LA-UR -85-1173
CONF-8504110--12

MASTER

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36.

LA-UR--85-1173

DE85 010765

TITLE: MCNP VARIANCE REDUCTION OVERVIEW

AUTHOR(S): J. S. Hendricks
T. E. Booth

SUBMITTED TO: Joint Los Alamos/CEA Meeting on Monte Carlo Methods, Cadarache, France

DISCLAIMER

Los Alamos National Laboratory
Los Alamos, New Mexico 87545

FORM NO. 836 R4

MCNP VARIANCE REDUCTION OVERVIEW

J. S. Hendricks and T. E. Booth
Radiation Transport Group X-6
Los Alamos National Laboratory
Los Alamos, New Mexico 87544

ABSTRACT

The MCNP code is rich in variance reduction features. Standard variance reduction methods found in most Monte Carlo codes are available as well as a number of methods unique to MCNP. We discuss the variance reduction features presently in MCNP as well as new ones under study for possible inclusion in future versions of the code.

MCNP VARIANCE REDUCTION OVERVIEW

INTRODUCTION

The main drawback of the Monte Carlo method is that it is often too expensive to do many calculations of interest. This is not because the method is slow, but because a great deal of computational time can be wasted following unimportant or statistically insignificant events to achieve a desired result. Thus the key to making the Monte Carlo method attractive is to somehow concentrate on important aspects of a given problem without wasting time on the unimportant ones. We call schemes that do this "variance reduction methods" because they reduce the statistical variance for the same amount of computer time. This can also be looked at as increasing the computational efficiency or convergence rate for a given solution accuracy.

The MCNP code is rich in variance reduction features. These are computational tools that make it possible for the code user to better sample statistical events in the regions of interest and to reduce wasted effort in unimportant regions. Crucial to variance reduction are the means of determining which parts of a problem are important and which are not. Identification of the important aspects of a problem is done by extensive output summaries and solution tallies and diagnostics. These output features are straightforward and will not be further discussed. We will instead present the variance reduction feature presently in MCNP and some of the work being considered for future versions of the code.

Central to the art of variance reduction is the concept of particle weight. In order to simulate the transport of a large number of physical particles it is not necessary to follow all of them: rather it is only necessary to follow a statistically significant sample of particle "histories". Each history is assigned a weight which in some sense represents the number of physical particles modeled. At any time during the random walk of the particle it may be split into N particle "tracks" provided that the weight is divided by N. Alternatively , it may be killed with probability 1/N ("Russian rouletted") at any time provided the weight of surviving particles is multiplied by N. All variance reduction schemes work by putting a large number of particles of low weight in regions of interest and allowing only a small number of particles with high weight in unimportant regions of phase space.

VARIANCE REDUCTION IN MCNP

The variance reduction methods presently available in MCNP will now be described. We start with the simplest methods first and proceed to describe increasingly complicated methods.

GEOMETRIC SPLITTING AND ROULETTE

The most straightforward Monte Carlo variance reduction method is probably geometric splitting and Russian roulette. The geometry is divided into a number of geometric regions, or cells, each of which is assigned a cell importance by the code user. Usually a great deal of experience, intuition and a number of short trial and error test problems are required for a user to

specify a good set of importances. In the method of geometric splitting, a particle crossing into a cell of higher importance is split, whereas a particle crossing into a cell of lower importance undergoes Russian roulette. In this way particles from the source can be biased towards the tally region. The importances could be determined from an adjoint calculation, but because solving the adjoint would be just as difficult as solving the forward problem trial and error is usually the method of choice. The advantage of this method is that it is very straightforward and intuitive. The method is also very effective provided that the geometry can be sufficiently divided into enough geometric regions for a good importance function to be specified.

WEIGHT CUTOFF

Another common and simple variance reduction tool is the weight cutoff. A minimum weight is specified in each region below which Russian roulette is played. This avoids the loss of time following very low weight and hence unimportant particles. In MCNP the weight cutoff is usually specified as proportional to the inverse cell importance in each geometric region.

TIME AND ENERGY CUTOFFS

The easiest way to speed up a Monte Carlo problem is to truncate the problem. In all problems, geometric truncation is done by limiting the problem geometry so that the whole universe is not included. Problem truncation may also be done with a time cutoff which discontinues particle tracking after a specified time. It may also be done with an energy cutoff which kills particles below a specified energy range. Care must be used when truncating problems in these ways so that an important part of the physical model is not eliminated. For example, if a lower neutron energy cutoff of 1 keV is specified for a problem where there is fission or where photons are being produced from neutron reactions then low energy neutron reactions which produce the fissions or photons will not be modeled. The Monte Carlo calculation will happily converge to the wrong result with no indication whatsoever that the problem has been truncated.

SOURCE ENERGY AND ANGLE BIAS

Source energy and angle biasing are very valuable in many Monte Carlo applications. They are usually easy to implement because the source energy distribution and initial direction are often well known and relatively simple functions. This method works by artificially starting more of the source particles in the direction and energy regime of most importance and then adjusting the weight of the source particles as follows:

$$w_s = p/p^*$$

where

w_s = the weight adjustment factor for the source particle that was modeled;

p = the value of the true probability density function for the source particle;

p^* = the value of the artificial probability density function that was used in sampling the particle.

Problem efficiencies may be improved by orders of magnitude when invoking source biasing for problems where the source sampling efficiency is a major component of the total problem solution. However, overuse of source biasing may cause unacceptably large weight fluctuations.

In MCNP the user has several choices of artificial probability density functions for source biasing. For source energy biasing a table of arbitrary length may be specified. For source angle biasing, the user specifies a preferred direction which is an arbitrary vector in space. He then has two options: cone biasing or exponential biasing. For cone biasing a cone is defined with the preferred direction vector as its axis. The user then specifies the probability of starting within that cone. For exponential biasing the probability of starting in any direction falls off exponentially from the probability of starting in the direction of the reference vector.

NEUTRON INDUCED PHOTON SOURCE WEIGHT CONTROL

Another MCNP variance reduction scheme is the neutron induced photon source weight control. The user has several options. First, he can specify that one photon be generated at each neutron collision. Second, he may specify a minimum photon weight in each geometric region such that photons generated below this weight play Russian roulette and up to ten photons may be produced at the collision with weights above the specified cutoff. Finally, he may specify a multiplier, M, of the neutron weight so that when a neutron of weight W has a collision, generated photons of weight less than W * M are rouletted and up to ten photons of weight greater than W * M may be generated from that collision. The multipliers, M, may again be specified for each geometric region. The value of these options is that the number of photons generated in each region can be controlled thus putting more neutron induced photons in important regions and following fewer in unimportant regions.

EXPONENTIAL TRANSFORM

The exponential transform is another method which can be used to bias particles toward the tally region. The idea of this method is to modify the transport equation by following change of variables

$$\psi^*(\underline{r},E,\underline{\Omega}) = \psi(\underline{r},E,\underline{\Omega})e^{a\underline{\Omega}_b\cdot\underline{r}}$$

where $\psi(\underline{r},E,\underline{\Omega})$ = untransformed flux at location $\underline{r}$, energy E and direction $\underline{\Omega}$

a = arbitrary constant
$\underline{\Omega}_b$ = some preferred biasing direction

The resulting difference between the transformed and untransformed transport equation then turns out to be an equation that looks exactly like the original

transport equation except that the source term is different and the total cross section of the transformed equation is

$$\Sigma_t^* = \Sigma_t - a\underline{\Omega}_b \cdot \underline{\Omega}$$

Whereas this might result in a negative cross section, it is customary to treat the exponential transform method in the following way. The source term and random walk is left unmodified except for the total cross section which is modified as

$$\Sigma_t^* = \Sigma_t(1 - p\underline{\Omega}_b \cdot \underline{\Omega})$$

where $0 < p < 1$ is a constant determining the degree of biasing. Thus the artifical total cross section Σ_t^*, is never negative. Further, the sampled path length in the random walk process is stretched in the preferred direction, $\underline{\Omega}_b$, and reduced in the opposite direction. This is why the method is also known as "path length stretching". Unfortunately, the transform does not work well when the particle population does not have an exponential distribution. In particular, our experience is that the transform works best in highly absorbing media and only poorly in scattering media. Also, selection of the proper input parameters for the transform is tricky. Further, in the absence of a good means of weight control it has been our experience that particle weight fluctuations caused by the transform can produced misleading and unacceptable results. The reason for the weight fluctuation is because at each collision the weight must be modified by a factor of

$$w_c = \frac{e^{-p\underline{\Omega}_b \cdot \underline{\Omega}\Sigma_t s}}{1 - p\underline{\Omega}_b \cdot \underline{\Omega}}$$

in order to correct for the artificial adjustment of the total cross section. The $(1 - p\underline{\Omega}_b \cdot \underline{\Omega})$ denominator term tends to build up with each collision so that particles reaching the same point in phase space by different random walks can have vastly different weights. We discourage the use of the exponential transform in the absence of a good weight control such as weight windows. When weight windows are used, the exponential transform is fairly insensitive to the choice of biasing parameter and problem efficiencies typically double.

IMPLICIT CAPTURE

By implicit capture it is meant that a particle's weight is reduced by the capture probability at each collision rather than being killed with the probability of capture as in the analog case. In this way particles are not killed after a great deal of effort has been expended to transport them long distances. Implicit capture may also be done along a flight path rather than at

collisions, but this is mathematically equivalent to a special case of the exponential transform, so it is not considered to be a separate variance reduction method in MCNP.

Implicit capture along a flight path is worthy of a little bit more discussion. If $p = \Sigma_a/\Sigma_t$ in the above exponential transform equations and if $\underline{\Omega}_b$ is set to $\underline{\Omega}$,

$$\Sigma_t^* = \Sigma_s$$

where

$$\Sigma_t^* = \text{the scattering cross section} = \Sigma_t - \Sigma_a = \Sigma_s .$$

Thus the total cross section is replaced by the scattering cross section and the random walk process samples the distance to scatter rather than the distance to collision. This scheme is commonly employed by astrophysicists in the transport of stellar x rays. However, it is our experience that whenever this scheme is advantageous, adding the directional preference, $\underline{\Omega}_b = \underline{\Omega}$, is always better and using a stretching parameter $p > \Sigma_a/\Sigma_t$ is even better yet.

POINT DETECTORS

Point detectors or next event estimators, may be thought of as a variance reduction tool because they allow one to calculate the flux at a point. Often the point is in a region far from the main geometry of the problem in a region where it would otherwise be difficult to transport particles. Whether the detector is far or near it is impossible to transport a particle by a random walk to an infinitesimally small point in space. The point detector works by estimating at each collision or source event during the random walk what the contribution to a specified point would be if the next event were a transport of the particle directly to the point. Thus at each collision or source event a "pseudoparticle" is scored at the point detector with the following weight:

$$W = \frac{W_o p(\mu) e^{-\lambda}}{2\pi R^2}$$

where

W_o = the exit weight of the collision or source particle;

$p(\mu)$ = value of the probability density function for scatter directly toward the point detector;

$\lambda = \sum_i \Sigma_{ti} X_i$ where Σ_{ti} is the total cross section in region i and X_i is the path length of the pseudoparticle trajectory in region i. This sum represents the mean free path through all

the regions between the collision or source event and the detector.

$R = \sum (X_i)$ = the distance between the collision or source event and the detector.

Next event estimators are very expensive because the trajectory of the pseudoparticles must be followed for every contribution to the point detector in order to calculate the proper attenuation factor, $e^{-\lambda}$. Thus point detectors must be used sparingly.

Point detectors are a standard feature of most Monte Carlo codes and so more details will not be given here. However, MCNP has a number of interesting variations. First, there are a number of various detector roulette games which will be described later. Second, there is the ring detector in which the point detector is located on a ring around that geometry in order to take advantage of two dimensional symmetry whenever possible. The detector point on the ring is chosen preferentially close to the azimuthal location of the collision or source event in order to realize maximum efficiency. Finally, MCNP has the once-more-collided point detector estimator which avoids the $1/R^2$ singularity of traditional point detectors. The once-more-collided estimator is intended for use whenever the detector point is located in a scattering medium and the R term in the denominator is small. However, our experience is that whenever one is in a scattering medium that would require the once-more collided estimator there are usually far better ways of calculating the flux or other quantities of interest.

DXTRAN

DXTRAN is a unique MCNP variance reduction tool. A next event estimation is used to deterministically transport the uncollided weight from collision and source points to a spherical surface, known as a DXTRAN sphere, which is superimposed over the problem geometry. That is to say, DXTRAN quasi-deterministically calculates the weight that scatters and arrives without collision at the DXTRAN sphere. The random walk transport is then continued inside the DXTRAN sphere. In a sense DXTRAN is a form of angle biasing because at each collision particles are forced to go in the direction of the DXTRAN sphere. When isolated regions of a problem geometry are of great importance DXTRAN is a very powerful and successful technique. However, all the cautions and limitations of next event estimators, or point detectors, apply to DXTRAN. Also, DXTRAN becomes very tricky when there is more than one DXTRAN sphere in a problem because the DXTRAN spheres may interact with each other, and it is difficult to properly sample the scattering regions which contribute to them.

DETECTOR ROULETTE GAMES

The use of next event estimators for DXTRAN and point detectors is enhanced in MCNP by two detector roulette games. The first detector roulette game allows the user to specify the probability, P, of any geometric region contributing to the next event estimator. Only 1/P collisions in the region are allowed to make the expensive pseudoparticle contribution to the detector. This contri-

bution is then multiplied by P. The second detector roulette game is more elaborate. At the source or collision point, W_o, $p(\mu)$, and R are all known. If the quantity

$$Q = \frac{W_o p(\mu)}{2\pi R^2}$$

is small, then the contribution to the detector can only be smaller since $e^{-\lambda}$ is always less than unity. Thus if Q is smaller than either a predetermined value or a predetermined fraction of the average contribution to the detector so far then the pseudoparticle plays Russian roulette. In this way only significant contributions to the detector are made since the insignificant ones are killed or built up to significant size by the Russian roulette game. Both these detector roulette games typically enhance the efficiency of next event estimators by as much as an order of magnitude. However, care must be taken that the small contributions are unimportant. For example, a small but significant high energy tail of a result may be undersampled by improper use of these schemes.

WEIGHT WINDOWS

The weight window method, which is another form of splitting and Russian roulette, is very useful for many applications. As with importance sampling, the geometry is subdivided into reigons of different importances. Then each region is assigned a set of upper and lower weight window bounds. Particles with weights above the upper bounds are split so that their weight drops into the window; particles with weights below the lower bounds are rouletted so that the survivors are also in the window. Not only does this splitting and Russian roulette cause particles to migrate toward the tally region, but also it provides effective weight control which makes use of the exponential transform possible. Further, in MCNP the weight window method is usually the most effective means of energy biasing. Weight windows can be specified in a number of energy ranges. Additionally, the MCNP weight window allows the users to specify a splitting criteria so that splitting is never more than M for one and Russian roulette is never played with a survival probability of less than 1/M . The weight window game may be played either at surfaces, collisions or both. This flexibility in choosing where to play the weight window game, in choosing a splitting criteria, and in having energy dependence makes the weight window far more powerful than the more traditional combination of geometric splitting and Russian roulette and weight cutoff.

The principle difficulty of the weight window method is that now the user has an unmanageable number of input importance functions to provide the code. Fortunately, several schemes (refs. 1-4) have been devised in which the Monte Carlo code automatically selects the input importance parameters or weight windows for the user. We call such schemes "importance generators" or "weight window generators". Not only do such importance generators facilitate use of the weight window, but also they may be thought of as a "forward adjoint" solution which gives important information about other aspects of the problem model such as whether or not the geometric regions are sufficiently divided or too truncated.

ENERGY SPLITTING AND ROULETTE

Energy splitting and roulette is a means of biasing energy space. Although this feature is available in MCNP, its use is not recommended since the energy-dependent weight window does a better job.

FORCED COLLISIONS

Forced collisions may also be used to reduce variance by creating collided particles in regions where collisions are unlikely. These collided particles may then be used for point detectors or DXTRAN or other Monte Carlo processes where large numbers of collisions are desirable to more efficiently approach the problem solution. The MCNP forced collision algorithm causes particles entering a spatial region to be split into a collided and an uncollided part with the appropriate weight adjustments. The collided part may then play Russian roulette if desired in order to limit the number of additional particles.

CORRELATED SAMPLING

MCNP provides for correlated sampling to estimate the change in a quantity resulting from a small perturbation of any type in the problem. This technique enables the evaluation of small quantities that would otherwise be masked by the statistical errors of uncorrelated calculations. MCNP correlates a pair of runs by providing each new history in the unperturbed and perturbed problems with the same initial pseudo-random number and thus the same sequence of subsequent numbers until the perturbation causes the sequences to diverge. This may be thought of as a variance reduction method in that perturbation studies may be made without having to run problems to as fine a convergence as would have to be done with correlated sampling.

FUTURE VARIANCE REDUCTION METHODS

A number of variance reduction methods for future use in MCNP are presently under study at Los Alamos.

ANGLE BIAS

A "synergistic" method (ref. 6) of angle bias has been developed for photons which shows great promise. Previous attempts at angle bias have been foiled by instability problems. In this new method the exponential transform is used to just offset the instabilities of angle bias. Unfortunately, the method requires the on-line solution of a very difficult integral equation. Our investigation has so far been limited to photons because the integral is easier to solve than for neutrons.

The idea of using two unstable techniques to produce a powerful stable composite should be studied with respect to other variance reduction problems.

DXANG

The DXANG method (ref. 5) is a kind of collision biasing method which utilizes aspects of DXTRAN, space-energy-angle weight windows, and the weight window generator. In brief, the user defines spherical regions of interest, as with DXTRAN, which are not part of the problem, but rather are superimposed over the geometry. These spheres are called DXANG spheres. At each collision a cone is defined by the DXANG sphere and the collision point. If the collision occurs inside the DXANG sphere then the transport continues as usual. If the collision occurs outside the sphere, a special DXANG particle is created uniformly inside the cone with a weight adjustment dependent upon the probability density for scattering at the sampled angle. The extra weight created by DXANG particle is balanced by killing the original particle if it scatters into the cone. Otherwise, the original particle is sampled normally with no weight correction. Separate weight windows are kept for DXANG particles and non-DXANG (analog) particles. To simplify the selection of these weight windows a space-angle weight window generator is used.

UNIFORM SAMPLING OF SOURCE DISTRIBUTIONS

Tony Warnock of Cray Research, Inc., and Bob Schrandt of Los Alamos are presently studying a scheme which samples distributions by using a uniform sampling of the interval (0,1) rather than a random number sequence. The uniform sampling guarantees that an arbitrarily small probability interval will be hit a predetermined number of times with a given sample size. Preliminary results indicate that the uniform sampling takes longer to sample the source but does a better job of representing the source.

OTHER METHODS

There is still a lot of room for improvement in variance reduction techniques. Methods such as DXANG, angle biasing and other forms of collision biasing have yet to be fully developed, and there are probably many other techniques yet to be discovered. An example of an ambitious method presently under development is the Tom E. Booth's current attempt (ref. 7) to bias the random number space rather than physical space; if successful, this technique would make Monte Carlo variance reduction truly problem independent. (See Booth's paper, "A Monte Carlo Learning/Biasing Experiment With Intelligent Random Numbers" which is also being presented at this conference. Booth is also exploring continuous importance functions which are formed by infinite series expansion.

REFERENCES

1 T. E. Booth, "Automatic Importance Estimation in Forward Monte Carlo Calculations," Trans. Am. Nucl. Soc., 41, 308 (1982).

2 J. S. Hendricks, "A Code Generated Monte Carlo Importance Function," Trans. Am. Nucl. Soc., 41, 307 (1982).

3 T. E. Booth and J. S. Hendricks, "Deep Penetration by Monte Carlo," Trans. Am. Nucl. Soc., 43, 609 (1982).

[4] T. E. Booth and J. S. Hendricks, "Importance Estimation in Forward Monte Carlo Calculations," Nucl. Tech./Fusion, 5, 90 (Jan 1984).

[5] T. E. Booth, "A Weight Window/Importance Generator for Monte Carlo Streaming Problems," Proc. Sixth Int. Conf. Radiation Shielding, Tokyo, May 16-20, 1983.

[6] J. S. Hendricks and L. L. Carter, "Anisotrophic Angle Biasing of Photons," Nucl. Sci. Eng., 89, 118 (February 1985).

[7] T. E. Booth, "A Monte Carlo Learning/Biasing Experiment With Intelligent Random Numbers," submitted to Nuclear Science and Engineering.