rm b},\epsilon) = \lim_{\epsilon\longrightarrow 0} F(z,\epsilon,\rho_{\rm b}), \qquad (17.19)$$

which allows one to calculate the “central axis” depth-dose for a finite radius beam by scoring the dose in a finite region from a zero-area beam. The gain in efficiency in this case is infinite! The radius, $\rho_{\rm b}$, can even be infinite to simulate a broad beam.

A few remarks about the reciprocity theorem and it’s derivation should be made. If the response function, $f(z,|\boldsymbol{\rho}|)$, has a finite lateral extent, then the restriction that the geometry should be semi-infinite may be relaxed as long as the geometry, including the inhomogeneous slabs, is big enough to contain all of the incident beam once the detection region radius and the beam radius are exchanged. Unfortunately, electron-photon beams always produce infinitely wide response functions owing to radiation scatter and bremsstrahlung photon creation. In practice, however, the lateral tails often contribute so little that simulation (and experiments!) in finite geometries is useful. Also, in the above development it was assumed that the detection region was infinitely thin. This is not a necessary approximation but this detail was omitted for clarity. The interested reader is encouraged to repeat the derivation with a detection region of finite extent. The derivation proceeds in the same manner but with more cumbersome equations.

17.3.4 Use of geometry symmetry

In the previous section, we saw that the use of some of the inherent symmetry of the geometry realised considerable increase in efficiency. Some uses of symmetry are more obvious, for example, the use of cylindrical-planar or spherical-conical simulation geometries if both the source and target configurations contain these symmetries. Other uses of symmetry are less obvious but still important. These applications involve the use of reflecting planes to mimic some of the inherent symmetry.

For example, consider the geometry depicted in fig. 17.7. In this case, an infinite square lattice of cylinders is irradiated uniformly from the top. The cylinders are all uniform and aligned. How should one approach this problem? Clearly, one can not model an infinite array of cylinders. If one tried, one would have to pick a finite set and decide somehow that it was big enough. Instead, it is much more efficient to exploit the symmetry of the problem. It turns out that in this instance, one needs to transport particles in only 1/8'th of a cylinder! To see this we find the symmetries in this problem. In fig. 17.7 we have drawn three planes of symmetry in the problem, planes **a**, **b**, and **c**[5]. There is reflection symmetry for each of these planes. Therefore, to mimic the infinite lattice, any particles that strike these reflecting planes should be reflected. One only needs to transport particles in the region bounded by the reflecting planes. Because of the highly symmetric nature of the problem, we only need to perform the simulation in a portion of the cylinder and the "response" functions for the rest of the lattice is found by reflection.

The rule for particle reflection about a plane of arbitrary orientation is easy to derive. Let $\vec{\mathbf{u}}$ be the unit direction vector of a particle and $\vec{\mathbf{n}}$ be the unit direction normal of the reflecting plane. Now divide the particle's direction vector into two portions, $\vec{\mathbf{u}}_{\|}$, parallel to $\vec{\mathbf{n}}$, and $\vec{\mathbf{u}}_{\perp}$, perpendicular to $\vec{\mathbf{n}}$. The parallel part gets reflected, $\vec{\mathbf{u}}'_{\|} = -\vec{\mathbf{u}}_{\|}$, and the perpendicular part remains unchanged, $\vec{\mathbf{u}}'_{\perp} = \vec{\mathbf{u}}_{\perp}$. That is, the new direction vector is $\vec{\mathbf{u}}' = -\vec{\mathbf{u}}_{\|} + \vec{\mathbf{u}}_{\perp}$. Another

[5] Note that this symmetry applies only to a square lattice, where the spacing is the same for the x and y-axes. For a rectangular symmetry, the planes of reflection would be somewhat different. There would be no plane **c** as for the square lattice in fig. 17.7.

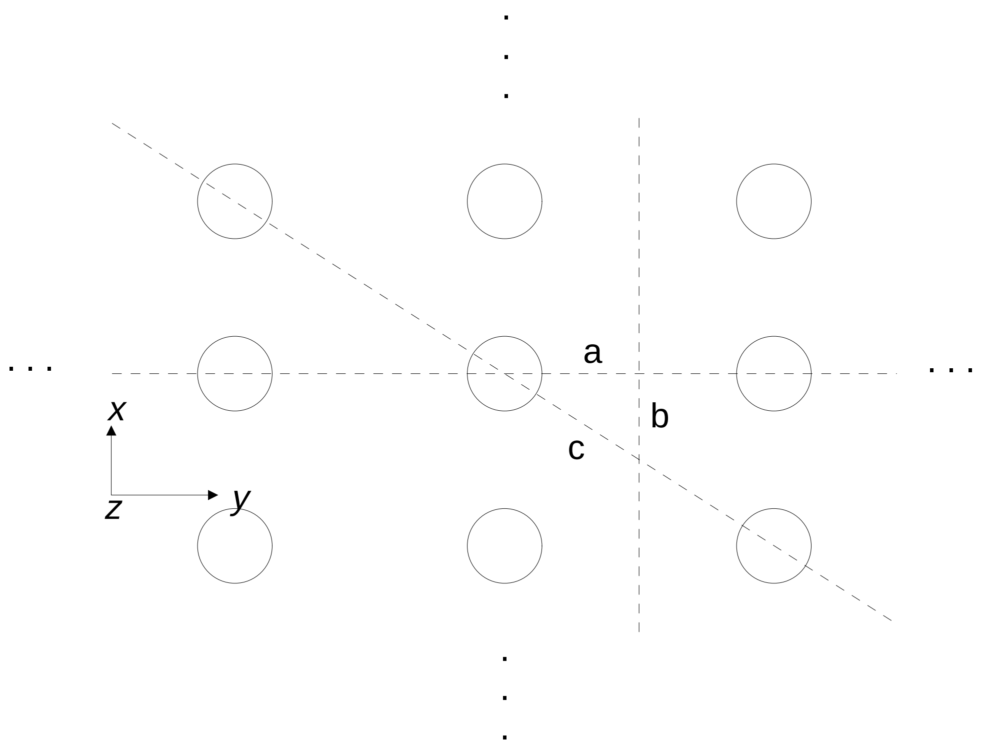

Figure 17.7: Top end view of an infinite square lattice of cylinders. Three planes of symmetry are drawn, **a**, **b**, and **c**. A complete simulation of the entire lattice may be performed by restricting the transport to the interior of the three planes. When a particle strikes a plane it is reflected back in, thereby mimicking the symmetry associated with this plane.

way of writing this is,

$$\vec{\mathbf{u}}' = \vec{\mathbf{u}} - \mathbf{2}(\vec{\mathbf{u}} \cdot \vec{\mathbf{n}})\vec{\mathbf{n}}. \tag{17.20}$$

Applying eq. 17.20 to the problem in fig. 17.7, we have: For reflection at plane **a**, $(u_x', u_y', u_z') = (-u_x, u_y, u_z)$. For reflection at plane **b**, $(u_x', u_y', u_z') = (u_x, -u_y, u_z)$. For reflection at plane **c**, $(u_x', u_y', u_z') = (-u_y, -u_x, u_z)$. The use of this reflection technique can result in great gains in efficiency. Most practical problems will not enjoy such a great amount of symmetry but one is encouraged to make use of any available symmetry. The saving in computing time is well worth the extra care and coding.

Bibliography

[Bie89] A. F. Bielajew. Electron Transport in $\vec{E}$ and $\vec{B}$ Fields. In T.M. Jenkins, W.R. Nelson, A. Rindi, A.E. Nahum, and D.W.O. Rogers, editors, *Monte Carlo Transport of Electrons and Photons*, pages 421 – 434. Plenum Press, New York, 1989.

[BMC89] A. F. Bielajew, R. Mohan, and C. S. Chui. Improved bremsstrahlung photon angular sampling in the EGS4 code system. *National Research Council of Canada Report PIRS-0203*, 1989.

[BR87] A. F. Bielajew and D. W. O. Rogers. PRESTA: The Parameter Reduced Electron-Step Transport Algorithm for electron Monte Carlo transport. *Nuclear Instruments and Methods*, B18:165 – 181, 1987.

[BRN85] A. F. Bielajew, D. W. O. Rogers, and A. E. Nahum. Monte Carlo simulation of ion chamber response to ^{60}Co – Resolution of anomalies associated with interfaces,. *Phys. Med. Biol.*, 30:419 – 428, 1985.

[Car81] G. A. Carlsson. Effective Use of Monte Carlo Methods. *Report ULi-RAD-R-049, Department of Radiology, Linköping University, Linköping, Sweden*, 1981.

[Fel67] W. Feller. *An introduction to probability theory and its applications, Volume I, 3rd Edition.* Wiley, New York, 1967.

[FRR90] B. A. Faddegon, C. K. Ross, and D. W. O. Rogers. Forward directed bremsstrahlung of 10 – 30 MeV electrons incident on thick targets of Al and Pb. *Medical Physics*, 17:773 – 785, 1990.

[HB85] J. S. Hendricks and T. E. Booth. Monte-Carlo Methods and Applications in Neutronics, Photonics and Statistical Physics. *(R Alcouffe, R Dautray, A Forster, G Ledanois, and B Mercier eds.)*, page 83, 1985.

[HH64] J. M. Hammersley and D. C. Handscomb. Monte Carlo Methods. *(John Wiley and Sons, New York)*, 1964.

[ICR84] ICRU. Radiation Dosimetry: Electron beams with energies between 1 and 50 MeV. ICRU Report 35, ICRU, Washington D.C., 1984.

[Kah56] H. Kahn. Use of different Monte Carlo sampling techniques. *in Symposium on Monte Carlo Methods, (H.A. Meyer ed.) (John Wiley and Sons, New York)*, pages 146 – 190, 1956.

[Lun81] T. Lund. An Introduction to the Monte Carlo Method. *Report HS-RP/067, CERN, Geneva*, 1981.

[MI75] E. J. McGrath and D. C. Irving. Techniques for Efficient Monte Carlo Simulation, Vols. I, II, and III. *Report ORNL-RSIC-38, Radiation Shielding Information Center, Oak Ridge National Laboratory, Oak Ridge, Tennessee*, 1975.

[NHR85] W. R. Nelson, H. Hirayama, and D. W. O. Rogers. The EGS4 Code System. Report SLAC–265, Stanford Linear Accelerator Center, Stanford, Calif, 1985.

[NNS$^+$90] Y. Namito, W. R. Nelson, S. M. Seltzer, A. F. Bielajew, and D. W. O. Rogers. Low-energy x-ray production studies using the EGS4 code system . *Med. Phys. (abstract)*, 17:557, 1990.

[RB84] D. W. O. Rogers and A. F. Bielajew. The use of EGS for Monte Carlo calculations in medical physics. *Report PXNR-2692, National Research Council of Canada, (Ottawa, Canada K1A 0R6)*, 1984.

[RB85] D. W. O. Rogers and A. F. Bielajew. Calculated buildup curves for photons with energies up to ^{60}Co . *Med. Phys.*, 12:738 – 744, 1985.

[REBvD88] D. W. O. Rogers, G. M Ewart, A. F. Bielajew, and G. van Dyk. Calculation of Electron Contamination in a ^{60}Co Therapy Beam. *In "Proceedings of the IAEA International Symposium on Dosimetry in Radiotherapy" (IAEA, Vienna), Vol 1*, pages 303 – 312, 1988.

[Rog84a] D. W. O. Rogers. Fluence to Dose Equivalent Conversion Factors Calculated with EGS3 for Electrons from 100 keV to 20 GeV and Photons from 20 keV to 20 GeV. *Health Physics*, 46:891 – 914, 1984.

[Rog84b] D. W. O. Rogers. Low energy electron transport with EGS. *Nucl. Inst. Meth.*, 227:535 – 548, 1984.

Problems

1.

Chapter 18

Code Library

This distribution resides on AFS in `~bielajew/Public/NE590`.

Conventions:

Subroutine and code names are written in `teletype font`.

The subroutines are distributed as combined Fortran77 source files. They are known to compile error-free on SunOS and Linux unix machines.

18.1 Utility/General

`Readme`

A general “readme” file, an ASCII condensation of this chapter.

stats.f

Statistical analysis routine.

Given:

$$\sum_{i=1}^{N} x_i \quad , \quad \sum_{i=1}^{N} x_i^2 \quad \text{and} \quad N \ ,$$

calculates:

$$\overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \quad \text{and} \quad s_{\overline{x}} = \sqrt{\frac{\left(\frac{1}{N}\sum_{i=1}^{N} x_i^2\right) - \overline{x}^2}{N-1}}$$

```
C23456789|123456789|123456789|123456789|123456789|123456789|123456789|12

      subroutine stats(sumx,sumx2,avgx,sigmax,N)

C  Given a sum of values and the associated sum of squares, this routine
C  computes the average and the estimated error of the mean.

      implicit none

      integer N  ! Input: Number of samples

      real
     *     sumx, ! Input: The sum of measurements
     *     sumx2 ! Input: The sum of the squares of measurements

      real
     *     avgx,  ! Ouput: The average
     *     sigmax ! Ouput: The estimated error of the mean

      avgx = sumx/N
      sigmax = sqrt(max(0e0,sumx2/N - avgx**2)/(N-1))

      return
      end
```

18.2 Subroutines for random number generation

`rngmc1.f`

Calculates:

$$X_{n+1} = \mathrm{mod}(aX_n + c, 2^{32}) \ ,$$

and returns random numbers uniform over the range $0 \leq r \leq 1$.:

$$r_{n+1} = 1/2 + X_{n+1}/2^{32} \ ,$$

where $a = 69069$, $c = 0$, $X_0 = 123456789$.

The sequence length is 2^{30}.

Restrictions: This routine only works on computer architectures that provide 2's complement 32-bit integers.

```
C23456789|123456789|123456789|123456789|123456789|123456789|123456789|12

      real function rng()

C  A simple multiplicative congruential pseudo random number generator using
C  one of Knuth's magic multpliers, 69069

C  Sequence length: 2**30

      implicit none

      integer ixx ! The pseudo-random integer

      data ixx/123456789/ ! The intialization value
      save ixx            ! Must be saved for subsequent passes

      ixx = ixx * 69069              ! The scrambling process
      rng = 0.5 + ixx*0.23283064e-09 ! The conversion to real 0 <= rng <= 1

      return
      end
```

`rngmc2.f`

Calculates:

$$X_{n+1} = \mathrm{mod}(aX_n + c, 2^{32}) \ ,$$

and returns random numbers uniform over the range $0 < r < 1.$:

$$r_{n+1} = 1/2 + X_{n+1}/2^{32} \ ,$$

where $a = 69069$, $c = 0$, $X_0 = 123456789$.

The sequence length is 2^{30}.

Restrictions: This routine only works on computer architectures that provide 2's complement 32-bit integers.

```
C23456789|123456789|123456789|123456789|123456789|123456789|123456789|12

      real function rng()

C  A simple multiplicative congruential pseudo random number generator using
C  one of Knuth's magic multpliers, 69069

C  Sequence length: 2**30

      implicit none

      integer ixx ! The pseudo-random integer

      data ixx/123456789/ ! The intialization value
      save ixx            ! Must be saved for subsequent passes

      ixx = ixx * 69069              ! The scrambling process
      rng = 0.5 + ixx*0.2328306e-09  ! The conversion to real 0 < rng < 1

      return
      end
```

`rngmc3.f`

Calculates:

$$X_{n+1} = \mathrm{mod}(aX_n + c, 2^{32}) \ ,$$

and returns random numbers uniform over the range $0 \leq r \leq 1.$:

$$r_{n+1} = 1/2 + X_{n+1}/2^{32} \ ,$$

where $a = 663608941$, $c = 0$, $X_0 = 123456789$.

The sequence length is 2^{30}.

Restrictions: This routine only works on computer architectures that provide 2's complement 32-bit integers.

```
C13456789|123456789|123456789|123456789|123456789|123456789|123456789|12

      real function rng()

C  A simple multiplicative congruential pseudo random number generator using
C  the default magic multiplier, 663608941

C  Sequence length: 2**30

      implicit none

      integer ixx ! The pseudo-random integer

      data ixx/123456789/ ! The intialization value
      save ixx            ! Must be saved for subsequent passes

      ixx = ixx *   663608941        ! The scrambling process
      rng = 0.5 + ixx*0.23283064e-09 ! The conversion to real 0 <= rng <= 1

      return
      end
```

rngmc4.f

Calculates:

$$X_{n+1} = \mathrm{mod}(aX_n + c, 2^{32}) \ ,$$

and returns random numbers uniform over the range $0 < r < 1.$:

$$r_{n+1} = 1/2 + X_{n+1}/2^{32} \ ,$$

where $a = 663608941$, $c = 0$, $X_0 = 123456789$.

The sequence length is 2^{30}.

Restrictions: This routine only works on computer architectures that provide 2's complement 32-bit integers.

```
C13456789|123456789|123456789|123456789|123456789|123456789|123456789|12

      real function rng()

C  A simple multiplicative congruential pseudo random number generator using
C  the default magic multiplier, 663608941

C  Sequence length: 2**30

      implicit none

      integer ixx ! The pseudo-random integer

      data ixx/123456789/ ! The intialization value
      save ixx            ! Must be saved for subsequent passes

      ixx = ixx *   663608941        ! The scrambling process
      rng = 0.5 + ixx*0.2328306e-09  ! The conversion to real 0 < rng < 1

      return
      end
```

`rnglc1.f`

Calculates:

$$X_{n+1} = \mathrm{mod}(aX_n + c, 2^{32}) \ ,$$

and returns random numbers uniform over the range $0 \le r \le 1.$:

$$r_{n+1} = 1/2 + X_{n+1}/2^{32} \ ,$$

where $a = 69069$, $c = 987654321$, $X_0 = 123456789$.

The sequence length is 2^{32}.

Restrictions: This routine only works on computer architectures that provide 2's complement 32-bit integers.

```
C13456789|123456789|123456789|123456789|123456789|123456789|123456789|12

      real function rng()

C  A simple congruential pseudo random number generator using
C  one of Knuth's magic multpliers, 69069

C  Sequence length: 2**32

      implicit none

      integer ixx ! The pseudo-random integer

      data ixx/123456789/ ! The intialization value
      save ixx            ! Must be saved for subsequent passes

      ixx = ixx * 69069 + 987654321  ! The scrambling process
      rng = 0.5 + ixx*0.23283064e-09 ! The conversion to real 0 <= rng <= 1

      return
      end
```

`rnglc2.f`

Calculates:

$$X_{n+1} = \mathrm{mod}(aX_n + c, 2^{32}) \ ,$$

and returns random numbers uniform over the range $0 < r < 1.$:

$$r_{n+1} = 1/2 + X_{n+1}/2^{32} \ ,$$

where $a = 69069$, $c = 987654321$, $X_0 = 123456789$.

The sequence length is 2^{32}.

Restrictions: This routine only works on computer architectures that provide 2's complement 32-bit integers.

```
C13456789|123456789|123456789|123456789|123456789|123456789|123456789|12

      real function rng()

C  A simple congruential pseudo random number generator using
C  one of Knuth's magic multpliers, 69069

C  Sequence length: 2**32

      implicit none

      integer ixx ! The pseudo-random integer

      data ixx/123456789/ ! The intialization value
      save ixx            ! Must be saved for subsequent passes

      ixx = ixx * 69069 + 987654321  ! The scrambling process
      rng = 0.5 + ixx*0.2328306e-09  ! The conversion to real 0 < rng < 1

      return
      end
```

`rnglc3.f`

Calculates:

$$X_{n+1} = \mathrm{mod}(aX_n + c, 2^{32}) \ ,$$

and returns random numbers uniform over the range $0 \leq r \leq 1.$:

$$r_{n+1} = 1/2 + X_{n+1}/2^{32} \ ,$$

where $a = 663608941$, $c = 987654321$, $X_0 = 123456789$.

The sequence length is 2^{32}.

Restrictions: This routine only works on computer architectures that provide 2's complement 32-bit integers.

```
C13456789|123456789|123456789|123456789|123456789|123456789|123456789|12

      real function rng()

C  A simple congruential pseudo random number generator using
C  the default magic multiplier, 663608941

C  Sequence length: 2**32

      implicit none

      integer ixx ! The pseudo-random integer

      data ixx/123456789/ ! The intialization value
      save ixx            ! Must be saved for subsequent passes

      ixx = ixx * 663608941 + 987654321  ! The scrambing process
      rng = 0.5 + ixx*0.23283064e-09     ! The conversion to real 0 <= rng <= 1

      return
      end
```

`rnglc4.f`

Calculates:

$$X_{n+1} = \mathrm{mod}(aX_n + c, 2^{32}) \ ,$$

and returns random numbers uniform over the range $0 < r < 1.$:

$$r_{n+1} = 1/2 + X_{n+1}/2^{32} \ ,$$

where $a = 663608941$, $c = 987654321$, $X_0 = 123456789$.

The sequence length is 2^{32}.

Restrictions: This routine only works on computer architectures that provide 2's complement 32-bit integers.

```
C13456789|123456789|123456789|123456789|123456789|123456789|123456789|12

      real function rng()

C  A simple congruential pseudo random number generator using
C  the default magic multiplier, 663608941

C  Sequence length: 2**32

      implicit none

      integer ixx ! The pseudo-random integer

      data ixx/123456789/ ! The intialization value
      save ixx            ! Must be saved for subsequent passes

      ixx = ixx * 663608941 + 987654321  ! The scrambing process
      rng = 0.5 + ixx*0.2328306e-09      ! The conversion to real 0 < rng < 1

      return
      end
```

rngranlux.f

The original RANLUX subroutine coded by Fred James [Jam94] following the theoretical development by Lüscher [L9̈4].

```
C The references are Lu:scher, Computer Physics Communications 79 (1994) 100
C and James, CPC 79 (1994) 111.

      SUBROUTINE RANLUX(RVEC,LENV)
C         Subtract-and-borrow random number generator proposed by
C         Marsaglia and Zaman, implemented by F. James with the name
C         RCARRY in 1991, and later improved by Martin Luescher
C         in 1993 to produce "Luxury Pseudorandom Numbers".
C     Fortran 77 coded by F. James, 1993
C
C       references:
C  M. Luscher, Computer Physics Communications  79 (1994) 100
C  F. James, Computer Physics Communications 79 (1994) 111
C
C   LUXURY LEVELS.
C   ------ ------      The available luxury levels are:
C
C  level 0  (p=24): equivalent to the original RCARRY of Marsaglia
C           and Zaman, very long period, but fails many tests.
C  level 1  (p=48): considerable improvement in quality over level 0,
C           now passes the gap test, but still fails spectral test.
C  level 2  (p=97): passes all known tests, but theoretically still
C           defective.
C  level 3  (p=223): DEFAULT VALUE.  Any theoretically possible
C           correlations have very small chance of being observed.
C  level 4  (p=389): highest possible luxury, all 24 bits chaotic.
C
C!!! ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C!!!  Calling sequences for RANLUX:                                  ++
C!!!      CALL RANLUX (RVEC, LEN)   returns a vector RVEC of LEN     ++
C!!!                   32-bit random floating point numbers between  ++
C!!!                   zero (not included) and one (also not incl.). ++
C!!!      CALL RLUXGO(LUX,INT,K1,K2) initializes the generator from  ++
C!!!               one 32-bit integer INT and sets Luxury Level LUX  ++
C!!!               which is integer between zero and MAXLEV, or if   ++
C!!!               LUX .GT. 24, it sets p=LUX directly.  K1 and K2   ++
C!!!               should be set to zero unless restarting at a break++
C!!!               point given by output of RLUXAT (see RLUXAT).    ++
C!!!      CALL RLUXAT(LUX,INT,K1,K2) gets the values of four integers++
C!!!               which can be used to restart the RANLUX generator ++
C!!!               at the current point by calling RLUXGO.  K1 and K2++
C!!!               specify how many numbers were generated since the ++
C!!!               initialization with LUX and INT.  The restarting  ++
C!!!               skips over  K1+K2*E9   numbers, so it can be long.++
C!!!   A more efficient but less convenient way of restarting is by: ++
C!!!      CALL RLUXIN(ISVEC)    restarts the generator from vector   ++
C!!!                   ISVEC of 25 32-bit integers (see RLUXUT)      ++
C!!!      CALL RLUXUT(ISVEC)    outputs the current values of the 25 ++
C!!!                 32-bit integer seeds, to be used for restarting ++
C!!!      ISVEC must be dimensioned 25 in the calling program        ++
C!!! ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      DIMENSION RVEC(LENV)
      DIMENSION SEEDS(24), ISEEDS(24), ISDEXT(25)
      PARAMETER (MAXLEV=4, LXDFLT=3)
      DIMENSION NDSKIP(0:MAXLEV)
      DIMENSION NEXT(24)
      PARAMETER (TWOP12=4096., IGIGA=1000000000,JSDFLT=314159265)
      PARAMETER (ITWO24=2**24, ICONS=2147483563)
      SAVE NOTYET, I24, J24, CARRY, SEEDS, TWOM24, TWOM12, LUXLEV
      SAVE NSKIP, NDSKIP, IN24, NEXT, KOUNT, MKOUNT, INSEED
      INTEGER LUXLEV
      LOGICAL NOTYET
      DATA NOTYET, LUXLEV, IN24, KOUNT, MKOUNT /.TRUE., LXDFLT, 0,0,0/
      DATA I24,J24,CARRY/24,10,0./
C                              default
C  Luxury Level   0     1     2   *3*    4
      DATA NDSKIP/0,   24,   73,  199,  365 /
Corresponds to p=24    48    97   223   389
C     time factor 1     2     3     6    10   on slow workstation
C                 1    1.5    2     3     5   on fast mainframe
C
C  NOTYET is .TRUE. if no initialization has been performed yet.
C              Default Initialization by Multiplicative Congruential
      IF (NOTYET) THEN
         NOTYET = .FALSE.
         JSEED = JSDFLT
         INSEED = JSEED
         WRITE(6,'(A,I12)') ' RANLUX DEFAULT INITIALIZATION: ',JSEED
         LUXLEV = LXDFLT
         NSKIP = NDSKIP(LUXLEV)
         LP = NSKIP + 24
         IN24 = 0
         KOUNT = 0
         MKOUNT = 0
         WRITE(6,'(A,I2,A,I4)')  ' RANLUX DEFAULT LUXURY LEVEL =  ',
     +        LUXLEV,'      p =',LP
            TWOM24 = 1.
         DO 25 I= 1, 24
            TWOM24 = TWOM24 * 0.5
         K = JSEED/53668
         JSEED = 40014*(JSEED-K*53668) -K*12211
         IF (JSEED .LT. 0)  JSEED = JSEED+ICONS
         ISEEDS(I) = MOD(JSEED,ITWO24)
   25    CONTINUE
         TWOM12 = TWOM24 * 4096.
         DO 50 I= 1,24
         SEEDS(I) = REAL(ISEEDS(I))*TWOM24
         NEXT(I) = I-1
   50    CONTINUE
         NEXT(1) = 24
         I24 = 24
         J24 = 10
         CARRY = 0.
         IF (SEEDS(24) .EQ. 0.) CARRY = TWOM24
      ENDIF
C
C          The Generator proper: "Subtract-with-borrow",
C          as proposed by Marsaglia and Zaman,
C          Florida State University, March, 1989
C
      DO 100 IVEC= 1, LENV
      UNI = SEEDS(J24) - SEEDS(I24) - CARRY
      IF (UNI .LT. 0.)  THEN
         UNI = UNI + 1.0
         CARRY = TWOM24
      ELSE
         CARRY = 0.
      ENDIF
      SEEDS(I24) = UNI
      I24 = NEXT(I24)
      J24 = NEXT(J24)
      RVEC(IVEC) = UNI
C  small numbers (with less than 12 "significant" bits) are "padded".
      IF (UNI .LT. TWOM12)  THEN
         RVEC(IVEC) = RVEC(IVEC) + TWOM24*SEEDS(J24)
C        and zero is forbidden in case someone takes a logarithm
         IF (RVEC(IVEC) .EQ. 0.)  RVEC(IVEC) = TWOM24*TWOM24
      ENDIF
C        Skipping to luxury.  As proposed by Martin Luscher.
      IN24 = IN24 + 1
      IF (IN24 .EQ. 24)  THEN
         IN24 = 0
         KOUNT = KOUNT + NSKIP
         DO 90 ISK= 1, NSKIP
         UNI = SEEDS(J24) - SEEDS(I24) - CARRY
         IF (UNI .LT. 0.)  THEN
            UNI = UNI + 1.0
            CARRY = TWOM24
         ELSE
            CARRY = 0.
         ENDIF
         SEEDS(I24) = UNI
         I24 = NEXT(I24)
         J24 = NEXT(J24)
   90    CONTINUE
      ENDIF
  100 CONTINUE
      KOUNT = KOUNT + LENV
      IF (KOUNT .GE. IGIGA)  THEN
         MKOUNT = MKOUNT + 1
         KOUNT = KOUNT - IGIGA
      ENDIF
      RETURN
C
C           Entry to input and float integer seeds from previous run
      ENTRY RLUXIN(ISDEXT)
         TWOM24 = 1.
         DO 195 I= 1, 24
         NEXT(I) = I-1
  195    TWOM24 = TWOM24 * 0.5
         NEXT(1) = 24
         TWOM12 = TWOM24 * 4096.
      WRITE(6,'(A)') ' FULL INITIALIZATION OF RANLUX WITH 25 INTEGERS:'
      WRITE(6,’(5X,5I12)’) ISDEXT
      DO 200 I= 1, 24
      SEEDS(I) = REAL(ISDEXT(I))*TWOM24
  200 CONTINUE
      CARRY = 0.
      IF (ISDEXT(25) .LT. 0)  CARRY = TWOM24
      ISD = IABS(ISDEXT(25))
      I24 = MOD(ISD,100)
      ISD = ISD/100
      J24 = MOD(ISD,100)
      ISD = ISD/100
      IN24 = MOD(ISD,100)
      ISD = ISD/100
      LUXLEV = ISD
        IF (LUXLEV .LE. MAXLEV) THEN
          NSKIP = NDSKIP(LUXLEV)
          WRITE (6,’(A,I2)’) ’ RANLUX LUXURY LEVEL SET BY RLUXIN TO: ’,
     +                         LUXLEV
        ELSE  IF (LUXLEV .GE. 24) THEN
          NSKIP = LUXLEV - 24
          WRITE (6,’(A,I5)’) ’ RANLUX P-VALUE SET BY RLUXIN TO:’,LUXLEV
        ELSE
          NSKIP = NDSKIP(MAXLEV)
          WRITE (6,’(A,I5)’) ’ RANLUX ILLEGAL LUXURY RLUXIN: ’,LUXLEV
          LUXLEV = MAXLEV
        ENDIF
      INSEED = -1
      RETURN
C
C                    Entry to ouput seeds as integers
      ENTRY RLUXUT(ISDEXT)
      DO 300 I= 1, 24
         ISDEXT(I) = INT(SEEDS(I)*TWOP12*TWOP12)
  300 CONTINUE
      ISDEXT(25) = I24 + 100*J24 + 10000*IN24 + 1000000*LUXLEV
      IF (CARRY .GT. 0.)  ISDEXT(25) = -ISDEXT(25)
      RETURN
C
C                    Entry to output the "convenient" restart point
      ENTRY RLUXAT(LOUT,INOUT,K1,K2)
      LOUT = LUXLEV
      INOUT = INSEED
      K1 = KOUNT
      K2 = MKOUNT
      RETURN
C
C                    Entry to initialize from one or three integers
      ENTRY RLUXGO(LUX,INS,K1,K2)
         IF (LUX .LT. 0) THEN
            LUXLEV = LXDFLT
         ELSE IF (LUX .LE. MAXLEV) THEN
            LUXLEV = LUX
         ELSE IF (LUX .LT. 24 .OR. LUX .GT. 2000) THEN
            LUXLEV = MAXLEV
            WRITE (6,'(A,I7)') ' RANLUX ILLEGAL LUXURY RLUXGO: ',LUX
         ELSE
            LUXLEV = LUX
            DO 310 ILX= 0, MAXLEV
              IF (LUX .EQ. NDSKIP(ILX)+24)  LUXLEV = ILX
  310       CONTINUE
         ENDIF
      IF (LUXLEV .LE. MAXLEV)  THEN
         NSKIP = NDSKIP(LUXLEV)
         WRITE(6,'(A,I2,A,I4)') ' RANLUX LUXURY LEVEL SET BY RLUXGO :',
     +        LUXLEV,'     P=', NSKIP+24
      ELSE
          NSKIP = LUXLEV - 24
          WRITE (6,'(A,I5)') ' RANLUX P-VALUE SET BY RLUXGO TO:',LUXLEV
      ENDIF
      IN24 = 0
      IF (INS .LT. 0)  WRITE (6,'(A)')
     +   ' Illegal initialization by RLUXGO, negative input seed'
      IF (INS .GT. 0)  THEN
        JSEED = INS
        WRITE(6,'(A,3I12)') ' RANLUX INITIALIZED BY RLUXGO FROM SEEDS',
     +      JSEED, K1,K2
      ELSE
        JSEED = JSDFLT
        WRITE(6,'(A)')' RANLUX INITIALIZED BY RLUXGO FROM DEFAULT SEED'
      ENDIF
      INSEED = JSEED
      NOTYET = .FALSE.
      TWOM24 = 1.
         DO 325 I= 1, 24
```

```
           TWOM24 = TWOM24 * 0.5
         K = JSEED/53668
         JSEED = 40014*(JSEED-K*53668) -K*12211
         IF (JSEED .LT. 0)  JSEED = JSEED+ICONS
         ISEEDS(I) = MOD(JSEED,ITWO24)
  325    CONTINUE
      TWOM12 = TWOM24 * 4096.
         DO 350 I= 1,24
         SEEDS(I) = REAL(ISEEDS(I))*TWOM24
         NEXT(I) = I-1
  350    CONTINUE
      NEXT(1) = 24
      I24 = 24
      J24 = 10
      CARRY = 0.
      IF (SEEDS(24) .EQ. 0.) CARRY = TWOM24
C        If restarting at a break point, skip K1 + IGIGA*K2
C        Note that this is the number of numbers delivered to
C        the user PLUS the number skipped (if luxury .GT. 0).
      KOUNT = K1
      MKOUNT = K2
      IF (K1+K2 .NE. 0)  THEN
        DO 500 IOUTER= 1, K2+1
          INNER = IGIGA
          IF (IOUTER .EQ. K2+1)  INNER = K1
          DO 450 ISK= 1, INNER
            UNI = SEEDS(J24) - SEEDS(I24) - CARRY
            IF (UNI .LT. 0.)  THEN
               UNI = UNI + 1.0
               CARRY = TWOM24
            ELSE
               CARRY = 0.
            ENDIF
            SEEDS(I24) = UNI
            I24 = NEXT(I24)
            J24 = NEXT(J24)
  450     CONTINUE
  500   CONTINUE
C         Get the right value of IN24 by direct calculation
        IN24 = MOD(KOUNT, NSKIP+24)
        IF (MKOUNT .GT. 0)  THEN
           IZIP = MOD(IGIGA, NSKIP+24)
           IZIP2 = MKOUNT*IZIP + IN24
           IN24 = MOD(IZIP2, NSKIP+24)
        ENDIF
C       Now IN24 had better be between zero and 23 inclusive
        IF (IN24 .GT. 23) THEN
           WRITE (6,'(A/A,3I11,A,I5)')
     +    '  Error in RESTARTING with RLUXGO:','  The values', INS,
     +     K1, K2, ' cannot occur at luxury level', LUXLEV
           IN24 = 0
        ENDIF
      ENDIF
      RETURN
      END
```

`rngranlux2.f`

`RANLUX` “subtract-with-borrow” pseudo-random number generator, long period, luxury = 2.

Restrictions: Must be linked with `rngranlux.f`.

```
C23456789|123456789|123456789|123456789|123456789|123456789|123456789|12

      real function rng()

C  Seeds and calls the ranlux routine with luxury level 2

      implicit none

      logical
     *    init ! Initialization switch

      integer
     *    ixx,   ! Internal: Seeding integer for ranlux
     *    luxury ! Internal: Luxury value for ranlux

      data ixx/123456789/ ! The intialization value
      save ixx            ! Must be saved for subsequent passes

      data luxury/2/      ! The intialization value
      save luxury         ! Save it

      data init/.false./  ! The intialization value
      save init           ! Save it

      if (.not.init) then
          call rluxgo(luxury,ixx,0,0) ! Initializes
          init = .true.               ! Flip the switch
      end if

      call ranlux(rng, 1)

      return
      end
```

rngranlux3.f

RANLUX “subtract-with-borrow” pseudo-random number generator, long period, luxury = 3.

Restrictions: Must be linked with rngranlux.f.

```
C23456789|123456789|123456789|123456789|123456789|123456789|123456789|12

      real function rng()

C  Seeds and calls the ranlux routine with luxury level 3

      implicit none

      logical
     *    init ! Initialization switch

      integer
     *    ixx,   ! Internal: Seeding integer for ranlux
     *    luxury ! Internal: Luxury value for ranlux

      data ixx/123456789/ ! The intialization value
      save ixx            ! Must be saved for subsequent passes

      data luxury/3/      ! The intialization value
      save luxury         ! Save it

      data init/.false./  ! The intialization value
      save init           ! Save it

      if (.not.init) then
          call rluxgo(luxury,ixx,0,0) ! Initializes
          init = .true.               ! Flip the switch
      end if

      call ranlux(rng, 1)

      return
      end
```

`rngranlux4.f`

`RANLUX` “subtract-with-borrow” pseudo-random number generator, long period, luxury = 4.

Restrictions: Must be linked with `rngranlux.f`.

```
C23456789|123456789|123456789|123456789|123456789|123456789|123456789|12

      real function rng()

C  Seeds and calls the ranlux routine with luxury level 4

      implicit none

      logical
     *    init ! Initialization switch

      integer
     *    ixx,   ! Internal: Seeding integer for ranlux
     *    luxury ! Internal: Luxury value for ranlux

      data ixx/123456789/ ! The intialization value
      save ixx            ! Must be saved for subsequent passes

      data luxury/4/      ! The intialization value
      save luxury         ! Save it

      data init/.false./  ! The intialization value
      save init           ! Save it

      if (.not.init) then
          call rluxgo(luxury,ixx,0,0) ! Initializes
          init = .true.               ! Flip the switch
      end if

      call ranlux(rng, 1)

      return
      end
```

18.3 Subroutines for particle transport and deflection

azimuthal0.f

Uniform sampling of azimuthal direction vectors in 2-D.

The direction cosines are determined using the following method:

$$u = \cos(2\pi r) \quad \text{and} \quad v = \sin(2\pi r)$$

where r is a random number uniform on the range $0 \leq r < 1$, or $0 < r \leq 1$, or $0 \leq r \leq 1$.

```
C23456789|123456789|123456789|123456789|123456789|123456789|123456789|12

      subroutine azimuthal(u,v)

      implicit none

C Computes unit normalized random transverse direction cosines u,v
C normalized so that u**2 + v**2 = 1

      real
     *    u,  ! Output: final x-axis direction cosine
     *    v   ! Output: final y-axis direction cosine

      real
     *    phi,   ! Internal: azimuthal angle phi
     *    twopi, ! Internal: value of 2*pi
     *    rng    ! Internal: a random number uniform on the range [0,1) or
                 !           (0,1] or [0,1].
                 !           (,) => exclusive
                 !           [,] => inclusive

      data twopi/6.283185/
      save twopi           ! Saved for repeated use.

      phi = twopi*rng()
      u = cos(phi)
      v = sin(phi)

      return

      end
```

`rotate.f`

Rotates $\vec{u} \longrightarrow \vec{u}'$ according to $\cos(\Theta)$.

```
C23456789|123456789|123456789|123456789|123456789|123456789|123456789|12

      subroutine rotate(u,v,w,ws)
      implicit none

C  Rotates the direction vectors (u,v,w) after a scatter by polar cosine ws

C  Some notation conventions:

C  Theta, Phi refer to the scattering angles in the reference frame where the
C         z-axis is aligned with the initial particle direction

C  theta, phi refer to the original particle directions with respect to the
C             laboratory frame of reference

      real
     *    u,  ! Input:  initial x-axis direction cosine
     *        ! Output: final x-axis direction cosine
     *    v,  ! Input:  initial y-axis direction cosine
     *        ! Output: final y-axis direction cosine
     *    w,  ! Input:  initial z-axis direction cosine
     *        ! Output: final z-axis direction cosine
     *    ws  ! Input:  cos(Theta)

      real
     *    sinThetas, ! Internal: sin(Theta), scattering angle
     *    sintheta,  ! Internal: sin(theta), initial angle theta
     *    sinthetai, ! Internal: 1/sin(theta)
     *    cosphi,    ! Internal: cos(phi), initial angle phi
     *    sinphi,    ! Internal: sin(phi), initial angle phi
     *    us,        ! Internal: scattering x-direction cosine,
                     !           sin(Theta)*cos(Phi)
     *    vs,        ! Internal: scattering y-direction cosine
                     !           sin(Theta)*sin(Phi)
     *    u2v2       ! Internal: u**2 + v**2 = sin(Theta)**2

C     Determine the azimuthal direction cosines
```

```
      call azimuthal(us,vs) ! Unit normalized 2-D transverse direction cosines
                            ! i.e. cos(Phi), sin(Phi)

C     Convert 2-D transverse to 3-D transverse direction cosines

      sinThetas = sqrt(1 - ws**2)
      us = sinThetas*us     ! i.e. us = sin(Theta)*cos(Phi)
      vs = sinThetas*vs     ! i.e. vs = sin(Theta)*sin(Phi)

      u2v2 = u**2 + v**2    ! i.e. u2v2 = sin(theta)**2

      if (u2v2 .gt. 0e0) then ! General scattering case

          sintheta  = sqrt(u2v2)   ! i.e. sin(theta)
          sinthetai = 1/sintheta
          cosphi    = u*sinthetai  ! i.e. cos(phi)
          sinphi    = v*sinthetai  ! i.e. sin(phi)

          u = u*ws + us*w*cosphi - vs*sinphi
          v = v*ws + us*w*sinphi + vs*cosphi
          w = w*ws - us*sintheta

          return

      end if

C     This is the scattering from initial Z-direction case

      u = us
      v = vs
      w = ws

      return

      end
```

18.4 Subroutines for modeling interactions

`isotropic.f`

Samples the z-axis direction cosine in isotropic scattering from the probability distribution:

$$p(\theta)\mathrm{d}\theta = \frac{\sin\theta\ \mathrm{d}\theta}{2} \qquad 0 \le \theta \le \pi$$

where θ is the polar angle. The direct method is employed giving

$$\cos\theta = 2r - 1$$

where r is a random number uniform on the range $0 \le r < 1$, or $0 < r \le 1$, or $0 \le r \le 1$.

```
C23456789|123456789|123456789|123456789|123456789|123456789|123456789|12

      subroutine isotropic(costhe)
      implicit none

C  Computes direction cosine for isotropic scattering

      real
     *    costhe ! Output:  cos(Theta)

      real
     *    rng ! Internal: random number

      costhe = 2e0*rng() - 1e0

      return

      end
```

`rutherford.f`

Samples the z-axis direction cosine in "Rutherfordian" scattering from the probability distribution:

$$p(\mu) = \frac{a(2+a)}{2}\frac{1}{(1-\mu+a)^2} \quad ; \quad -1 \le \mu \le 1,$$

where $\mu = \cos(\theta)$. The direct method is employed giving

$$\mu = 1 - 2a\frac{1-r}{a+2r}$$

where r is a random number uniform on the range $0 \le r < 1$, or $0 < r \le 1$, or $0 \le r \le 1$.

Restrictions: $a \ge 0$ otherwise the routine will stop execution. $a = 0$ is treated as a special case (no scattering).

```
C23456789|123456789|123456789|123456789|123456789|123456789|123456789|12

      subroutine rutherford(costhe,eta)
      implicit none

C  Rutherford-type scattering

      real
     *    costhe, ! Output:  cos(Theta)
     *    eta     ! Input:  screening factor

      real
     *    rng     ! Internal: random number

      if (eta .gt. 0e0) then
          costhe = rng()
          costhe = 1e0 - 2e0*eta*(1e0 - costhe)/(eta + 2e0*costhe)
          return
      elseif (eta .eq. 0e0) then
          costhe = 1e0
          return
      else
          write(6,*) 'Stopped in subroutine rutherford.'
          write(6,*) 'Eta was less than zero.'
          stop 1
      endif
      end
```

18.5 Subroutines for modeling geometry

`xplane.f`

Geometry routine for a plane normal to the x-axis.

```
C23456789|123456789|123456789|123456789|123456789|123456789|123456789|12

      subroutine xplane(p0,x0,u,s,inside,hit)
      implicit none

C  Calculates the distance of a particle to a planar surface normal to the
C  x-axis

      logical
     *    inside ! Input:  inside = .true.  => particle thinks it is inside
     *    ,      ! Input:  inside = .false. => particle thinks it is outside
     *    hit    ! Output: hit    = .true.  => particle would hit  the surface
                 ! Output: hit    = .false. => particle would miss the surface

      real
     *    p0,    ! Input:  Point at which the plane cuts the x-axis
     *    x0,    ! Input:  x-coordinate of the particle
     *    u ,    ! Input:  x-axis direction cosine of the particle
     *    s      ! Output: Distance to the surface (if hit)

      if (
     *    (inside.and.u.gt.0e0)      !headed towards the surface
     *    .or.
     *    (.not.inside.and.u.lt.0e0) !headed towards the surface
     *   )
     *then
          hit = .true.
          s   = max(0e0,(p0-x0)/u)
          return
      else
          hit = .false.
          return
      endif

      end
```

`yplane.f`

Geometry routine for a plane normal to the y-axis.

```
C23456789|123456789|123456789|123456789|123456789|123456789|123456789|12

      subroutine yplane(p0,y0,v,s,inside,hit)
      implicit none

C  Calculates the distance of a particle to a planar surface normal to the
C  y-axis

      logical
     *    inside ! Input:  inside = .true.  => particle thinks it is inside
     *    ,      ! Input:  inside = .false. => particle thinks it is outside
     *    hit    ! Output: hit    = .true.  => particle would hit  the surface
                 ! Output: hit    = .false. => particle would miss the surface

      real
     *    p0,    ! Input:  Point at which the plane cuts the y-axis
     *    y0,    ! Input:  y-coordinate of the particle
     *    v ,    ! Input:  y-axis direction cosine of the particle
     *    s      ! Output: Distance to the surface (if hit)

      if (
     *    (inside.and.v.gt.0e0)      !headed towards the surface
     *    .or.
     *    (.not.inside.and.v.lt.0e0) !headed towards the surface
     *   )
     *then
          hit = .true.
          s   = max(0e0,(p0-y0)/v)
          return
      else
          hit = .false.
          return
      endif

      end
```

zplane.f

Geometry routine for a plane normal to the z-axis.

```
C23456789|123456789|123456789|123456789|123456789|123456789|123456789|12

      subroutine zplane(p0,z0,w,s,inside,hit)
      implicit none

C  Calculates the distance of a particle to a planar surface normal to the
C  z-axis

      logical
     *    inside ! Input:  inside = .true.  => particle thinks it is inside
     *    ,      ! Input:  inside = .false. => particle thinks it is outside
     *    hit    ! Output: hit    = .true.  => particle would hit  the surface
                 ! Output: hit    = .false. => particle would miss the surface

      real
     *    p0,    ! Input:  Point at which the plane cuts the z-axis
     *    z0,    ! Input:  z-coordinate of the particle
     *    w ,    ! Input:  z-axis direction cosine of the particle
     *    s      ! Output: Distance to the surface (if hit)

      if (
     *    (inside.and.w.gt.0e0)      !headed towards the surface
     *    .or.
     *    (.not.inside.and.w.lt.0e0) !headed towards the surface
     *   )
     *then
          hit = .true.
          s   = max(0e0,(p0-z0)/w)
          return
      else
          hit = .false.
          return
      endif

      end
```

`csphere.f`

Geometry routine for a sphere, centered at the origin.

```
C23456789|123456789|123456789|123456789|123456789|123456789|123456789|12

      subroutine csphere(R,x0,y0,z0,u,v,w,s,inside,hit)
      implicit none

C  Calculates the distance to a sphere centered at (0,0,0)

      logical
     *    inside ! Input:  inside = .true.  => particle thinks it is inside
     *    ,      ! Input:  inside = .false. => particle thinks it is outside
     *    hit    ! Output: hit    = .true.  => particle would hit  the surface
                 ! Output: hit    = .false. => particle would miss the surface

      real
     *    R,     ! Input:  Radius of the cylinder
     *    x0,    ! Input:  x-coordinate of the particle
     *    y0,    ! Input:  y-coordinate of the particle
     *    z0,    ! Input:  z-coordinate of the particle
     *    u ,    ! Input:  x-axis direction cosine of the particle
     *    v ,    ! Input:  y-axis direction cosine of the particle
     *    w ,    ! Input:  w-axis direction cosine of the particle
     *    s      ! Output: Distance to the surface (if hit)

      real
     *     A,    ! Internal: Quadratic coefficient A
     *     B,    ! Internal: Quadratic coefficient B
     *     C     ! Internal: Quadratic coefficient C

      A = 1e0    ! i.e. u**2 + v**2 + w**2 = 1
      B = u*x0  + v*y0  + w*z0
      C = x0**2 + y0**2 + z0**2 - R**2

      call quadric(A,B,C,s,inside,hit) ! Get the generic quadric solution

      return
      end
```

`ccylz.f`

Geometry routine for a circular cylinder, aligned and concentric with the z-axis

```
C23456789|123456789|123456789|123456789|123456789|123456789|123456789|12

      subroutine ccylz(R,x0,y0,u,v,s,inside,hit)
      implicit none

C  Calculates the distance to a circular cylinder centered and aligned along
C  the z-axis

      logical
     *    inside ! Input:  inside = .true.  => particle thinks it is inside
     *    ,      ! Input:  inside = .false. => particle thinks it is outside
     *    hit    ! Output: hit    = .true.  => particle would hit  the surface
                 ! Output: hit    = .false. => particle would miss the surface

      real
     *    R,     ! Input:  Radius of the cylinder
     *    x0,    ! Input:  x-coordinate of the particle
     *    y0,    ! Input:  y-coordinate of the particle
     *    u ,    ! Input:  x-axis direction cosine of the particle
     *    v ,    ! Input:  y-axis direction cosine of the particle
     *    s      ! Output: Distance to the surface (if hit)

      real
     *     A,     ! Internal: Quadratic coefficient A
     *     B,     ! Internal: Quadratic coefficient B
     *     C      ! Internal: Quadratic coefficient C

      A = u**2  + v**2
      B = u*x0  + v*y0
      C = x0**2 + y0**2 - R**2

      call quadric(A,B,C,s,inside,hit) ! Get the generic quadric solution

      return
      end
```

quadric.f

Geometry routine for the general solution of closest point-of-flight positive intersection with a general quadric surface

```
C23456789|123456789|123456789|123456789|123456789|123456789|123456789|12

      subroutine quadric(A,B,C,s,inside,hit)
      implicit none

C  Calculates the first positive distance to an arbitrary quadric surface

      logical
     *    inside ! Input:  inside = .true.  => particle thinks it is inside
     *    ,      ! Input:  inside = .false. => particle thinks it is outside
     *    hit    ! Output: hit    = .true.  => particle would hit  the surface
                 ! Output: hit    = .false. => particle would miss the surface

      real
     *     A,    ! Input: Quadratic coefficient A
     *     B,    ! Input: Quadratic coefficient B
     *     C,    ! Input: Quadratic coefficient C
     *     s     ! Output: Distance to the surface (if hit)

      real
     *     Q     ! Internal: quadratic coefficient


      Q = B**2 - A*C

      if (Q .lt. 0e0) then
          hit = .false.
          return
      end if

      if (inside) then              !inside the surface
          if (B .le. 0e0) then      !headed away from surface
              if (A .gt. 0e0) then  !but, surface curving up
                  hit = .true.      !always a hit in this case
                  s = (sqrt(Q) - B)/A
                  return
              else                  !surface curving away and headed in
```

```
            hit = .false.     !never a hit in this case
            return
        end if
    else                      !headed toward the surface
        hit = .true.          !always a hit in this case
        s = max(0e0, -C/(sqrt(Q) + B))
        return
    end if
end if

!Must be outside the surface

if (B .ge. 0e0) then     !headed away
    if (A .ge. 0e0) then !surface curves away
        hit = .false.    !never a hit in this case
        return
    else                 !surface curves up
        hit = .true.     !always a hit in this case
        s = -(sqrt(Q) + B)/A
        return
    end if
else                     !headed toward the surface
    hit = .true.         !always a hit in this case
    s = max(0e0, C/(sqrt(Q) - B))
    return
end if

end
```

18.6 Test routines

`testiso.f`

Tests isotropic scattering

```
C23456789|123456789|123456789|123456789|123456789|123456789|123456789|12

      implicit none

C test isotropic scattering

      real
     *    costhe, ! Output:  cos(Theta)
     *    sinthe, ! Output:  sin(Theta)
     *    cosphi, ! Output:  cos(Phi)
     *    sinphi  ! Output:  sin(Phi)

      integer i

      do i = 1,1000

          call isotropic(costhe)
          sinthe = sqrt(1 - costhe**2)

          call azimuthal(cosphi,sinphi)

          write(1,*) 0e0, 0e0
          write(1,*) costhe, sinthe*cosphi
          write(1,*) 0e0,0e0
          write(2,*) 0e0, 0e0
          write(2,*) costhe, sinthe*sinphi
          write(2,*) 0e0,0e0
          write(3,*) 0e0, 0e0
          write(3,*) sinthe*cosphi, sinthe*sinphi
          write(3,*) 0e0,0e0
      end do

      end
```

testrng.f

Tests the random number generators

```
      implicit none
      integer i
      real rng, rn0, rn
      i = 0
      rn0 = rng()
      write(6,*) ’ Initial RN = ’,rn0
1     continue
      i = i + 1
      rn = rng()
      if (i .le. 50 .and. i .gt. 0) write(6,*) rn
      if (rn .eq. rn0) then
          write(6,*) ’RNG looped, stopping, i = ’, i
          write(6,*) ’Final RN = ’, rn
          stop
      endif
      goto 1
      end
```

Bibliography

[Jam94] F. James. RANLUX: A FORTRAN implementation of the high-quality pseudorandom number generator of Lüscher. *Computer Physics Communications*, 79:111 – 114, 1994.

[L9̈4] M. Lüscher. A portable high-quality random number generator for lattice field theory simulations. *Computer Physics Communications*, 79:100 – 110, 1994.