

The ANSWERS
Software Package

MCBEND

A Monte Carlo Program for
**General Radiation
Transport Solutions**

User Guide for Version 10

Chapter 7
An Introduction to the
Monte Carlo Method

CONTENTS

- 1 INTRODUCTION
- 2 ELEMENTS OF PARTICLE TRANSPORT
- 3 USE OF RANDOM NUMBERS
- 4 BASIC STATISTICS
- 5 A SIMPLE EXAMPLE OF THE MONTE CARLO METHOD
- 6 SAMPLING FROM A CONTINUOUS DISTRIBUTION
- 7 SAMPLING FROM A HISTOGRAM
- 8 BIASED SAMPLING
- 9 SOURCE SAMPLING
- 10 SCORING OF RESULTS
 - 10.1 Particle Flux
 - 10.2 Response Scoring
 - 10.3 Sensitivity Scoring
 - 10.4 Other Scored Items
 - 10.5 Error Estimates
- 11 GEOMETRIC MODELLING
- 12 VARIANCE REDUCTION
 - 12.1 Introduction
 - 12.2 Specifying Importances
 - 12.3 Implementation of Splitting
 - 12.4 Implementation of Russian Roulette
 - 12.5 Post Splitting Activity
 - 12.6 Effects of Imperfect Splitting
 - 12.7 Refinements in Variance Reduction
- 13 NUCLEAR DATA
- 14 MATERIAL DATA
- 15 ALBEDO DATA
- 16 CONTROL OF A MONTE CARLO CALCULATION

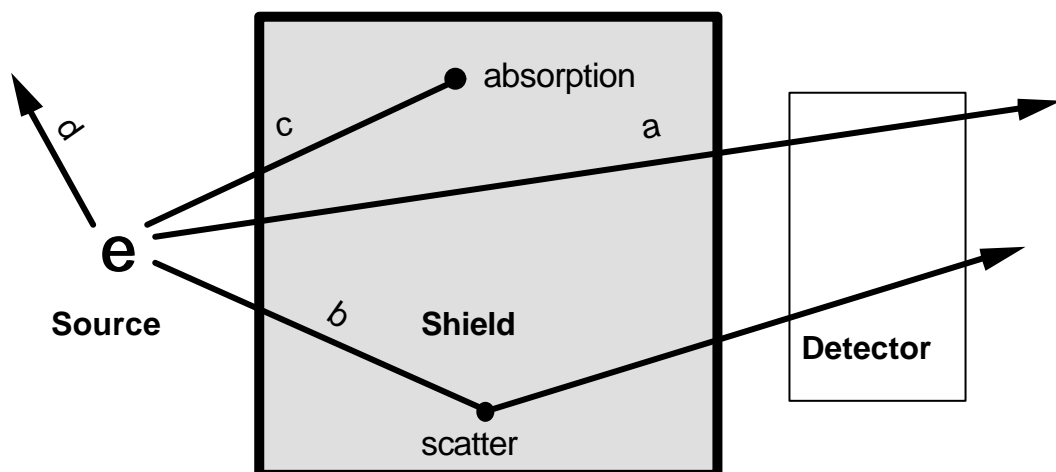
1 INTRODUCTION

This report describes the principal techniques used in the computer application of the Monte Carlo method to problems of radiation transport. It is specifically concerned with the code MCBEND which has been developed by Serco Assurance (formerly AEA Technology). Sections 2-8 describe the Monte Carlo method, introduce some of the terminology, summarise the statistical formulae used, and outline the techniques of taking random samples. Later sections describe how these techniques are applied in MCBEND to the processes of source sampling, particle tracking, result scoring and variance reduction.

The report is primarily written for new users of the code assuming they have little prior experience of Monte Carlo methods. Much of the material is drawn from the seminars and workshops which are presented by the ANSWERS Software service. The epithet 'Ladybird' has been applied to it - not inappropriately! It is not meant to be read from cover to cover but to serve as a reference on various topics.

The sections dealing with MCBEND specifically are not complete or comprehensive where other documentation exists or is planned. For example: the mechanics of Fractal Geometry and the creation of efficient models are covered in other ANSWERS documents. There are many reports which should be read after this one but, it is hoped, none which need precede it. The report does not concern itself with the input to the code - how data is entered; whence it is obtained; how MCBEND links to other codes.

2 ELEMENTS OF PARTICLE TRANSPORT



The above sketch shows the basic components of a typical problem in radiation shielding. Particles emitted from the source may:

- a) pass through the shield and reach the region of interest (the detector) without any interaction with the penetrated material
- b) suffer one or more scattering events in the shield material and reach the detector with reduced energy

- c) be absorbed in the shield
- d) escape from the system being modelled.

Combinations of these possibilities may also occur: scatter + escape, scatter + absorption etc.

It is possible to write down equations which describe the average behaviour of the particle population throughout the system. A completely general equation would have as its dependent variable the number of particles as a function of position, velocity and (possibly) time. Terms in the equation would include the addition of particles from the source, losses through absorption or escape, and terms to express the changes in velocity which occur at collisions. In its complete form the resulting equation (the Boltzman transport equation) cannot be solved analytically for practical problems. Methods have been developed for discretising the independent variables of space, velocity and time to give systems of equations which can be solved by numerical methods. These are classed as deterministic methods and usually involve many approximations in order to render them soluble within available computer resources. Deterministic methods include techniques such as Discrete Ordinates, Spherical Harmonics and Diffusion Theory (computer codes such as DOT, ANISN and SNAPSH) which make various simplifying assumptions. The equations generated may be solved by finite-difference or finite-element methods, often in geometries reduced to one or two dimensions.

An alternative method of treating the problem is to consider the fate of individual particles emitted by the source rather than the average behaviour of the population as a whole. This stochastic method is whimsically named the Monte Carlo technique because it involves many elements of gaming and dice rolling in its application. The Monte Carlo method is extremely close to the true, physical nature of the problem being solved. If a radioactive source is observed there is no way of predicting the energy and direction of the next emitted particle; one can only state the probabilities of these parameters having particular values. Similarly, when a particle is travelling through a material, one cannot predict when it will collide with the constituent nuclides or which one it will hit; only the probabilities are known.

Given a comprehensive set of nuclear data and infinite computer resources it would be possible, by the Monte Carlo technique, to reproduce in a computer model an exact simulation of the physical problem to be solved. In practice the known nuclear data contains uncertainties and can introduce systematic errors into the results - a problem shared by deterministic methods. Also, it is only possible to simulate the fate of a finite number of source particles which introduces statistical errors into the results. It is, however, possible to estimate the magnitude of the statistical error for a given Monte Carlo calculation; an important feature of the method.

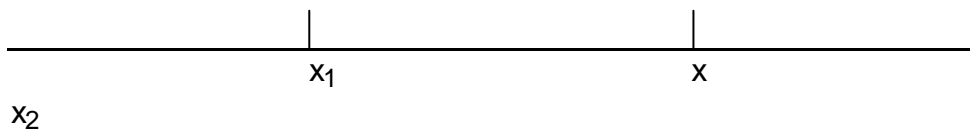
The essential steps taken in a Monte Carlo evaluation of a shielding problem are as follows:

- (i) Generate starting parameters (position, direction, energy) for a particle by random sampling from a specified source description.
- (ii) Track the particle within a geometric model of the shield system.
- (iii) Choose the position of a collision and its outcome from the modelled material distribution and a library of basic nuclear data.
- (iv) Record events in the particle history from which useful results (e.g. flux) may be estimated.
- (v) Repeat steps (i) to (iv) for a number of samples.
- (vi) Process the results to give particle fluxes etc. and evaluate the statistical uncertainties appropriate to the number of samples taken.

The techniques for performing these operations in a computer model are described in the following sections. Also described are methods of increasing the efficiency of the calculation by selectively following particle histories which make most contribution to the results of interest. Section 3 to Section 8 cover fairly general topics of the Monte Carlo method; specific application to the solution of radiation transport problems, with particular reference to the code MCBEND, begins in Section 9.

3 USE OF RANDOM NUMBERS

Monte Carlo methods make extensive use of random numbers to control the decision making when a physical event has a number of possible results. Consider the simple example of choosing a point along the x axis between the limits x_1 and x_2 .



If all values in the interval are equiprobable then a choice of x is given by:

$$x = x_1 + (x_2 - x_1) \mathbf{R}$$

where \mathbf{R} = Random number in the range 0.0 to 1.0

This represents a simple, linear scaling of a random number to apply it to an interval other than 0.0 to 1.0 More elaborate applications are possible for cases where the sampling is not uniform over the required interval. These extensions are described in later sections. The symbol \mathbf{R} will be used throughout this note to represent a random number:

$$0.0 < \mathbf{R} < 1.0$$

Most computers have available in their standard software a facility for generating random numbers. Pedantically these are pseudo-random numbers since they follow a definite sequence from a given set of starting parameters. Most packages have the following features:

- (i) a sequence of random numbers in the range 0.0 to 1.0 are generated one by one on request
- (ii) the sequence is effectively infinite for practical purposes
- (iii) the package can be initiated into a reproducible sequence by supplying one or more starting parameters (usually referred to as seeds)
- (iv) at any point in the sequence the seeds may be obtained which would permit re-entry into the sequence at that point.

Because of the extensive use of random numbers it is not possible to perform even the simplest of Monte Carlo calculations by hand and expect a computer to produce the same results. Two Monte Carlo computations with the same input data will not agree exactly unless initiated with the same random number seeds. This makes code verification and the provision of worked examples a rather difficult task. The usual solution to the problem is to evaluate the *expected* result and check that its difference from an actual result is within the statistical limits defined in the next section. The expected result is that which would be obtained if all probabilities were obeyed exactly in a batch of samples. For example: the *expected* result of 100 coin tosses is 50 heads and 50 tails.

4 BASIC STATISTICS

Because the Monte Carlo method is based on random sampling of events the results will be subject to statistical uncertainties. To clarify some of the analysis of later material, this section defines some of the terminology to be used and introduces some basic statistical formulae.

The results of a MCBEND calculation are obtained by recording the events in the life of each particle from its birth in the source to its death through capture or escape from the system of interest. Define the following:

SAMPLE = a particle emitted from the source
HISTORY = the sequence of events in the life of a sample
BATCH = total set of samples in a given calculation
SCORE = physically useful result derived from the histories of all samples in the batch (e.g.: flux)

A score contains contributions from all samples in the batch (some contributions may be zero) and in general the contribution will vary with sample. The mean contribution is given by:

$$\bar{X} = \frac{\sum_{i=1}^N X_i}{N}$$

X_i = Individual sample contribution
 N = Number of samples in the batch

A measure of the spread of individual contributions about the mean value is given by:

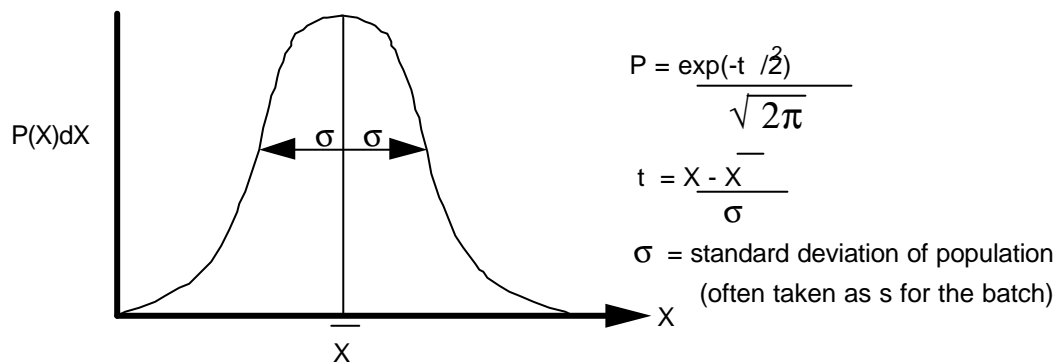
$$\begin{aligned}
 \text{variance, } s^2 &= \frac{\sum_{i=1}^N (X_i - \bar{X})^2}{N} \\
 &= \frac{\sum_{i=1}^N X_i^2}{N} - \bar{X}^2
 \end{aligned}$$

$$\text{standard deviation, } s = \sqrt{\text{variance}}$$

If the standard deviation is expressed as a percentage of the mean value we obtain the coefficient of variation:

$$\text{coeff. of variation, } v = 100 s / \bar{X}$$

The way in which individual results are distributed about the mean value may often (but not always!) be assumed to approximate a standard, statistical function called the normal distribution. This is sketched below:



The figure shows the probability $P(X)$ of a given result falling within a small interval dX about the value X when a sample is taken. The curve is symmetrical about the mean value of X which is the result with highest probability. The normal distribution has the following properties which are found by integrating the function between limits either side of the mean:

n Probability of a result being within
 n standard deviations of the mean

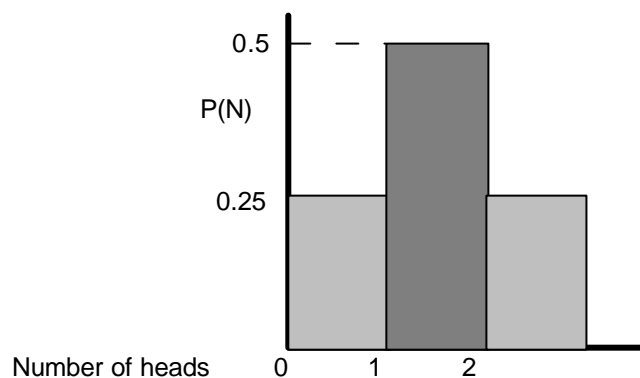
$$\text{i.e.: } P(\bar{X} - ns < X < \bar{X} + ns)$$

1 68.0%
 2 94.0%

national mean age is expected to lie within $\pm s/(N-1)$ of the single school result with a probability of 68%.

Returning to MCBEND: an estimate of the particle flux is given by the mean track length of particles passing through a given scoring volume. In order to evaluate an uncertainty for the flux the variation of this mean is required rather than the standard deviation of individual tracks. By analogy with the above: individual track lengths correspond to pupils, the batch of MCBEND samples is a school and the national population corresponds to the infinite number of possible particle histories. Details of flux estimators are given in later sections.

Another useful item of standard statistics is the binomial distribution which relates to events with an either/or outcome: eg coin tossing. For a batch of two coin tosses there are three possible results: 2 heads (probability 0.25); 1 head and 1 tail (probability 0.5); 2 tails (probability 0.25). The distribution of results is therefore :



This clearly differs from a normal distribution since it is a histogram - though it is symmetrical with a central peak. As the number of trials increases the histogram becomes progressively smoother and begins to approximate to the normal distribution; for 100 trials we would have 101 intervals along the x axis and very small steps. If p and q are the probabilities of the two results of a binary trial ($p + q = 1$) then the standard deviation of the outcome of N trials is:

$$s = \sqrt{Npq}$$

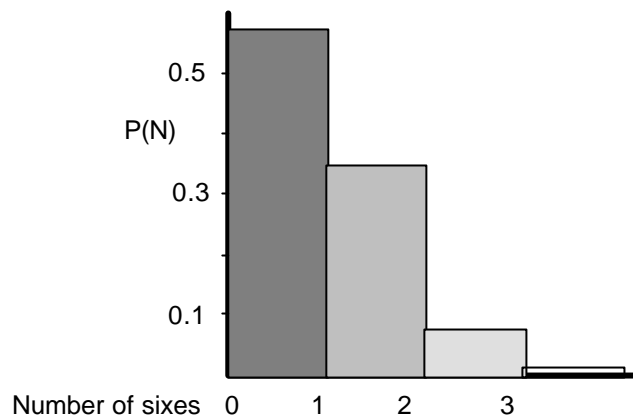
For coin tossing : $p = q = 0.5$

$s = 5$ for 100 trials

Assuming an approximately normal distribution for this batch size we expect, with 68% confidence, that 100 trials will produce between 45 and 55 heads.

The following example illustrates a case where p and q have differing values. If we roll a die, the probability of getting a six (p) is 1/6 and the probability of not getting a six (q) is 5/6. In three trials the probabilities of

getting 0, 1, 2, or 3 sixes are, respectively: 0.57870, 0.34722, 0.06944, 0.00463. This distribution is not symmetrical and is generally classified as skew .



As the number of die rolls increases to a large number (say $N > 1000$) the distribution peaks at $N/6$ so the skewness always remains. The shape of the peak does, however, begin to resemble the normal distribution. The greater the difference between p and q the greater the batch size required to reach a normal distribution around the peak.

Finally, there is another distribution (Poisson) which is useful for analysing events for which a non-occurrence has no significance. One cannot count how many times per week the site computer does not crash but may wish to analyse, from past records, the probability of failure. The Poisson distribution is also a useful substitute for the binomial distribution when p or q is very small.

For an example, consider the simultaneous facing of playing cards from two packs. What is the probability of facing an identical pair before the packs are exhausted? At each facing the card from pack A may be considered as a random choice of card. The probability of a card from pack B being the same is 1 in 52. The operation is repeated 52 times. Let Z be the expected, or average number of occurrences of an event (N.B. this is not the probability of an event). If an event has a 1 in M probability then in N trials the value of Z will be N/M . For our playing card example, therefore, $Z=1$. The Poisson distribution states that the probability of an event occurring n times is given by:

$$\begin{aligned} P(n) &= \frac{Z^n \exp(-Z)}{n!} \\ \text{variance} &= Z \end{aligned}$$

For the binomial distribution we have :

$$\begin{aligned} p &= 1/52 \\ q &= 51/52 \\ N &= 52 \\ P(n) &= {}^N C_n p^n q^{n-1} \\ &= \text{terms in the binomial expansion of } (p + q)^N \\ \text{variance} &= pqN \end{aligned}$$

The following results are obtained for the probabilities of n occurrences of identical cards being faced in one pass through the pair of packs:

n=	0	1	2	3	4	5	σ
binomial=	.3643	.3715	.1857	.06070	.01458	.00274	0.99
Poisson =	.3679	.3679	.1839	.06131	.01533	.00307	1.00

The similarity between the two distributions in this example is fairly obvious. Generally the Poisson distribution is easier to evaluate than the binomial. The figures show that betting on the occurrence of simultaneous cards is a good investment at even money!

5 A SIMPLE EXAMPLE OF THE MONTE CARLO METHOD

Consider the simple, non-nuclear problem of finding the area under a curve. To be perverse let the curve be defined by:

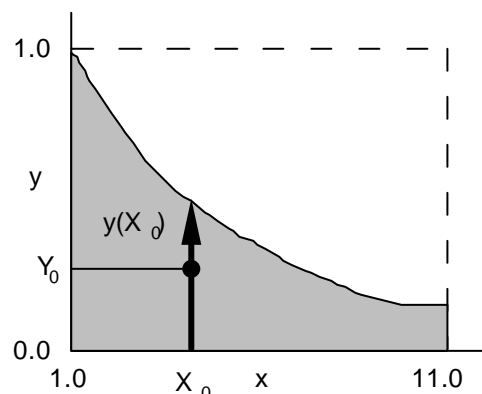
$$y(x) = \cos(\sqrt{\ln(x)}) \quad 1 < x < 11$$

The equivalent of the Boltzman transport equation would be:

$$\text{Area} = \int_1^{11} \cos(\sqrt{\ln(x)}) dx$$

This equation is simple and complete but (by me at least) insoluble! An example of a deterministic solution of this problem would be the use of Simpson's rule for integration; the accuracy of the result would depend on the number of mesh intervals chosen along the x axis.

Let us now consider how we could use a Monte Carlo method to solve the problem. The function to be integrated has the following approximate shape :



Suppose the problem is bounded by the rectangle: $1 < x < 11$ and $0 < y < 1$

The area of the rectangle is 10 units. Now choose a random point within the rectangle:

$$\begin{aligned} X_0 &= 1.0 + 10.0 R_1 \\ Y_0 &= R_2 \end{aligned}$$

where R_1, R_2 = random numbers

The point X_0, Y_0 is under the curve if $Y_0 < y(X_0)$

If we evaluate N points and n are found to be under the curve then an estimate of the area under the curve is given by :

$$\text{Area} = \frac{n}{N} \times \text{area of rectangle} = \frac{10n}{N} \text{ units}$$

The accuracy of this answer obviously depends on the number of samples taken. An estimate of the uncertainty may be determined from statistical formulae and the observed probability of scoring a point under the curve. The uncertainty itself will have an uncertainty since it is evaluated from the results of the trials; this second-order uncertainty is not usually required.

Let p = probability of scoring a point under the curve $\sim n/N$

Then the standard deviation of n assuming a binomial distribution (Section 4) is given by:

$$S_n = \sqrt{Np(1-p)} \quad \sim \sqrt{\frac{n(N-n)}{N}}$$

The standard deviation of the area estimate is given by:

$$S_A = \frac{S_n}{N} \times \text{area of rectangle}$$

The coefficient of variation for the area estimate is :

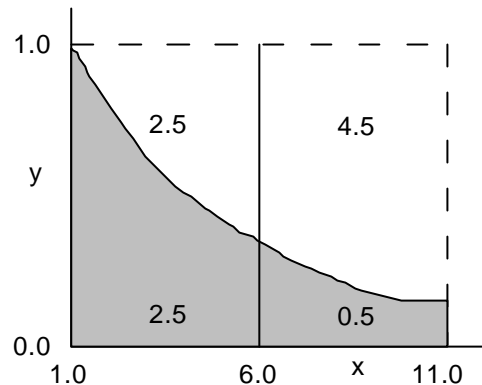
$$V_A = \frac{100 S_n}{n}$$

Suppose we have a series of curves which progressively fill more and more of the enclosing 10 unit rectangle. The expected results for batches of 100 samples are :

%of rectangle filled	expected n	S_n	Area $\pm V_A$
10	10	3.0	1.0 \pm 33%
30	30	4.58	3.0 \pm 15%
50	50	5.0	5.0 \pm 10%
70	70	4.58	7.0 \pm 6%
90	90	3.0	9.0 \pm 3%

The more the curve fills the enveloping rectangle the higher the probability, p , of sampling a point below it. For a given batch of samples (N) this will lead to a higher value of n and a lower coefficient of variation for the area estimate. Another important observation is that the uncertainty in the area estimate varies in inverse proportion to \sqrt{N} . Thus to halve the uncertainty the batch size must be quadrupled.

Consider now an elaboration on this simple method. Suppose the enveloping rectangle of our example function is bisected and that we know the areas marked in the following sketch.



The left half of the figure contains most of the area under the curve and the half rectangle is more completely filled than that on the right. We might get a better estimate of the area if we take more samples in the left half than in the right. Suppose we have a way of biasing the sampling so that (in a batch of 100) 80 occur in the left half, 20 in the right. The following results could be obtained :

Left half below curve	=	40	
Right half below curve	=	2	
Total below	=	42	
Area below	=	$(n/N) \times \text{area of total rectangle}$	
	=	4.2	= WRONG

This is a consequence of blatant cheating without any compensating measures; the result is deservedly incorrect. Let us try again, this time sampling 80 'little' points in the left half and 20 'big' points in the right. Since points cannot really have a size the more convincing term *weight* is usually used in Monte Carlo terminology. So that the same total weight is sampled in each half the values chosen should counter the bias in sample distribution: a ratio of 1 to 4 is required. Let us choose, arbitrarily:

W_L	= weight of a sample in the left half	= 0.5
W_R	= weight of a sample in the right half	= 2.0

The area under the curve is now given by :

$$\text{area} = \frac{\text{total weight sampled below curve} \times \text{area of total rectangle}}{\text{total weight sampled}}$$

Expected results for a batch of 100 samples in the chosen example are :

total weight started in left half	= 80x0.5	= 40
total weight started in right half	= 20x2.0	= 40
weight started below curve in left half		= 20
weight started below curve in right half		= 4
area under curve	= $\frac{20 + 4}{40 + 40}$ x area of rectangle	
	= 3.0	= CORRECT

Now consider what has happened to the standard deviation :

	<u>biased</u>	
<u>unbiased</u>		
samples scoring under the curve, n	=(80x0.5+20x0.1)	
	=42	30
probability of scoring under the curve	=0.42	0.30
S_n	=4.94	4.58
V_A	=11.8%	15.3%

One more point may be illustrated using this simple example: that is the effect of excessive biasing. Suppose the left:right weighting is 99:1 and that only 100 samples are taken. The average number of samples taken in the right half is 1. Assuming a Poisson distribution (Section 4) the probability of taking 0,1,2 or 3 samples in the right half are, respectively: 0.3679, 0.3679, 0.1839 and 0.0613. For each of these possibilities the binomial distribution may be used to evaluate the probable numbers above and below the curve in the right half. The large number of points (100 to 97) sampled in the left half may be assumed to be distributed 50:50 above and below the curve. The following results may be obtained for the distribution of points and the estimate of the area under the curve:

probability of result	points above curve on right	points below curve on right	estimate of area
0.3679	0	0	5.0
0.3311	1	0	2.5
0.0368	0	1	7.5
0.1490	2	0	1.66
0.0331	1	1	5.0
0.0018	0	2	8.34
0.0447	3	0	1.23
0.0149	2	1	3.74
0.0017	1	2	6.26
<u>0.0005</u>	0	3	8.77
<u>0.9815</u>			

All of the above results have a standard deviation of about 0.5 on the 'true' area value of 3.0 units. It is clear that some possibilities are many standard deviations in error and that the distribution is not normal. The value of the area estimate is strongly influenced by the location of the few, high weight points sampled in the right half of the range. The second and third rows of the above table, for example, give area estimates of 2.5 and 7.5 depending on whether the single point sampled on the right falls above or below the curve. If many batches of 100 samples were taken and each generated one sample on the right then the probabilities of it falling above or below the curve are 0.9 and 0.1 respectively. A mean result would therefore be $0.9 \times 2.5 + 0.1 \times 7.5 = 3.0$ i.e. the correct result. If a large number of samples are taken (say 1000 or even 10000) a good estimate of the area under the curve would be obtained even with the 99:1 biasing.

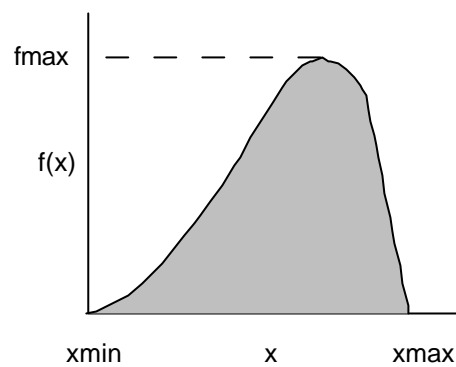
This fairly trivial example has been laboured to illustrate the following general, and very significant points :

- 1) If biasing is correctly compensated by weighting then the expected result is not changed. In practice the biasing is normally controlled by the weighting function so inconsistencies are eliminated.
- 2) Biased sampling can reduce the uncertainty of a result for a given number of samples.
- 3) Biased sampling could equally well increase the uncertainty of a result if applied in the wrong direction. Reversing the left-right (80:20) bias in the above example would give a coefficient of variation of 21.3%
- 4) The absolute values of the bias weights are not significant; it is the ratio of the weights which matters.
- 5) If excessive biasing is used then a large number of samples must be taken to ensure adequate sampling of rare, high weight events.
- 6) In the above example the solution to the problem was pre-supposed in order to evaluate illustrative, expected results. Biasing could have been introduced on the basis of a very approximate idea of the shape of the curve. A 60:40 bias might be applied on the grounds that: 'the left half looks a bit more significant than the right'; a 90:10 bias might be applied on the grounds that : 'the left half looks a lot more significant than the right'. This is very typical of a Monte Carlo calculation: the more idea one has about the solution the more efficiently one can bias the sampling. Biasing can be optimised if the answer is known exactly - but then, why bother with the Monte Carlo calculation anyway?

6

SAMPLING FROM A CONTINUOUS DISTRIBUTION

During the course of a Monte Carlo calculation it is frequently necessary to sample some physical event, the probability of which is described by a known mathematical function. Examples include the distance to the next collision and the energy of a neutron emitted from a fission spectrum. Let x be the physical quantity to be selected and $f(x)$ the function which describes the frequency of occurrence. A typical form of this function might be :



The probability of x having a value near the middle of its possible range x_{min} - x_{max} is indicated by this arbitrary example to be higher than the probabilities of its value lying near one of the extremes. In choosing values for x we effectively need to sample points uniformly from the area under the above curve. One method of doing this is to use the rejection technique which follows the following steps:

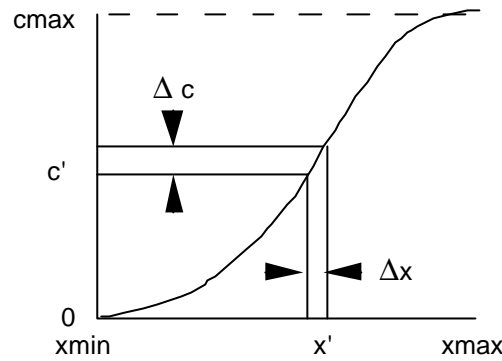
- 1) enclose the curve in a rectangle bounded by x_{min} - x_{max} in the x direction and of height f_{max} .
- 2) use two random numbers to select the co-ordinates of a point x,y within the rectangle.
- 3) reject the point and try another if it does not lie under the curve i.e. if $y > f(x)$
- 4) accept the value of x if the point does lie under the curve.

Over a large number of samples this technique will yield a set of values of x with the required distribution. It does, however, require two random numbers per trial and many trials will be required if the area under the curve is a small fraction of the area bounded by the rectangle. If $f(x)$ approaches zero asymptotically at either end of the range of x then a finite limit must be imposed on x with the sacrifice of some accuracy. This technique is of limited use in an efficient Monte Carlo code but is described here for completeness. Its use in MCBEND is mainly confined to sampling source points within complex source bodies of difficult shape.

A superior technique is to form a cumulative distribution function from $f(x)$ defined by:

$$\begin{aligned}
 c(x) &= \int_{x_{min}}^x f(x) dx \\
 &= \text{area under curve up to the value } x
 \end{aligned}$$

For a curve with the shape of the chosen example the form of $c(x)$ is :



Let c' be a point sampled uniformly from the range 0 - c_{max}
 Let Δc be a small interval about c
 Let x' and Δx be the corresponding point and interval on the x axis

$$\Delta c = \frac{dc}{dx} \Delta x = f(x') \Delta x$$

Hence the curve $c(x)$ relates a uniformly-sampled value c' to a value of x' sampled at the required frequency.

$$c(x') = c' \quad x' = c^{-1}(c')$$

If R is a random number in the range 0 - 1.0 then:

$$c' = R \cdot c_{max} \quad x' = c^{-1}(R \cdot c_{max})$$

Usually the functions described are normalised by dividing by the total area under the curve. This converts $f(x)$ into a probability function and gives a unit value for c_{max} . If the normalised function is denoted by c_p then values of x , sampled with the correct distribution, are found by repeated evaluations of:

$$x = c_p^{-1}(R)$$

This technique is applicable whenever the function $f(x)$ can be integrated analytically. It permits the sampling of a value x from a single random number with no rejections. It also copes with asymptotic extremities of $f(x)$.

An example, practical application of this technique is the evaluation of the distance to the next collision site of a moving particle. For a given energy and material the total cross-section μ is known and the probability of travelling a distance x without collision is:

$$p = \exp(-\mu x)$$

The reciprocal of μ is often referred to as the mean-free-path in the material and the product μx as b , the number of mean-free-paths to the next collision. We may therefore write:

$$p(b) = \exp(-b)$$

Integrating this function between the limits 0 and b gives the cumulative probability function:

$$c(b) = 1 - \exp(-b)$$

Values of b are therefore given by:

$$\begin{aligned} b &= c^{-1}(R) \\ &= -\ln(1-R) \quad (= -\ln(R) \text{ since } 1-R \equiv R) \end{aligned}$$

Note that this example requires no normalisation since the maximum value of c(b) approaches unity as b approaches its upper limit of infinity.

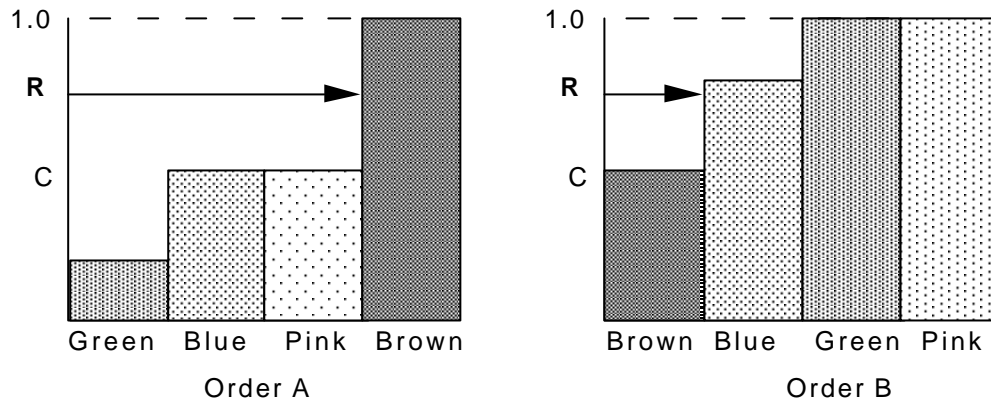
7 SAMPLING FROM A HISTOGRAM DISTRIBUTION

Many parameters chosen in a Monte Carlo calculation are limited to having one of a finite set of values. For example: if a collision occurs in water then the collision must be with either hydrogen or oxygen; a collision with a particular nuclide may result in an absorption, an elastic scatter or an inelastic scatter. The relative probabilities of the various events are known from nuclear data; a method is required of choosing one event from the possibilities on any single occasion.

Suppose we have a group of 100 people of which 50 have brown eyes, 30 have blue eyes and 20 have green eyes. The group includes no albinos so that if one person is selected at random the probability of their eyes having one of the following colours is:

green	0.2
blue	0.3
pink	0.0
brown	0.5

If the possibilities are arranged in an arbitrary order then the cumulative probability function, C, becomes a series of steps with height proportional to the probability of each event. The sketches below show two of the twenty four possible arrangements for the data given above.



A selection may be made by plotting a random number **R** on the C axis and tracing across to the step function. If the tread which next exceeds the value of **R** is identified then this corresponds to the event to be selected. The probability of selecting a particular event by this process is equal to the height of the riser in the step function which is identical to the probability of the physical event.

For a single trial the selection is dependent on the ordering of the events along the x axis. Sketch A above shows the selection of brown; sketch B (for the same random number) shows the selection of blue. Over a large number of trials this dependence on order is eliminated. Note that the tread corresponding to pink in the example has a zero height riser and (correctly) can never be chosen by this process.

This technique is simple to implement numerically. The heights of the steps in the cumulative are an increasing sequence of values in the range 0 to 1 inclusive. For a given choice of **R** the first number in the sequence which exceeds **R** corresponds to the event to be selected.

In a practical application of this technique the number of steps may become very large. Consider, for example a cuboid volume source subdivided in each dimension into ten intervals. If the source is defined in twenty energy groups then the selection of parameters for a source particle in MCBEND has to be made from 20000 possibilities. The height of each step in the cumulative distribution will be proportional to the volume of the mesh cell it represents and the source strength in the mesh in the relevant energy group. The ordering of the X, Y, Z, Energy combinations in the cumulative must be such that, having chosen a combination, the individual subscripts can be retrieved. In MCBEND the choice of a random number is followed by a binary search* of the step heights so that, for 20000 values, the relevant step height should be located within a maximum of 15 attempts ($2^{15} > 20000$). The ordering of the combinations usually follows FORTRAN subscript conventions.

Let NX,NY,NZ,NE be the number of mesh intervals in each dimension
 Let IX, IY, IZ, IE be the subscripts of an individual bin.

A single subscript within the system is generated by:

$$M = IX + (IY-1) NX + (IZ-1) NX.NY + (IE-1) NX.NY.NZ$$

The random number and cumulative effectively select a value of M which may be unravelled using the fixed values NX,NY,NZ,NE and the rules of integer arithmetic.

* A binary search is one in which the range to be searched is progressively halved. Take, for example, the game of guessing a number in the range 1- 128:

Suppose the number is	x=19	1	≤ x ≤	128
Is it less than 64?	Yes	1	≤ x <	64
Is it less than 32?	Yes	1	≤ x <	32
Is it less than 16?	No	16	≤ x <	32
Is it less than 24?	Yes	16	≤ x <	24
Is it less than 20?	Yes	16	≤ x <	20
Is it less than 18?	No	18	≤ x <	20
Is it less than 19?	No	19	≤ x <	19
	∴			x = 19

Obviously some numbers will be found quicker than others by this technique. The range to be searched does not have to be a power of two provided systematic rounding is applied when the range is narrowed to an odd number. In general the maximum number of guesses is n where 2ⁿ is the first power of two which is greater than or equal to the number of possible values.

8 BIASED SAMPLING

Suppose we have a cuboid volume source with a unit cross-sectional area in the YZ plane and subdivided into three regions in the X dimension. The size of the subdivisions and the source strength per unit volume in each region are shown in the following sketch.

Region i	=	1	2	3
Source S	=	0.5	0.6	1.0
Volume V	=	4.0	5.0	5.0
Total source S.V	=	2.0	3.0	5.0
X	=	0.0	4.0	9.0
				14.0

The total source in each region is the source strength S (particles/cc.sec) multiplied by the region volume V. The probability of a particle being born in region i is :

$$p_i = \frac{S_i V_i}{\sum_i S_i V_i} = \begin{matrix} 0.2 & 0.3 & 0.5 \end{matrix}$$

A cumulative probability may be formed from these values and sampled by the method described in Section 7. In both the physical situation and a Monte Carlo model the expected distribution from a batch of 240 samples would be 48, 72, 120 for regions 1, 2, 3 respectively.

Suppose now we decide that particles born in region 3 are more likely to make a contribution to a required score than those in region 2 and that region 2 is more important than region 1. Let this importance be quantified by a parameter I to which, in this example, we will assign values 1.0, 2.0, 8.0 for the three regions. A revised sampling probability may be evaluated for the regions which includes the importance factor:

$$S V I = \begin{matrix} 2.0 & 6.0 & 40.0 \end{matrix}$$

$$p_i' = \frac{S_i V_i I_i}{\sum_i S_i V_i I_i} = \begin{matrix} 2/48 & 6/48 & 40/48 \\ = & 0.042 & 0.125 & 0.833 \end{matrix}$$

A new cumulative probability formed from these values would give an expected distribution of source particles from a 240 sample batch of 10; 30; 200 for regions 1, 2 and 3 respectively. To compensate for this distortion of the sampling we need to assign weights to the source particles so that we are, in effect, sampling a few 'big' particles from region 1 and many 'small' particles from region 3. If the compensating weights are assigned values W we require, for each region, that:

$$p_i' W_i = p_i$$

$$W_i = p_i / p_i' = \frac{S_i V_i}{\sum_i S_i V_i} \times \frac{\sum_i S_i V_i I_i}{S_i V_i I_i} = \frac{\sum_i S_i V_i I_i}{\sum_i S_i V_i} \times \frac{1}{I_i}$$

Thus the weight is inversely proportional to the importance. In the example we have values of W= 4.8, 2.4, 0.6 for the three regions. Each particle started in the biased case in region 3 is worth only 0.6 of an unbiased sample in that region.

In the case of the 240 sample batch, with bias, the expected, total weight started in each region is:

$$240 \times p_i' W_i = \begin{matrix} 48.0 & 72.0 & 120.0 \end{matrix}$$

These figures match the sampling distribution for the unbiased case. Note that scaling of all the importances by a constant factor would not change the values of W ; importance values are purely relative.

The same principles could be used for continuous probability and weighting functions by replacing the summations in the above equations by integrals.

Let $f(x)$ be a function describing the physical probability of sampling a value of x

Let $g(x)$ be a function describing the importance we assign at x

Let x_1 and x_2 be the range from which x is to be sampled

Then the revised (biased) probability of sampling a value within the small interval dx about x is given by:

$$p'(x) dx = \frac{f(x)g(x)dx}{\int_{x_1}^{x_2} f(x)g(x)dx}$$

When a value of x is sampled from this biased distribution (using the method described in Section 6) then its weight should be assigned a value:

$$w(x) = \frac{\int_{x_1}^{x_2} f(x)g(x)dx}{\int_{x_1}^{x_2} f(x)dx} \times \frac{1}{g(x)}$$

An example of this situation is the use of a fission spectrum in a neutron source weighted by a function proportional to neutron energy.

A situation may occur when either the weighting function $g(x)$ or the sampling probability $f(x)$ is a histogram and the other is a continuous function. In this case the continuous function must be integrated over each interval of the histogram and an interval sampled from the resulting step function cumulative. If $f(x)$ is the continuous function then a specific value of x may be obtained by sampling within the chosen interval according to the original, unbiased distribution function. The weight assigned will be derived from the importance of the chosen interval. An example of this situation is the weighting of a fission spectrum with a set of importances defined in an energy group scheme.

If the weighting function is continuous then the weight assigned to a value chosen in a given interval could be based on the integrated interval weight or the 'spot' weight at the chosen value of x . There is little advantage in the latter option. An example of this situation is the weighting of a histogram source spectrum by a function proportional to energy.

The function of the source routines in MCBEND is to set the initial parameters of each new sample that is to be tracked. The values required are:

- (i) The position of the source point defined by its X, Y, Z co-ordinates
- (ii) The initial direction of the source particle defined by its direction cosines UT, VT, WT
- (iii) The energy of the source particle (E MeV)
- (iv) The initial weight of the particle (W)
- (v) The nature of the particle (neutron, gamma, charged particle)
- (vi) The speed of the particle (V) for time-dependent cases
- (vii) The time of birth of the particle (T) for time-dependent cases with a shaped pulse source.

The initial weight of the particle contains a normalisation factor which relates the results obtained in the MCBEND calculation to the physical system being modelled. This is usually the total source (particles/sec) in the model. An interpretation of this parameter is the number of physical particles represented by one MCBEND sample. If biased source sampling is used then the value is modified for individual samples to compensate for the bias; some MCBEND samples then represent more physical particles than others. The speed of the particle is derived from the energy and nature of the particle. The value of T (if relevant) is sampled from specified pulse shape data.

The remaining parameters may be grouped into three sets: space; energy; angle. In general the parameters in one of these sets are sampled independently from those in other sets which implies, for example, that the variation of source strength with energy is independent of the spatial location. In MCBEND certain interdependencies are allowed as follows:

- a) SPACE-ENERGY interdependence: the source variation with energy depends on spatial position but the angular variation is independent of both position and energy.
- b) ENERGY-ANGLE interdependence: the angular variation of the source depends on energy but the relationship is constant at all positions.
- c) SPACE-ENERGY-ANGLE for angular current sources.

These relationships apply to both the source strength and any weighting functions which may be imposed. When the spatial variation is independent of angle and energy the source is said to be separable.

There are three sets of source routines in MCBEND. The Unified Source describes the source using a set of freely oriented bodies similar to those used in the definition of the problem geometry (FG). Surface, point and line sources can be modelled and source bodies can be subdivided, as required. Specification of different source spectra is straightforward and

a number of built-in spectra are available. Automatic source weighting can be provided by the importance map. The Simple Source uses an XYZ or R-θ-Z geometry system with orthogonal meshes and allows a range of weighting functions and energy spectra. The Complex Source allows a geometrically flexible source definition but supports fewer weighting options than the simple source.

A volume source in MCBEND may be subdivided into small volume elements in each of which the source strength may be considered to be constant. A weighting function (importance) may also be specified as values for each elemental volume. The choice of element in which to start a given source particle is made by sampling from a cumulative probability function as described in Section 7. Having selected an elemental volume the co-ordinates of a source point are selected by uniform sampling within it. For an XYZ orthogonal mesh system each co-ordinate may be chosen independently. E.g.:

$$\begin{aligned} x &= x_1 + \mathbf{R} (x_2 - x_1) \\ x_1, x_2 &= \text{limiting co-ordinates of the mesh in the x dimension.} \end{aligned}$$

In cylindrical geometry the radial position of the point must be sampled from a probability function given by:

$$p(r) dr = 2 r dr / (r_2^2 - r_1^2)$$

Using the method described in Section 6 gives:

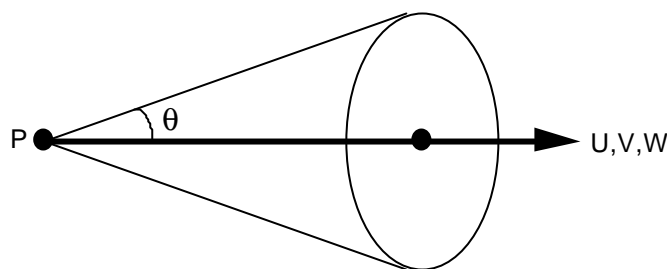
$$r = \sqrt{r_1^2 + \mathbf{R} (r_2^2 - r_1^2)}$$

A value of the azimuthal angle and the height within the cylinder may be sampled uniformly within appropriate limits followed by conversion to X, Y, Z source point co-ordinates. These principles are readily extended to freely-orientated cuboid and cylindrical sources by transforming axes. For more elaborate mesh volume shapes the rejection technique described in Section 6 may be used. The source shapes corresponding to volumes with some degenerate dimensions are also available in MCBEND e.g. planes, lines and single points.

The energy variation of the source strength and its associated weighting are usually separable from the spatial distribution. If described by a histogram of source strengths in each of a set of energy groups it may be sampled by the technique of Section 7 to choose a group number followed by uniform sampling within the selected group to give a specific energy value. For continuous distributions (e.g. a fission spectrum) the technique of Section 6 may be applied.

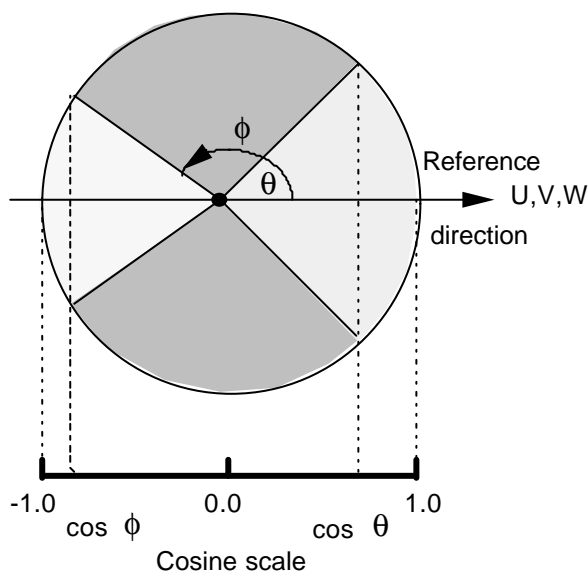
If the energy variation is dependent on position then the position and energy group are sampled from a four-dimensional probability table by a simple extension of the position sampling described above.

Most physical sources emit particles isotropically so that the directional parameters U, V, W may be sampled uniformly within their individual ranges of -1.0 to 1.0 . In some cases, however, it may be desirable to bias the sampling to produce more samples in certain directions or to relate the direction of emission to the particle energy. In such cases an angular quadrature is required which is formed by defining conical surfaces about a reference axis. A simple example of a single angular boundary defining two angular intervals is sketched below:



U, V, W are the direction cosines of a vector defining the reference direction. The cone with half-angle θ defines a surface on either side of which a different importance may be specified for particles emitted from a point P at the apex of the cone. A corresponding cone is used for any chosen location of P.

Any number of such surfaces may be defined and, in MCBEND, the angles are defined by their cosines to give a quadrature defined on a scale from -1.0 to 1.0 . An example with three boundaries (including a plane normal to the reference direction) is shown below.



The reference direction U, V, W may be defined as a fixed vector or derived in the code as the line joining the sampled source point P to a chosen reference point anywhere in the problem space. Angular weighting is usually used to bias the emission of particles from the source towards a detector region of interest. Care should be exercised in its use to avoid the consequences of excess biasing described in Section 5. It

is, however, acceptable to use importances of 0.0 in some intervals to prevent emission in that direction when this is physically meaningful.

10 SCORING OF RESULTS

10.1 Particle Flux

The function of the scoring routines in MCBEND is to derive physically meaningful results from the random walk of particles through the problem geometry. The primary quantity to be evaluated is the particle flux. This is formally defined as the number of particles threading a sphere of unit cross-sectional area at the point of interest. In principle one could position imaginary, unit spheres at positions of interest and count the number of particles which pass through. In practice this would not be very helpful: in a large problem the unit sphere might be so small that few particle tracks would pass through it; in an absorbing medium some tracks would not pass directly through the sphere; the mean flux in a distinctly non-spherical region may be required. Instead of such a direct approach the technique in Monte Carlo codes is to use devices known as estimators to evaluate the flux. The simplest form is probably the collision density estimator. This is based on the fact that the number of collisions per unit volume in unit time (C) is given by the expression:

$$\begin{aligned} C &= \mu \phi \\ \phi &= \text{particle flux n/cm}^2 \text{ sec.} \\ \mu &= \text{total macroscopic cross section of the material.} \end{aligned}$$

If the number of collisions (M) occurring in a specified region of the problem with volume V is monitored in a Monte Carlo calculation then ϕ may be evaluated as:

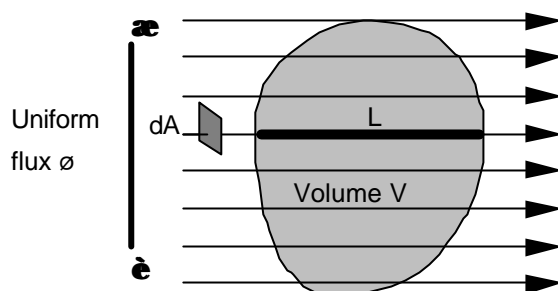
$$\phi = \frac{M}{\mu V}$$

In practice the MCBEND samples carry a weight W which is assigned in the source routines (see Section 9) and possibly modified during the tracking process (Section 12). This is a measure of the number of physical particles represented by a given sample. If a total batch of N samples are taken then the collision density estimator for particle flux becomes:

$$\phi = \frac{\sum_N (W \div \mu)}{VN}$$

If the scoring volume region includes subdivisions of differing materials and the flux is scored in energy groups then the estimator is still valid provided that the value of μ is appropriate for the energy value and material at the point of collision. This estimator is not very satisfactory in materials of low cross-section in which few collisions occur and does not produce any results at all in void regions.

An alternative is provided by the *track length* estimator which may be derived as follows:



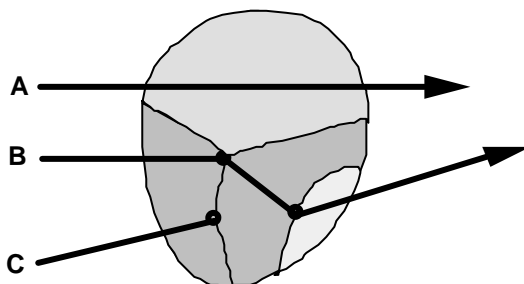
The sketch shows a uniform, mono-direction flux of particles ϕ incident on an arbitrarily shaped volume V . An elemental area dA is normal to the particle direction. L is the length of the particle track within the volume for particles passing through dA .

In unit time the number of particles passing through dA is ϕdA and the total length of particle tracks passing through dA is $\phi L dA$. Integrating the term $L dA$ over the entire area formed by projecting the volume on a plane normal to the particle direction simply gives a value V . The corresponding integral of $\phi L dA$ gives the total length of particle tracks in the volume; this may be accumulated during the course of a Monte Carlo calculation. Thus we have:

$$\begin{aligned} \text{Total track length in volume} &= \phi V \\ \phi &= \text{Total track length} / V \end{aligned}$$

So far the reasoning has considered a uniform flux of mono-directional particles. The same result would, however, be obtained if particles were travelling in some other direction or, more generally, for particles travelling in any arbitrary mixture of directions. It has also been assumed that all particle tracks pass straight through the volume but in practice there will be scattering or absorption of some particles entering any scoring volume containing a material.

Some examples of interrupted particle tracks are shown below:



Track A passes through the volume without interaction with the enclosed material; track B suffers two collisions with a change of direction at each;

track C is terminated within the volume by a collision. It is possible to subdivide the volume as shown so that each subdivision contains only complete, straight-line track segments.

Let V_i = volume of the i'th subdivision
 t_i = sum of the track lengths in the i'th subdivision
 ϕ_i = mean flux in the i'th subdivision
 V = total volume

Then the mean flux in the entire volume (ϕ) =
$$\frac{\sum_i \phi_i V_i}{V}$$

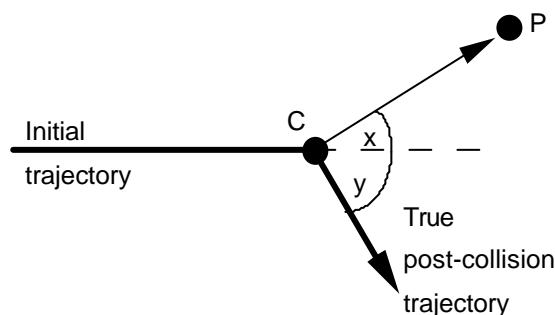
$$= \frac{\sum_i t_i}{V}$$

As in the case of the collision density estimator, each track carries a weight W which is the number of physical particles represented by a given MCBEND sample. The complete form of the track length estimator for a batch of N samples with t = track length recorded for any one sample is:

$$\phi = \frac{\sum_N tW}{VN}$$

The track length estimator of flux is therefore the mean weighted track length per unit volume. It is generally better than the collision density estimator and is valid in scoring regions containing no material.

Both the collision density and track length estimators provide values of the mean flux in user defined scoring volumes. They rely on particles passing through those regions during the particle histories. If detail is required of flux variation in space then small regions must be defined which will generally reduce the statistical accuracy of the flux estimates. A third form of estimator is available which will evaluate the flux at specific points in the problem. This is the point estimator which uses the following mechanism:



A particle has a collision at point C. The normal Monte Carlo processing of the collision process will select a new particle trajectory at a new energy by random sampling of nuclear data. The chosen angle of scatter

is y . If P is a point at which point estimation is required then the angle of deviation required to scatter the particle to the detector point is x . It is possible to interrogate the nuclear data to determine the probability per unit solid angle of scattering through an angle x and to find the post-collision energy which would result. The probability of reaching the point P without further collision is simply $\exp(-b)$ where b is the total number of mean-free-paths at the post-collision energy on the flight from C to P .

In MCBEND: when a collision occurs the parameters of the true post-collision trajectory are temporarily saved and the probabilities of reaching each requested detector point are calculated. The value of b is obtained by ray tracing through the problem geometry along the path CP . The point flux estimator therefore takes the form:

$$\begin{aligned} \phi &= \sum (p(x) \exp(-b) W/r^2) / N \\ p(x) &= \text{probability per steradian of scattering through angle } x \\ b &= \text{number of mean free paths from collision to detector} \\ r &= \text{geometrical distance from collision site to detector} \\ W &= \text{particle weight at the time of the collision} \\ N &= \text{total samples in the batch} \\ \Sigma &= \text{sum over all collisions for all particles.} \end{aligned}$$

The point estimator clearly has the potential for being very time consuming! In a large problem there is much work to be done in evaluating b for collision sites remote from a given detector point and the resulting contribution would probably (though not necessarily) be small. The frequency of collision in some materials at certain energies can be very high. At present no biasing is included in MCBEND to improve the efficiency of this estimator. It also suffers from a potential singularity when the collision site is very close to the detector.

In the above description of estimators the quantity ϕ has been referenced in a rather vague form. In practice the scoring of particle flux would normally be done in some energy group scheme so that contributions to the flux would be collected in separate 'bins' for each group.

10.2 Response Scoring

It is possible to fold into the flux evaluation a number of response cross-sections to give, by any of the three estimators, an evaluation of reaction-rates. Cross-sections for responses are defined as a histogram: a set of energy intervals are defined and the mean cross-section in each interval specified. Within each interval the response cross-section is considered constant but in most calculations (not multigroup - see Section 13) any degree of finesse may be used to specify rapidly changing cross-sections and an appropriate scheme may be used for each individual response. MCBEND picks the appropriate cross-section based on the particle energy at the time a contribution is made to the response scores so the results are not dependent on the energy group scheme used for flux scoring.

10.3 Sensitivity Scoring

The results of any calculation can only be as good as the nuclear data used to evaluate it. An important feature of MCBEND is the ability to calculate the sensitivity of the results to the cross-sections in the nuclear library data. The first-order sensitivity takes the form:

$$\text{Sensitivity} = \frac{c \, dR}{R \, dc}$$

R = a scored result
 c = cross-section in the data library.

The result R may be the flux or set of responses determined by collision density or track-length estimation. The cross-section c may be a partial cross-section of any nuclide in a chosen set of energy groups. Any number of combinations of cross-sections may be defined each of which takes the form:

{combination of nuclear reactions} {nuclide identifier} {material number}

Example combinations include:

- elastic scattering in iron in any material
- total cross-section of hydrogen in material 3
- absorption cross-section of any(all) nuclides in material 2
- total cross-section of any (all) nuclides in material 2

The sensitivity effectively gives the fractional change in a given result which would result from a (small) fractional change in a nominated piece of basic data.

For example: a sensitivity of 0.1 of a particular response to a particular cross-section indicates that a 5.0% change in that cross-section would give a 0.5% change in the response.

10.4 Other Scored Items

The contributions made to the response functions from particles in each of the flux scoring groups may be recorded. This is a useful diagnostic tool for checking that all significant parts of the particle energy range have been sampled.

For gamma-ray problems the energy deposited (with or without the effects of secondary electron transport) may be scored by accumulating the energy loss at each collision site. For cases including a treatment of electron/positron transport the charge deposition may be recorded.

At points of special interest some extra statistical data may optionally be presented to show, for example, the spread of the particle weights arriving in a given scoring region .

Some of the possibilities listed above can produce copious volumes of output and incur a penalty in the speed of program execution. MCBEND allows the user to limit the evaluation of each scored quantity to particular regions in the problem.

10.5 Error Estimates

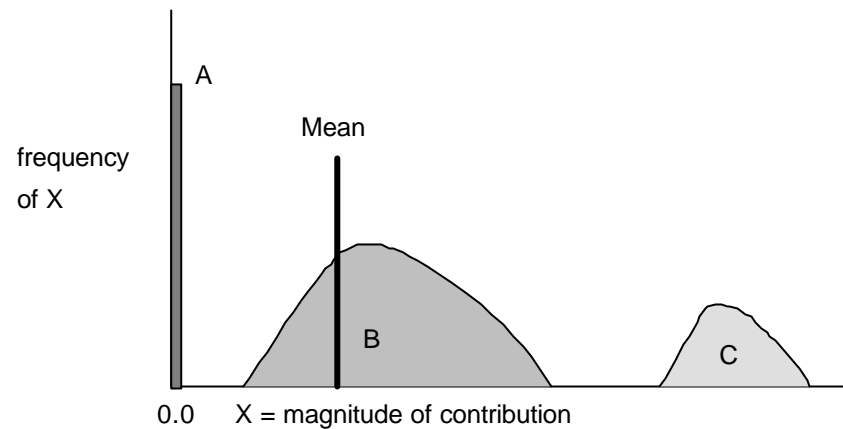
The evaluation of all the results described above are subject to statistical errors which generally reduce as the number of samples increases. Each scored quantity is an average of contributions made by all the samples taken. During a MCBEND calculation the contributions to each scored quantity made by the current sample are accumulated in an array of registers. Let one such register be called X. A given sample may make several contributions to X; for example a particle may scatter backwards and forwards through a scoring volume several times while remaining in the same scoring energy group. At the end of the sample the contribution and its square are accumulated in two further registers for the batch as a whole and X set to zero ready for the next sample. The scored results are all in the form of a mean value for the entire batch so the standard error on the mean is required to assess their statistical accuracy. This is defined in Section 4 and takes the form:

$$\begin{aligned} \text{Coefficient of variation} &= \frac{100}{\bar{X}} \frac{s}{\sqrt{N-1}} &= \frac{100}{\bar{X}} \sqrt{\frac{\sum X^2 - N(\bar{X})^2}{N(N-1)}} \\ & &= \frac{100}{\sum X} \sqrt{\frac{N \sum X^2 - (\sum X)^2}{(N-1)}} \end{aligned}$$

where Σ = sum over all N samples in the batch

MCBEND prints the coefficient of variation (%) for each scored result and, for most results, the estimated value ± 1 standard deviation. For most results it also monitors and displays the number of samples which have made a contribution to a particular score.

The following sketch shows the form which may be taken by the distribution of contributions to a score from individual samples. Note that the distribution is definitely non-normal in the statistical sense. The *mean* value is that used in the estimators and, as discussed in Section 4, the variance on the mean is expected to have a normal distribution.



- A = a delta function at 0.0 representing the number of samples which make no contribution to the score
- B = the contributions made by the bulk of the samples
- C = contributions made by occasional, high weight samples

The shape of the contribution frequency in B will depend on the variation of the particle weights arriving at a scoring region and the distribution of, for example, the length of the tracks used in track length estimation. The contributions in C may be from samples born in those parts of the source which are infrequently sampled and therefore carry a high weight. The presence or absence of such contributions will clearly influence the value of the mean and its coefficient of variation. The hazards of excessive biasing are discussed in Section 5.

11 GEOMETRIC MODELLING

It is obviously necessary to supply to a Monte Carlo code a description of the physical system being modelled. In MCBEND the volume of space of interest is subdivided into zones in each of which the material is considered to be uniform. Materials of different composition are each assigned a reference number by the user; the chemical composition of each material forms part of the input. Special material numbers are also used such as 0 for a vacuum (equivalent to air in most cases).

Fractal Geometry is a system of solid geometry modelling for Monte Carlo particle tracking codes. The principal objective is to subdivide the problem space into volumes (*zones*) of uniform material. Individual zones are defined as the intersections and differences of simple mathematical *bodies* such as cuboids, cylinders and spheres. The problem to be specified may optionally be divided into a hierarchical system of *parts* to simplify the construction and to take advantage of any replication which may be present.

12 VARIANCE REDUCTION**12.1 Introduction**

In a typical shielding problem the attenuation between source and detector might be by a factor 10^6 . In order to get a reasonable score at the detector - say a modest 100 scoring samples - the number of source particles which would need to be generated in this example case would be of the order 10^8 . This would make a Monte Carlo calculation so slow as to be impractical. The problem is that the bulk of the particles would be tracked part way through the problem and then lost (through absorption or escape) without making a useful contribution.

A solution to this problem is to *split* the particles at intervals in their history. When a particle is split, each fragment has its own subsequent history and it is hoped that some of the fragments will survive to make a useful contribution at the detector. In practice the splitting process is repeated at intervals throughout the history to achieve a significant benefit. Each sample carries a weight, W , which is a measure of the number of physical particles it represents. When it is split into n fragments the weight of each becomes W/n so that no bias is introduced in the expected result. When a fragment makes a contribution to a score at the detector after many splittings its weight will be much less than that with which it was born in the source. Since weight is included in the scoring estimators this means that it will make a small contribution. The overall effect of this process is to enhance the probability of a sample penetrating the shield but reducing the size of its scored contribution to represent the attenuation.

A complementary situation occurs when particles are travelling into a remote part of the shield from where they are expected to have little chance of reaching the detector. To terminate every such history may introduce bias but it may be inefficient to track each one to a natural conclusion. The solution here is to play a game of *Russian roulette* with a suitably chosen survival probability. If this probability is one in n then the surviving particles have their weight increased by a factor n to compensate for the fatalities.

This combination of splitting/Russian roulette (s/r) is a common technique in Monte Carlo codes and is the principle mechanism for variance reduction in MCBEND. The obvious problem is that of deciding when to apply s/r and to what extent. In MCBEND the volume of the problem space is subdivided by a set of fictitious surfaces and s/r considered when particles cross these surfaces. These are generally known as *splitting surfaces* even though they are used just as much for Russian roulette. The splitting surfaces divide the problem into a set of volume cells; the entire system is referred to as the *splitting mesh*.

For each splitting mesh cell a value of a parameter known as *Importance* is assigned. This is formally the expected contribution to a specific

detector which would be made by particles appearing in the cell and is mathematically equivalent to a solution of the adjoint problem. Fortunately, this function (potentially as difficult to derive as the Monte Carlo answer being sought) does not need to be known exactly in order to achieve practical variance reduction. One common technique is to simplify the problem and solve the adjoint equation using a diffusion theory code - possibly in a system with a reduced number of dimensions. The adjoint diffusion fluxes then provide a good working approximation to the importance function.

In its simplest form the logic in MCBEND is to split particles when they cross a splitting surface and pass from a region of lower importance to one of higher importance. Russian roulette is played when the importance decreases. The degree of s/r is governed by the ratio of the importances on either side of the boundary.

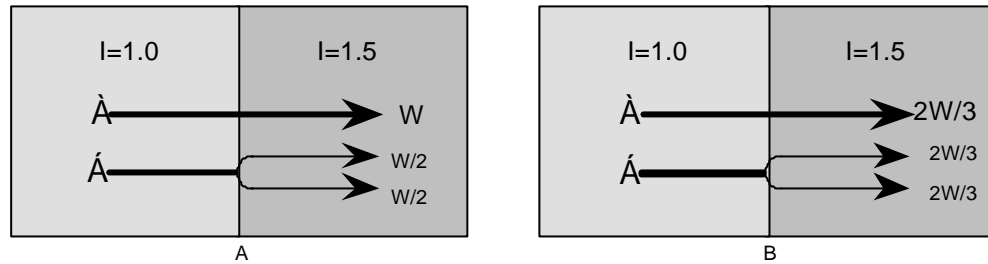
12.2 Specifying Importances

Having defined the principle and terminology of s/r we may now consider some of the details of its implementation. In MCBEND the splitting mesh may be chosen to be identical to the material mesh as described in Section 11 so that each distinct material zone has an assigned value of importance. This is, however, unsatisfactory when the problem contains large volumes of uniform material within which the importance function may vary dramatically. The alternative is provided of overlaying the material mesh with an independent splitting mesh so that splitting surfaces and importances may be optimised. The splitting mesh is then a set of orthogonal surfaces in XYZ or R θ Z geometry. An additional option is to overlay the material mesh with a finite-element type of importance mesh. This special option (FEMCAT = Finite Element Monte Carlo Accelerated Tracking) creates a specific, automated link between a MCBEND calculation and an adjoint solution determined by the finite-element diffusion code FENDER.

The energy range of the problem may also be divided into energy groups with s/r considered when particles lose energy and move from one group to another. The splitting energy group scheme is generally independent of that used for scoring or source definition except in multigroup cases when some degree of compatibility is required. An importance map is then, in general, a function of three spatial dimensions and an energy dimension. For time-dependent cases a component of the importance map which is a function of particle flight time may also be specified. This is separable from the spatial variation. In the following discussion on the mechanisms of splitting and Russian roulette the sketched importance maps will be one or two dimensional.

12.3 Implementation of Splitting

The first case to consider is the alternative ways of applying s/r when the ratio of importances across a boundary is less than 2.0:



The above sketches show two adjacent meshes with importances of 1.0 and 1.5. Clearly a particle cannot split into 1.5 fragments so some recipe is required. Sketch A shows the mechanism whereby a particle splits into two fragments on half of the boundary crossings (set ②) and is unsplit on the other occasions (set ①). This is a compromise between splitting or not splitting. It does not introduce any bias since the total weight for the two sets of particles ① & ② is the same immediately on either side of the boundary. An objection to this method is that the weights of particle fragments arriving in the right-hand side now differ by a factor of two. If this type of processing were continued throughout a problem then the fragment weights could become dispersed over a wide range. This in turn will increase the spread on the contributions to any scores and consequently increase their statistical uncertainty. This is counter-productive in a variance reduction technique.

Sketch B shows an alternative in which splitting is again on a 50:50 basis but regardless of the decision the particle weight is reduced by the importance ratio i.e. 1.5. This overcomes the problem of weight dispersal and is, as before, free from bias over a large number of samples. It is, however, a further source of statistical noise if relatively few scoring samples cross a particular splitting surface - especially if the ratio is close to either 1.0 or 2.0. The general procedure for such splitting when the importance change is other than the example 1.5 but still less than 2.0 is as follows:

$$\begin{aligned} \text{Ratio of importances} &= 1 + f \quad (0 < f < 1) \\ \text{Random number} &= R \end{aligned}$$

Split into 2 when $R < f$
 Do not split when $R > f$) in either case reduce the weight by a factor $1+f$

This process requires a significant amount of computing time if carried out many times. It often happens that a fine splitting mesh required in one energy group to describe a rapidly changing importance function leads to ratios of importances close to unity in some other group where the importance gradient is less severe. Under such conditions the process could be applied repeatedly for factors in the range 1.0 - 1.1. Since importance maps are generally approximate it is not worth the effort to split on such small ratios.

In MCBEND the logic used is:

No splitting takes place unless the importance changes by at least a factor of two

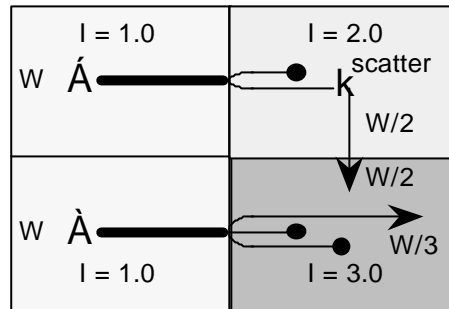
The restriction of splitting to a lower limit of two in turn poses another problem: what happens when the importance changes by a large factor over a series of meshes but by less than a factor of two at any individual mesh? A solution is to allow the particle to carry a parameter P . When the particle is born, P is assigned a value equal to the importance at the place and energy of birth; $P = P_0$. When crossing a splitting boundary into a mesh with importance I the splitting is based on the ratio I/P . If splitting into N fragments occurs then P is replaced by NP otherwise P is unchanged. When the particle scores a result the weight assigned at birth (W) is factored by P_0 / P .

An example may clarify this mechanism:

Importance I	4.0	5.0	9.0	12.0	16.0
Importance ratio		1.25	1.8	1.33	1.33
Parameter P	4.0 (= P_0)	4.0	8.0	8.0	16.0
$\frac{I \text{ after boundary}}{P \text{ before boundary}}$		1.25	2.25	1.5	2.0
Scoring weight $= W P_0 / P$	W	W	$W/2$	$W/2$	$w/4$
	1	2	3	4	5

A particle of weight W is born in a region with importance 4.0. No splitting occurs at the first boundary but at the second the ratio I/P exceeds 2.0 and binary splitting occurs with a doubling of P . Note that P is less than the mesh importance (9.0) in this mesh reflecting the fact that the true splitting ratio was 2.25 but only integer splitting is possible. In the fourth mesh the value of P falls even more short of the local importance but not enough for further splitting. On entering the fifth mesh a further binary split is appropriate. In the schematic representation of the particle track, one of the two fragments generated at the first splitting is lost in mesh 3 through a collision.

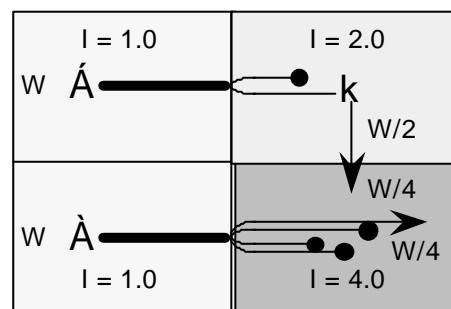
In principle, splitting could take place into any integer number of fragments but consider the situation shown below:



Particle ① crosses a boundary with ratio of importances 1:3 and splits into 3 fragments of weight $W/3$. Particle ② takes a route (including a scatter event) via the region with importance 2.0 and therefore splits into two fragments of weight $W/2$. On entering the lower right region the ratio of importances is 2:3 and no further splitting occurs. Thus we have two particle fragments arriving in the same importance region with differing weights. This spread of weights is generally undesirable and readily avoided if splitting is only allowed to occur in powers of two.

On crossing a splitting boundary MCBEND performs N-fold splitting N is the highest power of 2 which does not exceed the factor by which the importance increases.

A simplified argument is that importances are approximate and therefore splitting into 2 rather than 3; 4 rather than 7 etc. on any single occasion will not significantly affect the variance reduction in particle histories which generally undergo many splitting boundary crossings. A modification to the above example illustrates the constancy of weight which is maintained using this logic.



In this example the samples ① and ② arrive in the lower right-hand region with weight $W/4$ via two completely different routes. The same result would be obtained if the importance of the lower right mesh were any value in the range ; $4.0 < I < 8.0$

A population explosion of particle fragments could occur in cases where the importance changes dramatically across a boundary. To prevent this the number of fragments after splitting is limited to 32 with appropriate weight adjustment. This limit could lead to a spread in particle weights in a given region but its application should not be necessary in a

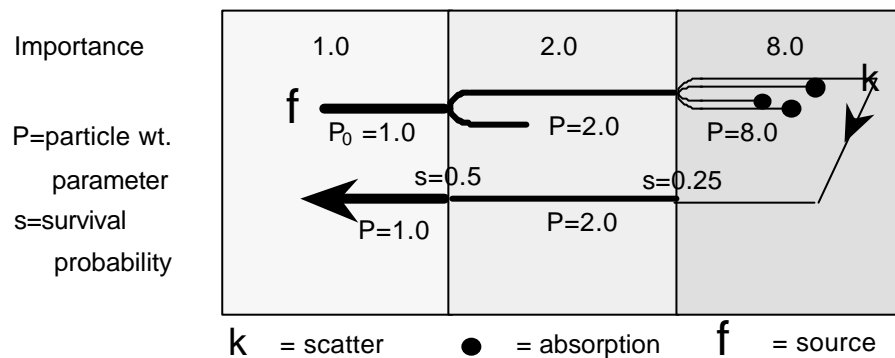
reasonable importance map. A spread of weights has less disastrous consequences than unlimited splitting.

12.4 Implementation of Russian Roulette

The process of Russian roulette is also subject to similar variations of implementation. If a particle crosses an importance boundary with a ratio of importances s ($s < 1$) then one method would be to allow a survival probability of s and raise the weight of surviving particles by a factor s . This would not, however, be strictly complementary to the splitting process as implemented in MCBEND where weight adjustments are based on powers of two. MCBEND logic for Russian roulette is therefore:

Survival probability = 1 in N ; weight increase = factor N
 N is the lowest power of two \geq (higher importance/lower importance)

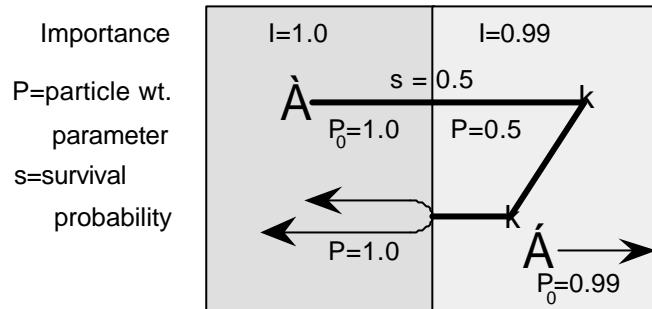
If all the importance ratios are powers of two then this logic for Russian roulette is quite reasonable as illustrated in the following sketch.



The particle, born in a region with importance 1.0, travels through regions with importance 2.0 and 8.0, scatters and travels back. The survival probabilities for Russian roulette on the return journey are denoted by the values of s . The scoring weight in any region is WP_0/P where W is the birth weight; this may be seen to be constant in any given region.

If the particle track has been subdivided by splitting and is in the form of a beam of fragments then Russian roulette reduces the population of the beam and increases the weight of the surviving fragments. This differs from the primitive alternative of considering death/survival for the entire beam.

For importances with values other than powers of two the Russian roulette in MCBEND is rather heavy handed. The following example illustrates the behaviour when the importance ratios are between 1.0 and 0.5 for roulette.



A particle ① is born in a region with importance 1.0 and moves into a region with importance 0.99. The logic in MCBEND will result in Russian roulette with survival probability 0.5. The parameter P for a survivor is reduced to 0.5. If the particle subsequently recrosses the same boundary then splitting will occur since the ratio $I(\text{next})/P(\text{last})$ is now 2.0. The parameter P then changes back to its original value for the first region. It can be seen that a particle born in the second region ② will have a scoring weight differing from particle ① in that region. A general consequence of only carrying out s/r in powers of two is that scoring weights of particles in a given region arriving via differing routes can be distributed over a range with an upper limit twice the lower limit. This is not an unacceptable spread; the estimators themselves (Section 10) generally introduce a spread of at least this order. Eg the variation of track lengths threading a scoring region.

It could be argued that Russian roulette should not be played until the ratio of importances falls below 0.5; this would be a more elegant complement to the splitting process. It would not, however, eliminate the two-fold spread in scoring weights. An argument in favour of the adopted logic is that roulette should always be played as soon as possible to reduce the effort of tracking particles travelling in unfavourable directions.

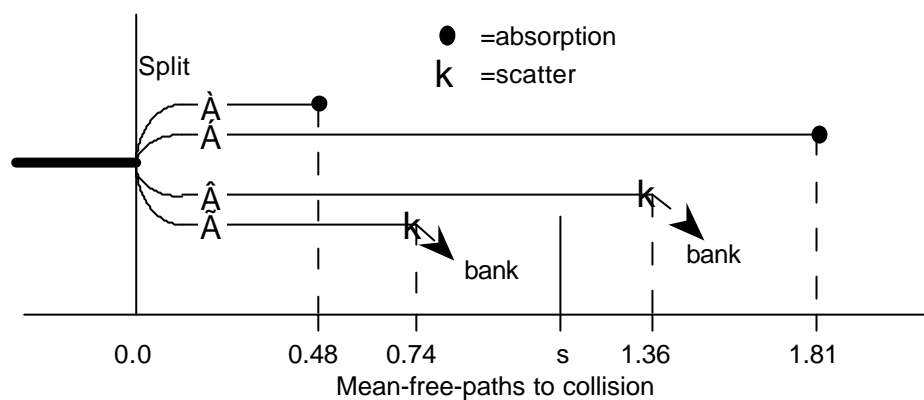
12.5 Post Splitting Activity

When a particle is split into n fragments each will follow a different history. The mechanism in MCBEND is to store the current parameters of $n-1$ fragments in an area of the program called the splitting bank. The history of the n 'th fragment is then followed to its conclusion (ie absorption, escape or death in Russian roulette). Another fragment is then retrieved from the splitting bank for processing. If further splitting occurs in the history then more entries are made in the splitting bank. Entries and withdrawals from the bank continue in this way until the bank is empty and the last fragment has died. The current sample is then considered to be complete.

A primitive application of this logic would be to bank the fragment parameters (position, direction, energy, weight etc.) at the splitting surface. However, all fragments created at a given splitting surface have all such parameters in common until their first collision. Tracking each fragment separately along the same initial trajectory would be inefficient - especially if several geometric boundaries are intersected before the first

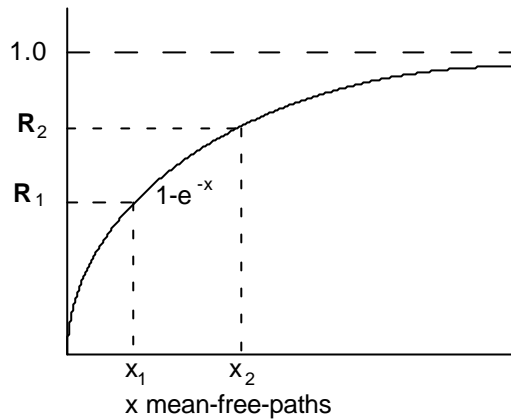
collision. MCBEND therefore uses the concept of a beam of particle fragments leaving a splitting surface. A beam of n fragments of reduced weight are tracked after leaving the splitting surface. When each member of the beam suffers its first collision its post-collision parameters (if it survives) are entered into the splitting bank. This continues until only one fragment remains in the beam; this one is tracked to extinction.

As illustrated in Section 6 the number of mean free paths to collision from any point is given by $-\ln(\mathbf{R})$. Each particle in a beam of n will travel a different number of mean free paths to its first collision. In principle, n such values could be determined and then sorted into order so that fragments could be dropped out of the beam into the splitting bank in sequence along the beam trajectory. This may be represented schematically as follows:



A particle splits into four fragments ① ② ③ ④. Four values for the number of mfp to collision are sampled and ordered; typical values are marked on the scale. On processing the first collision (fragment ①) an absorption is recorded, the fragment terminates and the beam population reduces to three. At the next collision (fragment ④) a scatter occurs and the parameters of the new trajectory are banked. A similar action is taken for fragment ③ and, finally, fragment ② is tracked to extinction. At this point the post-collision parameters for fragment ③ are withdrawn from the bank and its history processed. When terminated, the post-collision parameters of ④ are withdrawn from the bank.

Suppose, however, that there is a further splitting surface at S . The beam population may then be revised and new values for the mfp to collision for new fragments must be determined and sorted. With elaborate splitting and re-splitting this process of sampling mfp to collision and sorting could become very inefficient. The problem is overcome in MCBEND by one of two alternative processes: either stratified sampling or the nearest collision beam algorithm. The cumulative distribution function for sampling mfp to collision has the form:

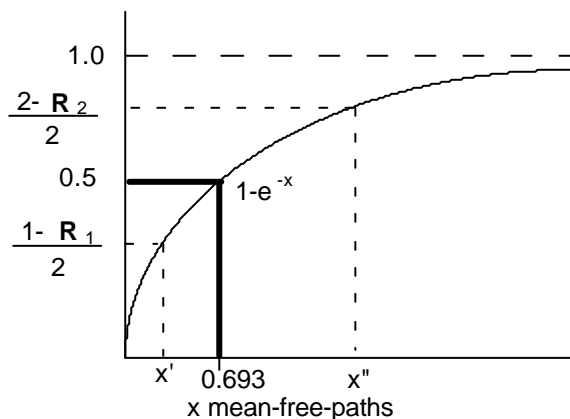


The number of mean-free-paths to collision (x) is sampled as $-\ln(1-R)$. The example shows two such samples. If taken independently there is no way of determining which will be the shorter without sorting. For a large number of samples the chosen random numbers would be expected to divide equally into those greater than 0.5 and those less than 0.5. If, when a pair of samples of R are required, one is taken from each side of 0.5 there will be no net bias over a large number of samples.

This is the method of stratified sampling and is achieved by the following scalings:

$$\begin{aligned}
 R' &= (1 - R_1)/2 & 0.0 < R' < 0.5 & \quad (0.0 < R_1 < 1.0) \\
 R'' &= (2 - R_2)/2 & 0.5 < R'' < 1.0 & \quad (0.0 < R_2 < 1.0) \\
 \text{ie } R' &< R'' \\
 x' &< x''
 \end{aligned}$$

If the scaling and evaluation for x' is performed first it will give the number of mfp to the first collision in the beam of two. Evaluation of the distance to the second collision (x'' mfp) may be deferred until it is required. The process is represented diagrammatically as:



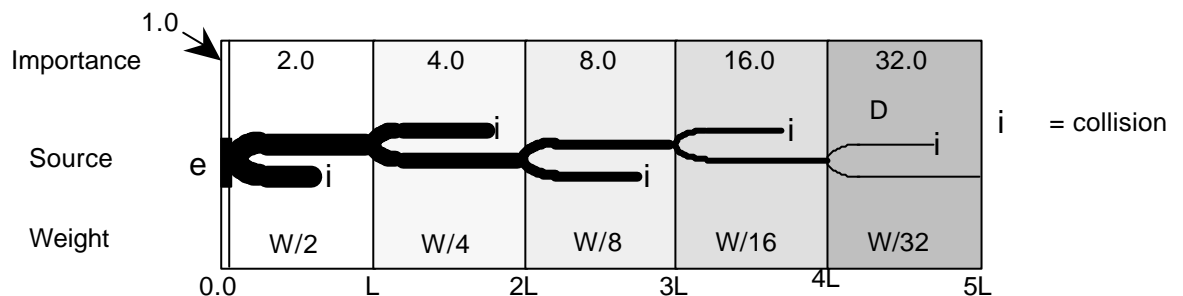
In this processing one fragment always collides before travelling $\ln 2$ ($=0.693$) mean-free-paths; the other travels at least this distance. This

technique of stratified sampling may be generalised for a beam of fragments leaving a splitting surface with n members as:

$$R' = (i - R_i)/n \quad x_i = -\ln(R') \quad (i-1)/n < R' < i/n \quad 0 < R_i < 1.0$$

where i ranges from 1 to n and counts the fragments in the beam in the order in which they collide. As in the simple case for two-fold splitting the value of x_i does not need to be calculated until fragment $i-1$ has had its collision. If at any stage the beam reaches another splitting boundary the beam population may be revised and collision sites for the fragments must be sampled using new values of n and i. This will not require the discarding of much information for the original beam (only the distance to collision for one fragment) and is therefore quite efficient. In principle the same random number could be used for R_i but this economy is not implemented in MCBEND.

A dramatic illustration of the effects of stratified sampling is given by considering the transmission of monoenergetic particles through a uniform slab from a monodirectional plane source. The attenuation will be a simple exponential and a 'perfect' binary splitting will have doubling planes every 0.693 mean-free-paths through the slab. The behaviour of the particles is sketched below :



For the purpose of this example a splitting plane is inserted immediately after the source: this is not normal practice. Two particle fragments enter each full-sized mesh interval. The stratified sampling causes one to collide within the mesh and the other to reach the next splitting boundary. For every particle started, a contribution will be made in the end region marked D. If the length of the terminated fragment track in each region is denoted by C we have:

$$\begin{aligned} \text{expected mean value of } C &= \frac{1.0 < C < L}{\ln 2} \\ &\sim 0.443 L \end{aligned}$$

The track length estimator will accumulate, for region D, a value of :

$$(LW/32) (1+C) \text{ for each sample started}$$

$$\text{Expected value of the contribution.} = (LW/32) (1.443)$$

Stratified sampling is the older method used in MCBEND for sampling the number of mean free paths to the next collision. It is quite efficient - dramatically so in the idealised example above - but is difficult to implement in conjunction with some of the options added to MCBEND in recent years. The alternative method is known as the nearest collision beam algorithm. It relies on the property of random numbers that:

the result of taking the largest of N random numbers is equivalent to taking the N th root of a random number.

Since $1-R$ is itself a random number in the range $0 < R < 1$ then the problem of choosing the number of mean free paths to the first collision of a beam of N fragments is equivalent to finding:

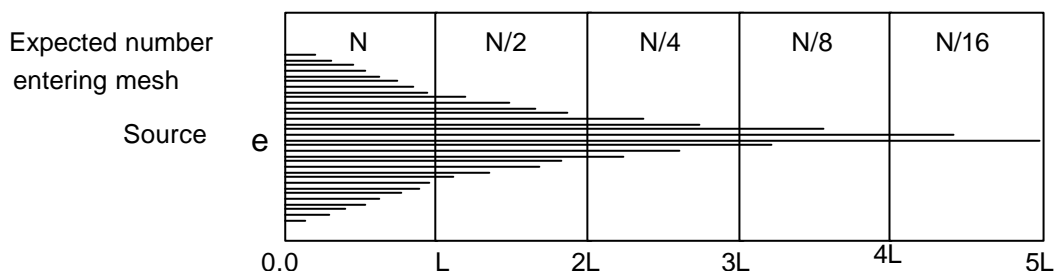
$-\ln(R_N)$ where R_N is the largest of N random numbers.

This in turn is equivalent to sampling $-\ln(R^{1/N})$ or $\frac{-\ln(R)}{N}$

This algorithm allows the distance to collision of each member of the beam to be sampled from a single random number and, unlike stratified sampling, does not commit other members of the beam to travelling a specified minimum distance. It less efficient than stratified sampling in idealised problems but not significantly so in general, realistic cases.

The nearest collision beam algorithm has become the recommended option in MCBEND and is the default in MCBEND9 onwards. Stratified sampling may still be selected if required.

Consider now the analog case i.e. with no splitting. The following sketch shows the expected behaviour of a batch of N samples:



In the last mesh the number expected to arrive is $N/16$ all having the unmodified starting weight W . Of these one would expect half to cross the region uncollided and the other half to collide. The expected mean track length for the colliding fragments will be $0.443L$ as described above. The track length estimator will accumulate a value:

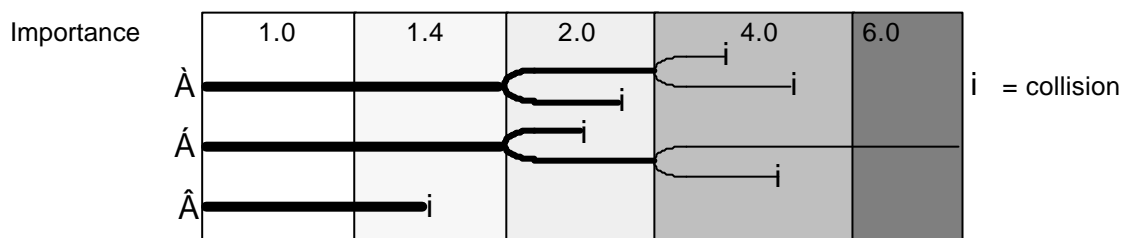
$$= \frac{W (N/16) (0.5L + 0.5 (0.443 L))}{(LW/32) (1.443)} \quad \text{for large } N \text{ per sample started.}$$

Happily, this is in agreement with the splitting case. Consider now the statistical errors in the two examples. Suppose we have a batch of 64 samples. With splitting, 64 pairs of particle fragments will reach the last region. One of each pair will always pass through; the other will collide. The parameter C will therefore be estimated from 64 collisions. In the analog case one would expect four samples to reach the last mesh and two of them to collide within it. C would therefore be estimated from two collisions. These figures of 'two' and 'four' in the analog case would of course be subject to statistical noise; in some batches of 64 samples none would reach the last mesh.

For this very stylised problem the advantages and lack of bias of the splitting algorithm are clearly demonstrated.

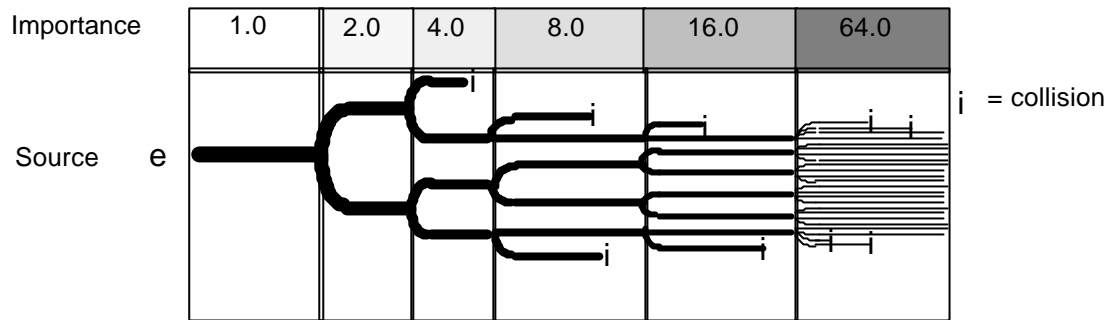
12.6 Effects of Imperfect Splitting

In a practical case the perfection of splitting illustrated above can not be achieved; the following examples illustrate the effects of deviating from the ideal. None of the effects are particularly serious unless existing in extreme. In most real cases one would expect all the following situations to occur for some samples in a given batch



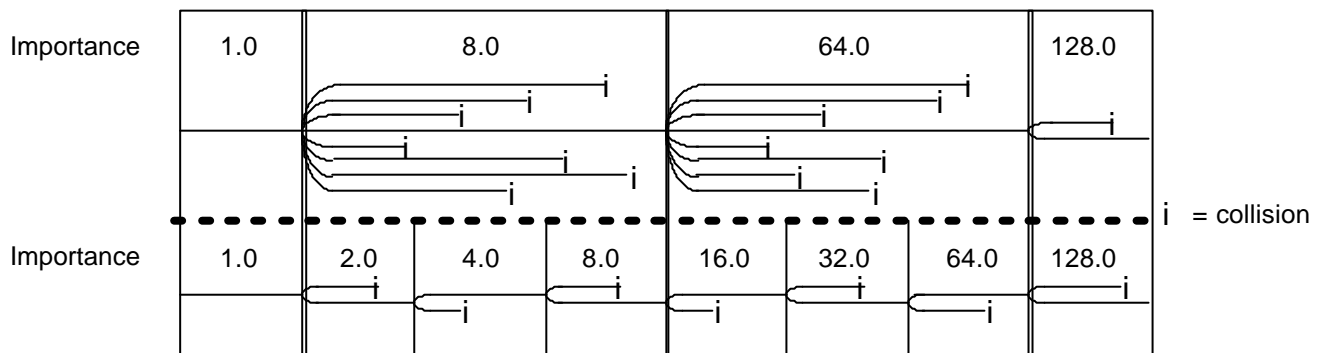
Inadequate splitting

Insufficient splitting can occur if the importance does not change quickly enough to counter the loss of particles through absorption, escape or Russian roulette. Excess particle loss can also occur if the splitting planes are too far apart thus reducing the frequency of splitting. In the example, track ① terminates before reaching the end of the mesh because the fourth mesh interval is too wide and both fragments are absorbed before re-splitting is considered. Track ③ ends in collision before the importance has changed enough to incur splitting. Even under these conditions some tracks ② can still penetrate the problem if the random sampling of distance to collision chooses a sequence of larger-than-average values. The consequences in the example are that only one in three of the samples shown reach the right hand boundary.



Excessive splitting

Excessive splitting occurs when the splitting planes are too close together or the importance is changed by too large a factor. In the above example the splitting and re-splitting of the single particle is not balanced by a corresponding number of losses. The beam population grows with penetration and can, in very severe cases, become so unstable that entries into the splitting bank continue to outnumber withdrawals until the bank is full or the calculation runs out of allocated time. In less severe cases the problem merely becomes inefficient as the number of particle fragments (and their down-scattered progeny) all have to be processed but each cascade still only counts as one sample.



Infrequent Splitting

The upper part of the above sketch shows the effect of infrequent splitting into many fragments; the lower part shows regular doubling for comparison. The effect of stratified sampling in these ideal cases is such that one fragment of each sample reaches the right hand edge of the problem in either case but the number of collisions processed (marked *) differs significantly. The infrequent splitting produces many collisions the progeny of which must all be banked and subsequently tracked. All information obtained from the consequent effort is part of a single sample and will not generally give efficient variance reduction.

To reiterate an earlier statement: in practical cases a mixture of all these effects is likely to occur to some degree. Even in a simple slab geometry with a well known importance map the use of isotropic sources will produce behaviour variations between samples. A particle travelling normally to a set of splitting planes will be subjected to more frequent splitting than one travelling at an angle.

The performance of an importance map may be assessed from an output edit of MCBEND which shows the number of particle fragments (regardless of weight) which enter each importance mesh in each energy group. Sudden changes in the values along the principal route from source to detector indicate under/over splitting. The decreasing flow of fragments into regions of progressively lower importance should be clear in the edit. An importance map may be refined on the basis of this data (obtained, perhaps, from a short pilot run) by re-positioning splitting mesh boundaries or altering importance values or, more commonly, obtained automatically by the use of the MAGIC module.

12.7 Refinements in Variance Reduction

Although this chapter has been long and detailed it does not cover the entire range of features in MCBEND variance reduction. The following notes refer briefly to some of the embellishments and interaction with other code activities.

(1) The mesh and/or energy group scheme used to define the importance function may also be used for scoring. This is often used when the material geometry description contains large, uniform zones of material but scoring is required in several subdivisions. It may be more convenient to use the splitting mesh rather than introduce fictitious extra boundaries in the geometry description.

(2) If an importance map is not very well known a binary splitting mode can be switched on which invokes doubling at every boundary where the importance increases and Russian roulette with 50% survival at every decrease. This is less efficient than the normal mode but is relatively stable and avoids population explosions of particle fragments

(3) The importance map may also be used to control the sampling of source particles when the simple source is used (Section 9). This reduces the spread of particle weights in a given region by equalising (to within a factor two) the weight of particles migrating into a region and the weight of those born in that region. This option is associated with the source definition and referred to as *Automatic weighting*.

(4) When using an importance map defined on a finite element mesh (typically an array of irregular triangles or triangular prisms) the s/r activity is not considered at boundary crossings since these are time-consuming to evaluate. Instead, the code considers s/r at regular optical distances along the particle tracks and evaluates the importance at spot points by interpolation from the finite-element node values.

13 NUCLEAR DATA

The purpose of nuclear data is to describe the frequency and the outcome of interactions between particles and the material through which

they are passing. Two principal forms are used in MCBEND: multigroup and point energy.

In multigroup data the average behaviour of particles in a set of energy groups is determined by pre-processing libraries of basic nuclear data such as ENDF-B. Pre-processing performs averaging of the fine structure of nuclear cross-sections using selected weighting spectra. The form used in MCBEND is common to that used in deterministic codes such as ANISN and DOT which represent the scattering cross-sections as coefficients in truncated Legendre polynomials. When a source particle is generated in a multigroup case its specific energy is indexed into the multigroup energy boundaries to identify a group number. Thereafter the particle makes transfers between groups according to the probabilities in an inter-group scatter matrix and its energy change is only monitored as a group number. This imposes restrictions on the flexibility with which scoring may be requested or importance functions defined. Each must be in the basic multigroup scheme or a reduced form created by assigning several multigroups to each of a set of coarser scoring/importance groups. Response functions must also be defined in the basic multigroup scheme.

The multigroup data is stored in a library covering a range of nuclides. This is processed in a given MCBEND case to produce equivalent data for the user-defined materials. Multigroup data may be used for the tracking of either neutrons or gamma rays or, given appropriate data, for coupled neutron-gamma calculations. In this form the production of gammas from neutron capture is merely a form of inter-group transfer. By transforming the inter-group scatter matrices the data may also be used to perform adjoint calculations. The sensitivity option (Section 10) is not available in multigroup calculations.

Multigroup data is inherently less accurate than the point data described below. Its principal uses are:

- (i) comparisons between MCBEND and deterministic codes using common data:
- (ii) executing adjoint calculations:

The alternative to multigroup data is point energy data. In this form the nuclear data is stored at a sufficiently large number of energy points that little accuracy is lost by interpolating between them. Particle tracking using point data retains the particle energy as a continuous variable; the changes of energy resulting from interactions with atoms in the materials are evaluated from appropriate scattering laws.

In neutron cases the data used in MCBEND has been processed from basic nuclear data compilations such as ENDF-B, UKNDL or JEF to form libraries of relevant nuclides. A given MCBEND case reads the chosen library and extracts those nuclides present in the user-defined materials. The material compositions are combined with this data to evaluate the total material cross-sections and the partial cross sections of

each individual nuclide within it. When a collision occurs in a material a specific nuclide is chosen from a cumulative distribution formed by the fractional macroscopic cross-sections of the constituent nuclides. Having chosen a nuclide, its partial cross-sections are sampled to choose a specific type of reaction (e.g. absorption; elastic scatter; $n,2n$ etc.). For the chosen reaction the energy and angle of deviation of any scattered particles are then evaluated using scattering laws appropriate to the type of reaction and the energy range of the incoming particle.

Special treatments are available for thermal neutrons: these detailed cross-sections and models to allow for the relative velocities between neutrons and the thermally agitated target nuclei. For neutrons with energies below a few eV the thermal motions of scattering nuclides profoundly affect the outcome of collisions. There are three important processes that occur here:

- 1 Inelastic thermal scatter - a colliding neutron may either gain or lose energy from the motions of a bound target nucleus and some of the kinetic energy may be transferred to excited molecular quantum levels e.g. rotation, vibration.
- 2 Incoherent elastic scatter - the wave-like behaviour of the neutron combined with the random nature of some scattering media gives rise to incoherent elastic scatter where the neutron is scattered without change of energy. The angular distribution of the scattered neutrons is forward peaked and continuous.
- 3 Coherent elastic scatter - if the nuclei of the scattering medium are arranged in a well-defined array (i.e. a crystalline material) then coherent elastic scatter may occur. Scattering takes place into well defined directions which are a function of the wavelength of the incident neutron, and the lattice parameters. This type of scattering is not currently supported in MCBEND.

For the inelastic thermal scattering the cross sections are given through the use of a $S(\alpha,\beta)$ function which is obtained from the analysis of the collisions using quantum mechanics. The data describing the $S(\alpha,\beta)$ function for certain bound light nuclides is included in the MCBEND Continuous energy database, and is obtained from the JEF library. MCBEND currently uses these data for hydrogen in water and hydrogen in polythene, where this type of scatter is especially important. MCBEND also treats elastic incoherent scatter for hydrogen in polythene, where this is an important factor in determining the cross sections at thermal energies. This type of scattering is negligible for hydrogen in water.

Thermal scattering in all other nuclides is treated by applying a monatomic free gas model at the ambient temperature. Even for bound nuclei the monatomic gas model is applied when the incident energy is well in excess of the energy levels for internal oscillations.

Point energy neutron data is currently the only form of which may be used by the sensitivity option.

In point energy gamma cases a similar logic is used to represent the energy of particles and variation of data. Because gamma ray cross-sections are a relatively smooth function of energy and atomic number the data for individual nuclides are readily combined into material data for the compositions supplied by the user. During particle tracking the processed material data is interrogated and selection made, at each collision, of the probabilities for Compton scatter, photo-electric absorption or pair production. The post-collision energy and angle for Compton scatter are derived from the Klein - Nishina formula.

Normally the annihilation of the electrons and positrons generated in pair production events is assumed to take place at the site of their creation. An available refinement is to track the charged particles until annihilation occurs. This requires the inclusion of an additional library of data for charged particle migration. This data is used in a model which condenses the charged particle history into steps with continuous energy loss; this avoids processing the high number of individual collisions which occur in reality. The interchange between charged particles and photons works both ways; gammas produced by e^+ / e^- annihilation are tracked as such. The calculation may begin with charged particles if required by defining them as sources.

Coupled neutron-gamma cases cannot be executed directly in MCBEND using point-energy data but a facility is included for linking calculations for the two particle types. The collision events in a neutron case are saved to an interface file. A subsequent gamma calculation reads this file and combines it with gamma production data to give a gamma source. This two-stage process allows different splitting meshes and importance maps to be used for the differing particle types. The options of point estimation and sensitivity scoring (Section 10) are available in point-energy gamma calculations if gamma-ray data in the DICE format (used for neutron data) is used.

For further information see S Newbon, S J Chucas, 'Applications guide for Coupled (n, γ) Calculations' ANSWERS/MCBEND(94)13.

14

MATERIAL DATA

The libraries of nuclear data described above all contain information for individual nuclides. The macroscopic cross-sections for the materials in a particular MCBEND calculation are derived by combining the library data with user-defined material compositions. During the specification of the problem geometry the distribution of different materials is given by user-allocated material numbers. The chemical composition corresponding to each material is defined in MCBEND using a module (MINNIE) which is common to all ANSWERS shielding and criticality codes. This allows compositions to be specified using weight or volume fractions of constituent nuclides.

A useful refinement in MINNIE is the concept of a mixture. This is a chemical combination which may be included in the subsequent definition of materials. Suppose, for example, one wishes to represent a fuel bundle as a smear of its component parts. Mixtures may be defined to specify each of: fuel metal, cladding and coolant. These mixtures may then be used as volume fractions of a smeared core material. Another typical application is that of representing smeared regions of supporting webs and coolant; steel and water may be defined as mixtures and combined with different volume fractions to generate a number of smears with differing metal to water ratios.

In addition to conventional material compositions there are some special material numbers used in MCBEND.

Material 0 is a vacuum in which no interactions take place. In most problems this is an acceptable representation of air space between components.

Material -1000 is a perfect absorber. Particles striking material -1000 are immediately absorbed. A fictitious, unit track length is assigned to particles entering material -1000 to generate a finite score by track length estimation. This has little physical significance but does give some indication of the degree of absorption taking place in the special material. It is permitted to generate source particles within material -1000; these are instantly absorbed and a new source particle sampled. It is therefore possible to use material -1000 for shaping a geometrically complicated source but source normalisation may be affected.

Material -2000 is used in Fractal Geometry to signify the limit of the problem. Particles tracking across a boundary into a zone of material -2000 are considered to have left the system of interest and are terminated. It is not permitted to create source particles in material -2000.

Material -3000 is used in Fractal Geometry to define reflecting boundaries. Particles crossing a boundary into a zone of material -3000 undergo specular reflection at the interface.

Material -6000 is used in Fractal Geometry to define a black absorber in the MAGIC module and vacuum in the geometry module.

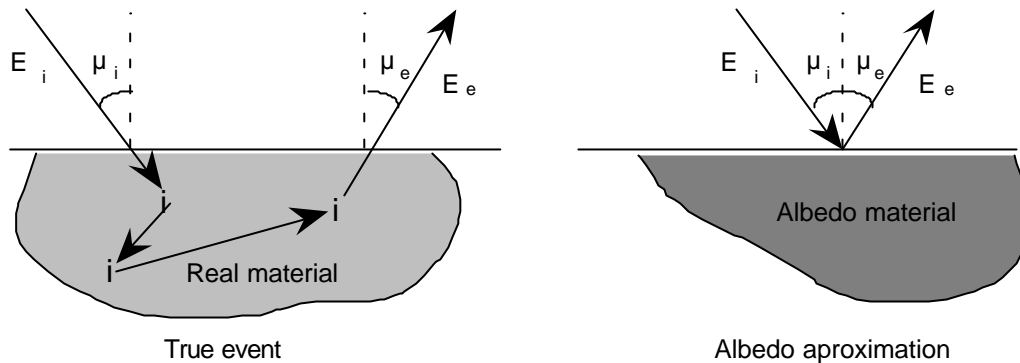
Material -n represents the n'th albedo material. The principles and applications of albedo materials are described in the following section.

15

ALBEDO DATA

An albedo surface is one from which incident particles are reflected with a change of direction and, possibly, a change of energy. Use of albedo

surfaces in MCBEND is an approximation which replaces the detailed tracking beyond a surface with a simple, but non-specular reflection.



The left hand sketch above shows a particle with energy E_i incident on a material boundary at an angle with cosine μ_i to the surface normal. After three collisions (*) it re-emerges with energy E_e at an angle with cosine μ_e to the surface normal at some distance from the point of entry. In the albedo approximation of the right hand sketch the process is replaced by a form of reflection at the point of incidence. Albedo data consists of a set of probabilities from which E_e and μ_e may be sampled as a function of E_i and μ_i . In a detailed set of data the azimuthal angle of the emergent particle would also be sampled from a defined distribution but in current versions of MCBEND the azimuthal distribution is assumed isotropic.

The albedo treatment is only available for point energy neutron or gamma-ray calculations. The data is supplied in an energy group scheme and consists of several independent components.

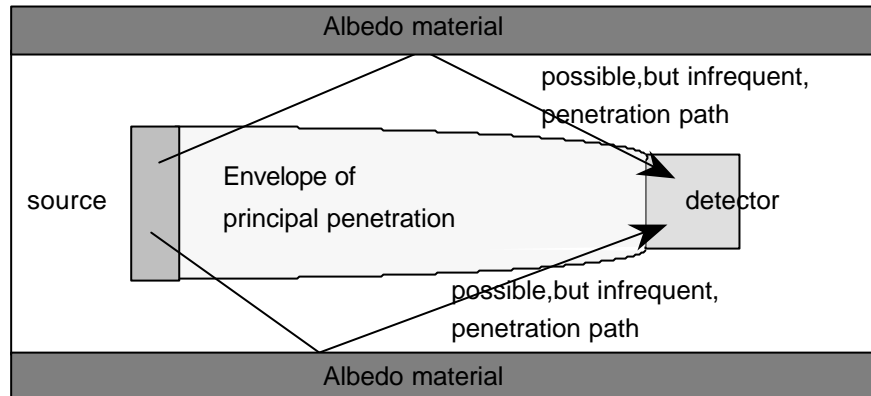
$$\text{Albedo} = \beta(g_i, g_e) \times \mu_i^{-n} (1 - n/2) \times \mu_e^{1-m} (1 - m/2) / \pi$$

The term β is a triangular matrix giving the probabilities that an incident particle in albedo group g_i will be re-emitted in an albedo energy group g_e . The group g_e may be the same as, or of lower energy than, group g_i . If the total probabilities for emergence into possible groups is less than unity then the residual represents the probability of absorption. Having chosen an emergent group then MCBEND samples an energy within it ($1/E$ weighting) to allow tracking to continue in the point-energy mode.

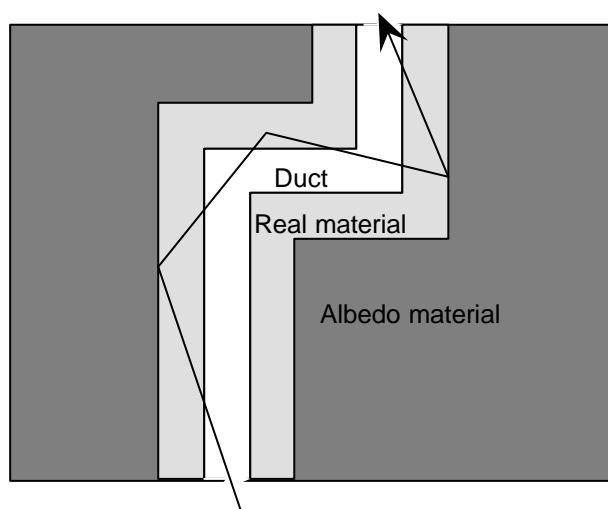
The next term imposes a dependence on the incident angle of the particle; the parameter n may be defined for each albedo group. The cosine of the incident angle is raised to the power $-n$; the rest of the term is a normalisation. If n is given a value 0 then there is no dependence on incident angle; larger values enhance the probability for re-emission as the angle of incidence is increased. Recommended values of n have been determined empirically and are supplied with the albedo data for MCBEND.

The final term imposes a dependence on the emergent angle; the parameter m may be defined for each albedo group. The cosine of the emergent angle is sampled from a distribution of the form μ^{1-m} ; the rest of the term is a normalisation factor. If m is assigned a value 1 then the emergent distribution is isotropic. Recommended values of m have been determined empirically and are supplied with the albedo data for MCBEND.

The advantage of the albedo treatment is speed but with the sacrifice of accuracy. Two typical examples of its usefulness are illustrated below.



The sketch shows a situation where most particles travelling from source to detector are confined to an envelope near the centre of the problem. The possibility exists of particles travelling to the edges of the problem and then scattering back towards the centre. These are expected to make a significant, but not dominant contribution to the result. Rather than treating the peripheral scatter events explicitly, an albedo material is introduced to give a faster (though approximate) treatment of such events.



This second example shows a case in which the principal route for particles penetrating the shield is by streaming along a duct coupled with scatter events close to the duct surface. Particles travelling deep into the

shield material are unlikely to return to the principal route. The duct is lined with a normal material to a thickness of one or two mean-free-paths so that scatter events close to the duct and penetrations near corners are treated exactly. The less significant possibility of particles returning from deep in the shield is modelled by reflection from an albedo surface. The depth of the real material can be chosen to suit the accuracy required and the complexity of the case. In the extreme, the duct wall may be represented as an albedo surface without any buffering real material.

For further information see S J Chucas, 'Applications Guide for Neutron Streaming' ANSWERS/MCBEND(93)12.

16**CONTROL OF A MONTE CARLO CALCULATION**

When executing a deterministic code there is a well defined end to the calculation which is reached when the equations have been solved for all mesh points and all energy groups. There is no corresponding end point to a Monte Carlo calculation; as progressively more samples are taken the results become more comprehensive and the statistical uncertainties reduce. The calculation is usually terminated when a specified number of samples have been taken or an allocation of computer resources (c.p.u. time) has been consumed. In MCBEND either or both these limits may be set; if both are specified then sampling stops when the first limit is reached. It is unlikely that a user estimate of these limits will be just sufficient to give results with desired accuracy at points of primary interest. MCBEND therefore includes the option of saving all the data at the end of a calculation to a dump file. The calculation may be continued from the stopping point by reading the dump file in a restart calculation.

Most major calculations are performed as a sequence of dumps and restarts. This allows the user to monitor the improvement in statistical accuracy as the calculation progresses and stop at an appropriate point. If progress towards acceptable variance is slow then the calculation can be abandoned at an early stage and revisions made to the input data for a subsequent attempt. Regular dumping and restarting also guards against the consequences of machine failure during very long calculations. If this should happen then the calculation can be restarted from the latest dump with little wastage of effort.

The information stored in a dump includes the current random number generator seeds (Section 3) so that the result of a sequence of dumps and restarts is equivalent to a single, continuous calculation.

In a practical Monte Carlo calculation the number of collision events, particle tracks and geometrical boundary crossings may run to several million. Although the program uses double precision arithmetic where appropriate and includes tests for many 'special case' events it is inevitable that there will be occasional failures during tracking. There is always the possibility of the rogue particle which perversely has a collision and changes direction just as it crosses two material boundaries simultaneously! With rounding errors this may cause the particle to

become confused over its current zone. Tracking errors at a late stage in a calculation may also occur if a particle visits an obscure part of the geometry model which has been incorrectly defined by the user. The action taken by MCBEND after ten such events is to restore the calculation to its state at the end of the previous sample and create a dump file. The last offending particle is then re-tracked with a print of all the events in its history for subsequent de-bugging. Finally the results of the calculation up to the last successful sample are printed.

If the particles are identified as rogues or the geometry errors are considered insignificant then the calculation may be restarted from the dump.

Some of the items of output may be suppressed/reactivated from one restart calculation to the next. This allows, for example, the print of particle fragment inflows (Section 12) to be monitored during the initial stages of a calculation and then suppressed until the end. A query print option is also available which will give details of every event (collisions, boundary crossings, reflections etc) which happens in the calculation. This obviously produces copious output. The option can be activated to commence after a specified number of samples. The evaluation of some scored quantities may be requested but their output suppressed until the final step in a sequence of restarts.