Common Misconceptions in Monte Carlo Particle Transport

Thomas E. Booth LA-UR 11-03627

June 22, 2011

Abstract

Monte Carlo particle transport is often introduced primarily as a method to solve linear integral equations such as the Boltzmann transport equation. This paper discusses some common misconceptions about Monte Carlo methods that are often associated with an equation-based focus. Many of the misconceptions apply directly to standard Monte Carlo codes such as MCNP and some are worth noting so that one does not unnecessarily restrict future methods.

1 Introduction

Many, perhaps most, Monte Carlo code users understand Monte Carlo methods via linear integral equations. This is quite understandable as standard books in the field usually emphasize the connection of Monte Carlo transport and the transport equation. This connection has proven very useful both for teaching about Monte Carlo and for developing and analyzing many variance reduction methods. The success of the books also indicates that the readers have found the transport equation approach to understanding Monte Carlo transport useful. Indeed, the fact that so many Monte Carlo books emphasize the transport equation indicates that the experts writing the books have found the transport equation a very useful perspective both in practice as well as in teaching.

On the other hand, viewing Monte Carlo primarily as a way to solve the transport equation gives rise to a number of common misconceptions in the following areas

1. Solving a problem is synonymous with solving an equation.
2. Monte Carlo transport and solving “the” transport equation.
3. Monte Carlo and S_N Boltzmann transport solutions can be combined to get pulse height tally estimates.
4. Monte Carlo and probability of initiation estimates.
5. Monte Carlo particle transport is a Markov process.

6. Weight represents a number of particles.
7. Weight is always positive.
8. A necessary condition for unbiased estimates is that the particle density is preserved.
9. A sufficient condition for unbiased estimates is that the particle density is preserved.
10. A particle has one weight.
11. Many ways to get a zero variance calculation.
12. Splitting and roulette concepts and information collection.
13. Zero variance, importance, and optimal weight windows.

2 Comments on "Analog" Monte Carlo and the Transport Process

Probably most Monte Carlo transport practitioners understand and use the term "analog Monte Carlo" in mostly the same way. That is, convenient probability densities are abstracted from the physical transport process. These convenient probability densities are then embedded in a transport code. For example, the neutron distance to collision is sampled conveniently from an exponential distribution without modeling the detailed interactions between the neutron and each nuclide along its path.

For this paper, the term "analog" describes a direct sampling of these abstracted probability densities. For the most part, people have abstracted very similar probability densities from the physical transport process. Nonetheless, it is probably worthwhile to note that an analog sampling in the context of this paper refers to the particular probability densities that MCNP [1] has abstracted from the physical transport process. Roughly speaking, an analog Monte Carlo sampling of a particle transport problem in MCNP is what one gets when no variance reduction techniques are used.

The term "transport process" is also used in the context of MCNP's abstraction of the physical transport process. An analog transport process is an analog simulation of the abstracted physical transport process. Similarly, a nonanalog transport process is a nonanalog simulation of the abstracted physical transport process.

3 The Educational Bias of Equation Solving

From junior high onward (10-15 years), solving a technical problem is usually presented as almost synonymous with solving an equation. This conflation of "solving a problem" and "solving an equation" is unnecessary and often misleading with respect to Monte Carlo transport calculations. There is nothing wrong with noting the relationship of a Monte Carlo transport simulation to a transport equation. The tendency in many Monte Carlo explanations, however, is to make numerous assumptions that severely restrict the Monte Carlo techniques considered. The restricted Monte Carlo simulation is then sometimes unduly forced to look more similar to the transport equation than necessary or prudent. The restricted Monte Carlo often is easier to explain because it is simpler than without the assumptions. Usually, the assumptions are either poorly discussed or not even explicitly mentioned. When mentioned at all, there is often no indication that a Monte Carlo calculation can be done *without* the assumptions.

The conflation of "solving a problem" and "solving an equation" is so prevalent that Monte Carlo is often misleadingly described simply as a way to solve the transport equation rather than the broader set of transport problems that the codes can solve. For example, comments from an introductory Los Alamos lecture course slide are [2]

> Two basic ways to approach the use of Monte Carlo methods for solving the transport equation:
>
> - Mathematical technique for numerical integration
> - Computer simulation of a physical process
>
> Each is "correct."
>
> - Mathematical approach is useful for: Importance sampling, convergence, variance reduction, random sampling techniques, eigenvalue calculation schemes, . . .
> - Simulation approach is useful for: collision physics, tracking, tallying, . . .

This slide is confusing for two reasons. First, there are many transport problems that Monte Carlo codes can solve that are not described by the transport equation. Second, many people would erroneously conclude that the simulation approach is not useful for

- Importance sampling, convergence, variance reduction, random sampling techniques, eigenvalue calculation schemes, . . .

The simulation approach *is* useful for *all* of the items above. As an example, the weight window and weight window generator approaches were developed [3] solely via a simulation approach with no use of the transport equation. Another slide [2] with the title "Monte Carlo & Transport Equation," lists as one of several assumptions

> - Markovian - next event depends only on current $(\mathbf{r}, \mathbf{v}, t)$, not on previous events.

Note that although the transport problem *could* be simulated with Markovian Monte Carlo, there is no reason to restrict either the Monte Carlo, or the user's thinking, to Markovian Monte Carlo simply because the kernels in the transport equation only depend on the current $(\mathbf{r}, \mathbf{v}, t)$. Most Monte Carlo codes allow non-Markovian techniques. One of the most commonly used and effective variance reduction techniques is the weight window and the sampling is non-Markovian because the sampling depends on the particle weight as well as $(\mathbf{r}, \mathbf{v}, t)$. By *default* MCNP calculations are non-Markovian because a weight cutoff (roulette) game is played as the default option.

Note that a Monte Carlo code can be written directly from the abstracted physical process without ever even considering a transport equation. Average particle behavior in the Monte Carlo process certainly is described by the transport equation, just as a ball's motion is described by Newton's equation. But as a philosophical matter, saying that Monte Carlo is "solving" the transport equation seems a bit like saying that a ball is "solving" Newton's equation.

4 Monte Carlo Transport and Solving "The" Transport Equation

It is often said that Monte Carlo solves the transport equation. This is an interesting statement because the transport equation is usually derived by averaging over the abstracted transport process, which is just the analog Monte Carlo process. The transport equation is farther removed from the physical transport process because the transport equation is a specific average over the transport process. There are many different possible averages over the transport process that result in many different transport equations. One should understand "the" transport equation as convenient, but loose, terminology and not associate any special uniqueness because "the" precedes "transport equation".

A lot of information is lost in the averaging process and, in most cases, Monte Carlo codes allow estimation of quantities for which the transport equations displayed in the literature do not apply. In particular, the typical transport equations totally ignore the correlation between particles. Thus any estimate, such as the pulse height tally in MCNP, that depends on the correlation between particles is not described by the standard transport equation.

Transport equations, of course, can be written to include correlation between particles, but authors typically choose not to display such equations. If a Monte Carlo user wishes to use transport equations to analyze and/or improve his Monte Carlo calculation, it is important to understand what transport equations are relevant to the calculation. This last statement seems obvious, but people have sometimes talked about the pulse height tally in MCNP in the same breath as a transport equation that ignores the correlation between particles.

5 The Transport Equation, Monte Carlo Codes, and Collective Particle Estimates

As mentioned in the previous section, not all transport problems can be solved by solving the transport equation. Any estimates that depend on a correlated collection of particles cannot be obtained by solving the transport equation because the transport equation does not consider the correlation. Even in Monte Carlo codes, problems can arise if the physics modeled is not microscopically correct.

In MCNP, for instance, when a neutron collides with a nuclide there is an average gamma production distribution that is averaged over all possible neutron interactions with the nuclide. Thus, the gamma production is uncorrelated with which neutron interaction is sampled. Thus any estimate that depends on the joint transport of the neutron and the induced photon cannot be correct because the necessary correlation is absent. This averaging is an unbiased procedure when the desired estimates (e.g. photon fluences) do not depend on the correlation between the neutron and the neutron-induced photons.

When the Monte Carlo simulation is microscopically correct, many collective particle estimates can be made by analog Monte Carlo simulations in standard Monte Carlo codes. Inasmuch as it is *collections* of particles that tally, any statistical weights must be associated with the *collections* rather than individual particles. Doing a microscopically correct analog Monte Carlo simulation amounts to assigning a statistical weight of 1 to the collection. Sometimes such analog simulations are either simply impossible or too computer intensive.

5.1 Probability of Extinction

Estimating the probability of extinction is a good example of a simulation that is impossible to do with analog Monte Carlo. In its simplest form, the probability of extinction (poe) problem consists of dropping a neutron into a supercritical system. There is some probability that an infinite (divergent) nuclear chain results and some probability that the chain becomes extinct. Because it is impossible for a computer to ever finish simulating an infinite chain, some people erroneously believed that Monte Carlo could not obtain statistically exact poe estimates. It is worth noting that approximate Monte Carlo poe methods have been implemented in Monte Carlo codes (e.g. [4–6]). It is not clear whether this erroneous belief has always been the reason for terminating an *apparently* divergent chain according to some approximate criterion (e.g. when the chain length exceeds 10,000). Note that statistically exact poe estimates are possible [7, 8], provided nonanalog Monte Carlo methods are applied to the neutron *chains* rather than to the *individual* neutrons. The transport equation is irrelevant to the poe problem.

5.2 Pulse Height and Coincidence Tallies

Many estimates (e.g. pulse height tallies and coincidence tallies) that depend on collections of particles, although theoretically possible to do with analog Monte Carlo, are extremely computer intensive. Nonanalog Monte Carlo methods (e.g. [9–15]) can often solve the computer time problem. The nonanalog methods must be applied to the *collections* of particles that contribute to the desired estimate. The transport equation is irrelevant to these collective particle tallies (except in the special case when the collection consists of one particle.)

The insidious focus on "the" transport equation rather than the transport problem sometimes produces interesting claims. Despite the fact that "the" transport equation is not correct for pulse height tallies, there are claims (e.g. see [16, 17]) that S_N codes can be used to obtain pulse height tallies such as the MCNP F8 tally. There appears to be no cautions about the approximations made when pulse height tallies are obtained with Boltzmann transport codes that do not have correlated particle information. Instead, there sometimes seems to be a confusion about what Monte Carlo codes do and do not do. In particular, [17] comments

> Monte Carlo radiation transport codes solve the Boltzmann transport equation by faithfully simulating the life histories of radiation particles or photons as they interact in the underlying physical medium.

This is a very curious statement because MCNP uses a non-Boltzmann transport simulation to get the pulse height tally. That is, the MCNP pulse height tally used in [17] as a basis for comparison does *not* use a Boltzmann transport simulation.

A coupled S_N to Monte Carlo approach is given in [16] and the entire calculation is an S_N calculation in [17]. Sometimes these claims are even empirically validated by direct comparison with the MCNP F8 tally. The problem is that the pulse height tally depends on correlation between particles that the S_N codes ignore. Good comparisons with the F8 tally can be obtained only when it is unlikely that two correlated particles will both reach the detector.

Consider a simple correlation example. MCNP needs to include the correlation between photons (e.g. from pair production and double fluorescence) in order to get theoretically correct (within statistics) estimates of pulse height tallies. For instance, if both 0.511 MeV photons from pair production get absorbed in the detector then MCNP needs the correlation so that one hit in the 1.022 MeV energy bin is tallied (rather than two hits in the 0.511 MeV bin, which would be physically incorrect). Neither [16] nor [17] make it clear what happens in the deterministic calculations if both photons from a pair production get absorbed in the detector. How does a deterministic Boltzmann code distinguish between two correlated photons from a pair annihilation versus two independent photons?

6 Markov and Nonmarkov Processes and Random Walks

In an analog simulation of nature, the next step of a particle's random walk depends only on its current phase space location P (usually position, direction, energy, and time). That is, an analog Monte Carlo process, like the physical process, is a Markov process.

Nonanalog simulations of particle transport depart, in one way or another, from the analog process. Nonanalog methods are also known as variance reduction methods because the intent of using nonanalog methods is to reduce the variance in the estimated mean for a given computer time. Note that nonanalog simulations need not be Markov processes.

Most theoretical Monte Carlo discussions assume that a particle's random walk is independent of the particle's weight. Under this assumption, a particle's score is directly proportional to its weight and the r^{th} score moment for a particle of weight w is w^r times the r^{th} score moment for a unit weight particle. [18, page 163]. To give some idea of the common appeal of this wide-reaching assumption, note that [18] first mentions this assumption in a footnote.

A cautionary note is perhaps worthwhile here. Because weight independent (natural) Markov simulations are more tractable mathematically, they account for almost all of the theoretical discussions in the Monte Carlo literature. One should not be mislead into concluding that weight independent simulations are more important, better, or more widely used than weight dependent simulations. Many of the large production Monte Carlo codes allow weight dependent simulation. MCNP, which is probably the most widely used Monte Carlo transport code in the world, has always done weight dependent simulation as a default.

Once one breaks free of thinking about Monte Carlo as a way of simulating equations with Markovian kernels, the types of possible nonanalog techniques expand dramatically[19]. Of course, not all nonanalog techniques will reduce variance, but it is worthwhile to understand the flexibility of nonanalog Monte Carlo. Examples that surprise some people are now given. For simplicity, the examples are weight independent nonmarkov processes, because the mathematics is easier. (Similar weight dependent examples are possible.)

Consider a simple slab penetration problem in a nonmultiplying medium and score the particle weight w if the particle penetrates and 0 otherwise. Consider a particle of weight $w = 1$ entering its fourth collision in a material composed of two nuclides. Define

1. ρ_i = the probability of collision with nuclide i ($\rho_1 + \rho_2 = 1$).

2. b_i = the biased probability of with nuclide i ($b_1 + b_2 = 1$).

3. $p_i(s, w_i)ds$ = the probability that a particle of weight w_i entering collision with nuclide i will subsequently score in ds about s.

The expected score for an analog sampling of the nuclide is

$$\left\langle S_{analog}\right\rangle = \rho_1 \int p_1(s,1)s\ ds + \rho_2 \int p_2(s,1)s\ ds \tag{1}$$

For a biased sampling of the collision nuclide, the particle weight is multiplied by the ratio of the true probability to the sampled probability, so

$$w_i = \frac{\rho_i}{b_i} \tag{2}$$

The expected score for this biased sampling of the nuclide is

$$\left\langle S_{bias}\right\rangle = b_1 \int p_1(s,w_1)s\ ds + b_2 \int p_2(s,w_2)s\ ds \tag{3}$$

For this example, if one assumes weight independent random walks, then for any random walk a particle of weight w scores w times what a particle of weight 1 scores. That is,

$$p_i(s,w)ds = p_i(s/w,1)d(s/w) \tag{4}$$

Using Eq. 4 in Eq. 3

$$\left\langle S_{bias}\right\rangle = b_1 \int p_1(s/w_1,1)s\ d(s/w_1) + b_2 \int p_2(s/w_2,1)s\ d(s/w_2) \tag{5}$$

Letting $s_i = s/w_i$ and noting Eq. 1

$$\begin{aligned}\left\langle S_{bias}\right\rangle &= b_1 \int p_1(s_1,1)w_1 s_1\ ds_1 + b_2 \int p_2(s_2,1)w_2 s_2\ ds_2 \\ &= \rho_1 \int p_1(s,1)s\ ds + \rho_2 \int p_2(s,1)s\ ds = \left\langle S_{analog}\right\rangle \end{aligned} \tag{6}$$

so that the mean score is preserved.

Other than being a discrete probability density, the b_i are completely arbitrary. Suppose, for instance, that η was the random number used to select the collision nuclide in the particle's first collision and let

$$b_1 = \eta p_1 \tag{7}$$

$$b_2 = 1 - b_1 \tag{8}$$

be the biased probabilities for selecting the nuclide i on the fourth collision. Eq. 6 still shows that the mean score is preserved even though the sampling is no longer Markovian because the sampling depends on more than just the preceding collision.

Somewhat more interesting is that the sampling of the collision nuclide on the fourth collision can be made to depend on the nuclide sampling for the seventh collision. For example, suppose that before the fourth collision a random number ξ is generated to be used later to sample the collision nuclide for the seventh collision. (If there are fewer than seven collisions, then ξ will not, of course, be used to sample the collision nuclide.) Because the distribution of scores at the fourth collision now depends on ξ, define

- $p_i(s, w_i|\xi)ds$ = the probability that a particle of weight w_i entering collision with nuclide i will subsequently score in ds about s, given that the collision nuclide for the seventh collision is selected using ξ.

Now let the b_i depend on ξ. For a biased sampling of the collision nuclide, the particle weight is the ratio of the true probability to the sampled probability

$$w_i(\xi) = \frac{\rho_i}{b_i(\xi)} \tag{9}$$

The expected score (conditional on ξ) for this biased sampling of the nuclide is

$$\langle S_{bias}|\xi\rangle = b_1(\xi)\int p_1(s, w_1(\xi)|\xi)s\ ds + b_2(\xi)\int p_2(s, w_2(\xi)|\xi)s\ ds \tag{10}$$

If one assumes weight independent random walks, then for any random walk, a particle of weight w scores w times what a particle of weight 1 scores. That is,

$$p_i(s, w|\xi)ds = p_i(s/w, 1|\xi)d(s/w) \tag{11}$$

Using Eq. 11 in Eq. 10

$$\begin{aligned}\langle S_{bias}|\xi\rangle = b_1(\xi)\int p_1(s/w_1(\xi), 1|\xi)s\ d(s/w_1(\xi)) +\\ b_2(\xi)\int p_2(s/w_2(\xi), 1|\xi)s\ d(s/w_2(\xi))\end{aligned} \tag{12}$$

Letting $s_i = s/w_i(\xi)$, Eq. 12 becomes

$$\begin{aligned}\langle S_{bias}|\xi\rangle = b_1(\xi)\int p_1(s_1, 1|\xi)s_1 w_1(\xi)\ ds_1 +\\ b_2(\xi)\int p_2(s_2, 1|\xi)s_2 w_2(\xi)\ ds_2\end{aligned} \tag{13}$$

Using Eq. 9, Eq. 13 becomes

$$\langle S_{bias}|\xi\rangle = \rho_1\int p_1(s_1, 1|\xi)s_1\ ds_1 + \rho_2\int p_2(s_2, 1|\xi)s_2\ ds_2 \tag{14}$$

Noting that

$$p_i(s_i, 1) = \int p_i(s_i, 1|\xi)d\xi \tag{15}$$

and integrating over ξ

$$\begin{aligned}\langle S_{bias}\rangle = \int\langle S_{bias}|\xi\rangle d\xi = \rho_1\int p_1(s_1, 1)s_1\ ds_1 + \rho_2\int p_2(s_2, 1)s_2\ ds_2\\ = \langle S_{analog}\rangle\end{aligned} \tag{16}$$

so that the mean score is preserved. The fact that the fourth collision nuclide sampling can depend on the sampling of the seventh collision nuclide surprises many people (it even seems to bother some).

7 Comments on Statistical Weight

From the standpoint of many, perhaps most, major Monte Carlo transport codes, weight is a particle attribute, like energy and position. That is, the weight is carried along with the particle, banked with the particle, and so forth. Weight is simply a tally multiplier. It is often convenient to interpret the weight as the number of physical particles represented by the computer particle. Heuristically, one expects that if the Monte Carlo process preserves the expected weight at each event, then the result will be an unbiased mean. For the most part, this is a very useful view of the Monte Carlo process, but it is perhaps useful to point out some cases for which this view needs some elaboration and/or modification. The purpose here is to illustrate some of the subtleties in the concept of "particle weight." Note that many of the examples are not very realistic, they are contrived for simplicity.

7.1 Preserving the Expected Weight is not Always a Sufficient Condition for an Unbiased Mean

Preserving the expected weight, by itself, will not ensure an unbiased estimate. The estimator must depend on weight in a correct way also. As an obvious example, if the number of particles crossing a surface is desired, then tallying "1" (regardless of weight) every time a particle crosses the surface will give the correct tally for an analog calculation, but will in general be wrong when variance reduction techniques change the weight.

For deterministic (nonrandom) estimators, unbiasedness is normally assured by making the tally function proportional to weight. Not all common estimators are deterministic. The point detector in MCNP[1, 3-106] is a random estimator because it plays roulette games when the optical path to the detector gets large. For random estimators, one requires that the expected tally (rather than the individual tally itself) be proportional to weight.

The conceptual mistake many people make is to separate the estimation process from the transport process. These two processes can be tied rather intimately in some unusual ways and one has to ensure that the combined process is unbiased. Consider estimating the number of particles that cross the cell shown in Fig. 1 without colliding.

A typical transport and estimation procedure is: with probability $p = exp(-\Sigma T)$ the particle crosses the cell without collision and tallies w and with probability $1-p$ the particle collides and no tally is made. The particle is then followed from either the point where it crossed the surface or the point where it collided.

Another possible way to estimate the number of collisionless flights across the cell is shown in Fig. 2.

For Fig. 2, the tally is not dependent on whether the particle of weight w collides or not. Instead, the estimation is done using a "pseudoparticle" that only exists for the estimation procedure. (The pseudoparticle initially has the same phase space coordinates as the transported particle.) The pseudoparticle is sampled using the same probabilities as the transported particle, but the pseudoparticle is terminated after the estimation procedure is completed, it is not transported. Hence the term pseudoparticle, because it is not part of the transport. Transport then continues with the original particle. Thus the particle might not cross the surface without colliding, but it might contribute to the tally because the pseudoparticle did cross the surface without colliding. Note that if the estimation process used on the pseudoparticle is not correct, then the estimate can be erroneous despite the fact that the expected particle weight has been preserved. It is often so obvious how the pseudoparticle should be treated that the pseudoparticle's role in maintaining an unbiased estimate is not discussed. (One correct method tallies w when the pseudoparticle crosses the surface without collision.)

7.2 Preserving the Expected Weight is not Always a Necessary Condition for an Unbiased Mean

The previous subsection showed that preserving the expected weight is not always a sufficient condition for an unbiased mean. Now, it is shown that preserving the expected weight is not always a necessary condition for an unbiased mean. Experienced Monte Carlo practitioners correctly might suspect some legerdemain here. Consider Fig. 2 again. Inasmuch as the tally depends (for the current transport step) not on the particle's weight, but on the weight associated with the pseudoparticle, the particle weight can be set to any arbitrary value, provided the particle weight is returned to w when the particle collides or crosses the surface. Thus, preserving the expected weight is not necessary for this step in the transport process. With the tally not responding to the original particle, one possible interpretation is that the particle weight is zero for that step. Things will get even more curious in the next subsection.

7.3 Multiple Particle Weights

Particle weight is normally conceived of as a single value for each particle. Not only can one conceive of particles having multiple weights, multiple weights are used in some production transport codes. Before jumping to the practical uses of multiple weights, two simple examples are discussed.

Building on the previous two subsections, suppose that the code uses two *different* estimators for the number of particles crossing the surface. The first estimator uses the original particle as in Fig. 1 and the second estimator uses the pseudoparticle as in Fig. 2. In this case, the original particle should have weight w so that the first estimator is correct, but it can still have zero weight for the second estimator. That is, the particle can have a different weight for each estimator.

For another simple example, suppose that a particle of weight w reaches a surface as shown in Fig. 1. Upon crossing the surface, split the particle into two particles each of the original weight w. The total expected weight is not preserved by this split, but unbiased estimates can again be made by a bit of legerdemain with the estimators. Label the particles 1 and 2. Label the estimators with positive integers. Let the odd numbered estimators respond only to particle 1 and let the even numbered estimators respond only to particle 2. This can be viewed as follows. The presplit particle contributed to all tallies and thus can be considered to have a weight vector (w, w). After the split, particle 1 has weight vector $(w, 0)$ and particle 2 has weight vector $(0, w)$.

Turning to practical uses of multiple weights, note that perturbation and correlated sampling methods use different weights for the reference system and the perturbed system. For example, Ref. [18, page 307] explicitly uses a weight vector in the discussion of correlated sampling.

The dxtran method in MCNP is very similar to the second example above. Upon surviving a collision, a particle is partitioned into two particles. The "dxtran particle" represents the uncollided particles that arrive on a user specified dxtran sphere. The "nondxtran particle" represents the remainder of the particles. Note that the nondxtran particle has the original weight, w, at the collision exit point and the dxtran particle has a nonzero weight. Thus, the total particle weight is always larger than w at the collision exit point. The trick here is that the dxtran particle has zero weight for any tallies made before crossing the dxtran sphere and appropriate weight for any tallies afterward. Conversely, the nondxtran particle has weight w for all tallies made before crossing the dxtran sphere and zero weight for any tallies afterward.

Multiple weights can also be used to get low variance estimates for multiple tallies. Consider a particle with a single weight in a slab penetration problem. Suppose the numbers of particles exiting the slab in the three energy ranges 1.00 to 1.01, 1.01 to 1.02, and 1.02 to 1.03 MeV are desired. Note that a typical zero variance sampling for the energy range 1.00 to 1.01 MeV means that every particle has to exit the slab within this energy range. This means that no particles exit in the other two energy ranges. Thus, a random walk process that gives a zero variance estimate for one energy range gives an infinite variance estimate for all other energy ranges. Most of the sampling to get a zero variance solution in one interval is going to be very similar to the sampling to get a zero variance solution in either of the other two intervals. It seems ridiculous that a zero variance solution in one interval forces an infinite variance in the other intervals. Reference [20] shows that it is possible to get zero variance solutions in all three intervals at once using particles that carry three weights. The method works by simultaneously applying several different importance functions, one for each tally, in a correlated way. Although zero variance estimates are impractical because the importance functions are not known exactly, low variance solutions are possible with approximate importance functions. The method in [20] follows a single particle with multiple nonzero weights until the correlation between the importance functions decreases enough that the particle must, statistically, execute different random walks for different tallies.

7.4 Nonpositive Particle Weight

People often think of weight as a nonnegative number, but many nonanalog techniques work with negative weights as well as mixtures of negative and positive weights. Indeed, even complex weights can in principle be used. A trivial, if not especially useful, example concerns splitting. A particle track of weight w crossing a surface can be split into pieces in a number of ways. A few possibilities are (note $i^2 = -1$)

1. two particles with weights $w/2$ and $w/2$
2. three particles with weights $w/2$, $w/4$, and $w/4$
3. four particles with weights $w/2$, $w/2$, $w/4$, and $-w/4$
4. four particles with weights $w/2$, $w/2$, $iw/4$, and $-iw/4$

Another more complicated splitting procedure, somewhat similar to item 3, will be discussed later.

Although the splitting example is totally contrived, negative weights occur much more naturally in nonfundamental mode eigenfunction estimates. In criticality calculations, the first (fundamental ψ_1) eigenfunction is everywhere positive, but the second eigenfunction (ψ_2) must have some regions where $\psi_2 < 0$ and some regions where $\psi_2 > 0$ because the eigenfunctions are orthogonal and require $\int \psi_1(P)\psi_2(P)dP = 0$. The regions where $\psi_2 < 0$ are represented by negative weight particles. Note that when the eigenfunctions are complex, then complex particle weights can be used to represent the eigenfunctions.

In residual or "reduced source" methods, one uses Monte Carlo to estimate the difference between the exact solution and an approximate solution. Let L be a linear transport operator and $Q(P)$ be the source. That is,

$$L\big[\psi(P)\big] = Q(P) \tag{17}$$

Now suppose that one has an estimate, $\psi_0(P)$, and one defines the difference function $D(P)$ by

$$\psi_0(P) = \psi(P) - D(P) \tag{18}$$

Inserting Eq. 18 into Eq. 17 yields

$$L\big[\psi_0(P) + D(P)\big] = Q(P) \tag{19}$$

$$L\big[D(P)\big] = Q(P) - L\big[\psi_0(P)\big] \tag{20}$$

Defining a "reduced source"

$$R(P) = Q(P) - L\big[\psi_0(P)\big] \tag{21}$$

then the resulting new transport problem is

$$L\big[D(P)\big] = R(P) \tag{22}$$

where now the source, $R(P)$, is partly positive and partly negative and is naturally represented by positive and negative weight particles.

The author does not know of any standard transport code that uses the following trick, but it might be useful for some difficult sampling issues. In any case, it *is* possible and illustrates another case where negative weights might be used. Suppose that one needs to sample an angle $0 \leq \theta \leq 2\pi$ from a probability density $p(\theta)$ that is very difficult and time consuming to sample. Define

1. $p(\theta) =$ true probability density that is difficult to sample
2. $f(\mathbf{r}, w|\theta) =$ score produced by a particle at angle θ with the random walk sampled from the random vector $\mathbf{r}$ selected from the unit random number generator $u(\mathbf{r}) = 1$.
3. $q(\theta) =$ probability density that approximates $p(\theta)$ and is easy to sample.

Suppose the average score generated from the current point onward over all possible θ and subsequent random walks is desired. That is,

$$\int \int f(\mathbf{r}, w|\theta)p(\theta)d\theta d\mathbf{r} \tag{23}$$

is desired. The sampling can be split into two parts by noting

$$\int \int f(\mathbf{r}, w|\theta)p(\theta)d\theta d\mathbf{r} = \int \int f(\mathbf{r}, w|\theta)q(\theta)d\theta d\mathbf{r} + \int \int f(\mathbf{r}, \mathbf{w}|\theta)(p(\theta) - q(\theta))d\theta d\mathbf{r} \tag{24}$$

An estimate of

$$\int \int f(\mathbf{r}, w|\theta)q(\theta)d\theta d\mathbf{r} \tag{25}$$

can be made with a particle of weight w by sampling θ from $q(\theta)$, sampling the random walk specified by $\mathbf{r}$, and and scoring $f(\mathbf{r}, w|\theta)$.

For an unbiased mean, the expected score has to be proportional to the particle weight. That is,

$$(1/w) \int f(\mathbf{r}, w|\theta)d\mathbf{r} = (1/w') \int f(\mathbf{r}, w'|\theta)d\mathbf{r} = \int f(\mathbf{r}, 1|\theta)d\mathbf{r} \tag{26}$$

In particular,

$$\int f(\mathbf{r}, w)|\theta)(p(\theta) - q(\theta))d\mathbf{r} = \int f(\mathbf{r}, w(p(\theta) - q(\theta)))|\theta)d\mathbf{r} \tag{27}$$

so that the last term in Eq. 24 is

$$\int \int f(\mathbf{r}, w)|\theta)(p(\theta) - q(\theta))d\theta d\mathbf{r} = \int \int f(\mathbf{r}, w(p(\theta) - q(\theta)))|\theta)d\theta d\mathbf{r} \tag{28}$$

This can be estimated by sampling θ from the probability density $1/(2\pi)$ and following a particle of weight $w' = w(p(\theta) - q(\theta))$ along the random walk specified by $\mathbf{r}$, and and scoring $f(\mathbf{r}, w(p(\theta) - q(\theta))|\theta)$. Note that w' will sometimes be negative and sometimes be positive.

Alternatively, because $q(\theta)$ is easy to sample, one could alter the procedure. Instead of sampling from the probability density $1/(2\pi)$ one could sample from $q(\theta)$ and multiply by the additional weight factor

$$c(\theta) = \frac{1/(2\pi)}{q(\theta)} = \frac{1}{2\pi q(\theta)} \tag{29}$$

That is, with weight

$$w'(\theta) = w\frac{(p(\theta) - q(\theta)))}{2\pi q(\theta)} \tag{30}$$

the right side of Eq. 27 can be expressed as

$$\int\int f(\mathbf{r}, w'(\theta)|\theta) q(\theta) d\theta d\mathbf{r} \tag{31}$$

This can be estimated by sampling θ from the probability density $q(\theta)$ and following a particle of weight $w'(\theta)$ along the random walk specified by $\mathbf{r}$, and and scoring $f(\mathbf{r}, w'|\theta)$. Note that w' will sometimes be negative and sometimes be positive.

8 Zero Variance Misconceptions

Zero variance Monte Carlo schemes have been around almost since the beginning of Monte Carlo. A fairly recent paper by Hoogenboom [21] both notes that the schemes go as far back as Kahn's [22] work and claims

> We will prove that for a given estimator there is only one zero-variance scheme possible with a unique biasing of the source function and the transition and collision kernels.

Hoogenboom's proof is an interesting and worthwhile result that follows in the general tradition of zero variance proofs that make numerous assumptions about what a Monte Carlo transport calculation must look like. Some typical, often unstated, assumptions are:

1. If S is the set of possible random walks when no biasing is used, then S is the set of possible random walks in the zero variance biasing schemes. That is, the Monte Carlo techniques considered change only the *probabilities* of the random walks. The set S of possible random walks is unaltered.

2. No splitting techniques can be used in the zero variance scheme.

3. None of the biased kernels can depend on particle weight.

4. The biased kernels are Markovian.

5. The biased kernels depend solely on the state of a given particle. That is, it is not allowed that biasing the kernels of track 20 of history N might depend on the state of track 4 of history N.

6. Combing techniques that take K input tracks within history N and "comb" them into M tracks within history N are not allowed.

7. There is only one track per history.

Hoogenboom's uniqueness claim should be interpreted in the proper context intended in [21]. By violating one or more of these assumptions, it is easy to demonstrate *another* zero variance scheme. For instance, one can make a simple alteration to the zero variance scheme in [21] by introducing a 2:1 split and then sampling each of the split particles applying Hoogenboom's same zero variance sampling procedures to each of the split particles.

One negative consequence of tying zero variance procedures so closely to particular froms of the transport equation is that the essential requirement for zero variance procedures across many different Monte Carlo fields is obscured. Each Monte Carlo field tends to use very special techniques in that field and ignores the generalities both within and across Monte Carlo fields.

There is nothing special about a transport problem that requires playing with transport equations to get zero variance solutions to transport problems. Zero variance is a quite general property of linear Monte Carlo problems. Playing with transport equations ignores this generality, just as playing with other equations in other fields to get zero variance solutions ignores the commonality with transport Monte Carlo.

The real key to zero variance solutions for any linear Monte Carlo calculation is to expected score weight all decisions. One *specific* simple example of expected score weighting all decisions is given in [23]. Random numbers determine decisions in a Monte Carlo code, so [23] shows that expected score weighting the selection of random numbers produces a zero variance solution to any linear Monte Carlo problem. For example, note that an MCNP calculation using any set of MCNP variance reduction techniques and any tally is still a linear Monte Carlo problem, so score weighting the random numbers will produce a zero variance solution.

Zero variance concepts and procedures are far richer and more general than the way they are usually introduced in transport Zero variance procedures not only exist for nonBoltzmann transport problems such as the probability of initiation problem [7], they exist for all linear Monte Carlo problems in any field, and for any desired estimate.

9 Splitting and Rouletting

Many people view splitting and rouletting primarily in the context of an importance function: split when the particle's importance increases and roulette when the particle's importance decreases. This is often a useful view, but it helps to have a deeper conceptual insight to avoid instances when it is not a useful view.

A Monte Carlo code collects information about desired quantities (i.e. tallies) by following particle histories. Without the particles, very little can be known about the tallies because it is the particles that are responsible for collecting information. Variance reduction techniques are designed to collect more information in the same amount of computer time. One can collect more information by getting more information per history and/or by using more histories. Splitting works by collecting more information per history. Roulette works by terminating particles that are not collecting enough information and using the time saved to increase the number of histories that can be run in a fixed computer time.

Consider splitting first. A physical particle will have one next event. For example, a physical particle in free-flight cannot have its next collision in three locations. In an analog Monte Carlo simulation a particle can likewise only have one next event. Splitting is a mathematical artifice that collects information on more than one possible next event. For example, splitting a computer particle about to undergo a distance to collision sampling into 3 identical particles allows investigation of three possible outcomes of the sampling instead of one possible outcome. That is, extra information is being collected because of the splitting. The extra information collected usually reduces the sample variance, but having to following the extra split particles decreases the number of histories that can be run in a fixed computer time.

Now consider roulette. Roulette usually *increases* the sample variance (σ^2) because a terminated particle collects no further information. So, in what sense is roulette a variance *reduction* technique? Roulette is intended to decrease the variance of the mean

$$\sigma_m^2 = \frac{\sigma^2}{N} \tag{32}$$

by increasing the number of histories N that can be run in a fixed computer time. (The distribution of means is known to be a normal distribution with variance σ^2/N via the central limit theorem.)

In summary, splitting hopes to decrease σ^2 more than splitting decreases N and roulette hopes to increase N more than roulette increases σ^2 so that σ_m^2 will decrease. Splitting will be counterproductive when there is not enough extra information to be gained to justify the cost of following all the split particles. This is the reason that Monte Carlo codes generally avoid splitting particles going into a void even if the void has a relatively high importance. All split particles crossing a void reach the other side of the void without collision, so there is no information gained to justify the added time required. Roulette will be counterproductive when the information lost is large compared with the time saved. Note that when a particle is terminated by roulette that the information lost is the same whether the survival probability is 1/10 or 1/100. Suppose that continuing the particle track takes, on average, computer time t. Rouletting 1/10 saves $0.9t$ and rouletting 1/100 saves $0.99t$. Note that the more aggressive 1/100 roulette game only saves $0.09t$ more time than the 1/10 roulette game yet increases the statistical weight by ten times over the 1/10 roulette game. This often increases σ^2 more than N increases.

10 Optimal Weight Windows, Zero Variance, and Importance

The previous section explained the general workings of splitting and roulette. This section discusses some of these ideas in the context of the weight window technique.

The weight window is a splitting and roulette technique that depends on the particle's weight and phase-space location. A weight window consists of three weight levels in each phase-space (e.g. space-energy) region

1. a lower weight bound w_l

2. an upper weight bound w_u (it is required that $w_u \geq 2w_l$ and here we assume the default $w_u = 5w_l$)

3. a Russian roulette survival weight w_s

Let a particle have weight w. The weight window does the following depending on w.

1. If $w_l \leq w \leq w_u$ then the particle is within the weight window and no action is taken.

2. If $w > w_u$ then the particle is above the window and the particle is split by the minimum integer k such that $w_l \leq w/k \leq w_u$, so that the split particles are within the weight window

3. If $w < w_l$ then the particle is below the window and the particle is rouletted with survival probability w/w_s, so that if the particle survives, it will have weight w_s within the window.

A typical problem confronting an MCNP user is the selection of good weight window lower bounds. (The upper bounds are usually defaulted to 5 times the lower bounds.) Fortunately, the weight window method is not extremely sensitive to the lower bounds and the weight window is usually nearly optimal over a broad range. A good weight window can often be obtained by setting the lower weight bound to be inversely proportional to the importance function. The importance function may either be estimated by Monte Carlo (e.g. the weight window generator) or provided by a deterministic code (e.g. see [24]).

In fact, using a weight window that is inversely proportional to the importance function has become so common that many (hopefully not most) users believe that such a weight window is an optimal weight window. In general, it is not optimal. On the other hand, because the weight window is nearly optimal over a broad range, it is true that such a weight window is often nearly optimal.

In fact, some people believe that if a perfect importance function were known, then such a weight window would have to be optimal.

Part of the confusion seems to be connected with the fact that it is theoretically possible to do a zero variance Monte Carlo calculation using an exact importance function. If the importance function is almost right, then a low variance calculation results. As the importance function becomes better and better the variance becomes smaller and smaller. The trouble is that the weight window is not a zero variance technique and so:

1. Even with a pointwise weight window derived from a perfect pointwise importance function, one cannot get a zero variance solution.

2. Because a zero variance sampling scheme, using a nearly perfect importance function, will produce a nearly optimal Monte Carlo solution, does not mean that a weight window sampling scheme, using a nearly perfect importance, function will produce nearly optimal Monte Carlo solution.

Using importance information to reduce the variance is interesting from a philosophical viewpoint. The importance function is the *first* moment (mean or $M_1(P)$) of the score distribution and yet one is trying to reduce the *second* moment ($M_2(P)$), and thereby reduce the history variance $\sigma^2 = M_2 - M_1^2$. Three questions that come to mind are

1. Why should one expect that nonanalog techniques that use $M_1(P)$ information would be useful for minimizing $M_2(P)$?

2. When does using $M_1(P)$ to minimize $M_2(P)$ work?

3. When does using $M_1(P)$ to minimize $M_2(P)$ not work?

A possible reason that one might expect that nonanalog techniques that use $M_1(P)$ information would be useful for minimizing $M_2(P)$ is that for zero variance schemes, using the importance function ($M_1(P)$) *does* minimize $M_2(P)$. For nonzero variance techniques, such as the weight window, this procedure of using importance information to achieve nearly optimal weight windows is often problematical. Sometimes the procedure works well and sometimes it does not.

From a particle perspective, variance reduction efforts should focus on the high scoring particles. But, using importance information to bias the random walks is a focus on the average particle behavior. An analysis of why this procedure works well for some problems and not for others starts with an analysis of the zero variance scheme. In a zero variance scheme, the set of particles that contributes most of the second moment is identical to the set of particles that contributes most to the first moment. Stated mathematically, zero variance implies that $M_2(P) = M_1^2(P)$. This means that first moment (importance) information can essentially be used as a standin for second moment information.

For low variance situations, $M_2(P)$ still looks fairly similar to $M_1^2(P)$ and importance information still can be used roughly as a standin for second moment information. From a particle perspective, if the set of particles contributing most to M_2 is similar to the set of particles contributing most to M_1, then a low variance calculation results. In such cases, one can expect that a weight window inversely proportional to the importance function will be somewhere in the window's broad nearly optimum region. Particle penetration of a simple concrete slab is a typical example where an importance based weight window works very well. The random walks of the high scoring particles are not too different from the random walks of typical particles.

On the other hand, there are many cases where there is *no* set of weight window parameters for which the random walks of the high scoring particles are somewhat similar to the random walks of typical particles. In some cases, even if a user had an optimal weight window (perhaps obtained empirically by testing a huge number of possible weight windows), the particles that contribute most to M_2 may contribute little to M_1. For example, 90% of the variance may be due to particles that contribute only 0.01% of the mean. In these cases, trying to minimize variance by considering average particle scoring (importance) information cannot succeed because the high scoring particles are extremely different from the average scoring particles. Stated mathematically, these cases have $M_2(P) >> M_1^2(P)$ and so M_2 looks nothing like M_1^2 and thus importance information cannot be used as as a standin for $M_2(P)$ information.

Particle penetration of a concrete slab with a small diameter void duct is a typical example of a problem for which for all possible sets of weight windows the variance is dominated by the extremely rare particles that stream directly through the duct and score. Most of the mean will come from particles that slog their way through the concrete but most of the variance will come from particles that stream through the duct. The essential variance problem is that a particle that streams up the duct goes from a very low importance region to a very high importance region in one step. If the window is based on importance information, the particle's weight window region might change by 10^6 from the bottom of the duct to the top of the duct. Splitting by $10^6 : 1$ will be ineffective because splitting works by collecting information on different possible random walks. Once the particle is near the tally surface, there are not really 10^6 different things for a particle to do.

In fact, the essential problem is that the sampling of the special direction up the duct has introduced a huge amount of variance in the sampling process. One can reduce the variance introduced *after* the direction sampling to zero by splitting $\infty : 1$, but any variance introduced *before* the split is not affected by the splitting. To make an importance based weight window effective in this situation, another technique that biases the direction sampling up the pipe has to be used. Because of the direction biasing, the particle's weight will be greatly reduced and then be more consistent with the weight window region it has entered and huge splits will not occur.

As a practical matter, a weight window inversely proportional to the importance seems to work well when the particle weight is always roughly consistent with the weight window and no large splits are required. In this case, there are enough different things for the split particles to do to make splitting effective. The required splits can be small either because the problem is such that no transition from one region to another has a large weight window change (e.g. the simple slab example) or because the transition between regions with a large weight window change (e.g. the duct example) is biased so that the bias modified weight is roughly consistent with the weight window. Mathematically, large splits are an indication that $M_2(P) >> M_1^2(P)$ and that the second moment information looks nothing like the first moment information.

There are potential traps for unwary weight window users who blindly take the importance function output from an S_N code and produce a weight window from this importance function. Although this will work well in some cases, it will be problematic in other cases for the reasons indicated above. It is particulary worrisome to have source weights near unity and weight windows of 10^{20}. As explained in the previous section, roulette saves time but increases the variance. What is the extra time saved by using windows going all the way up to 10^{20} instead of say, allowing no weight windows larger than 100? Using a window of 10^{20} is an instance of focusing too much on the mean and too little on the variance aspects of the problem. What is truely worrisome is that such harsh roulette games may *totally* prevent some random walks from occuring and one will never see an instance of a huge weight (say W) particle scoring. When this happens the mean may be low because it is totally missing any representative sample of the some random walks. That is, the mean may be low because it is missing large W contributions. Worse still, the true error may be substantially larger than the estimated error because the estimated M_2 may be low because it is missing the large W^2 contributions. The variance of the variance may be wildly low because it is missing W^4 contributions. Thus the calculation may pass statistical tests even though the calculation is not converged.

In short, caveat emptor using weight windows spanning 20 orders of magnitude just because the S_N importance function spans 20 orders of magnitude. The S_N is supplying information about the first moment and this is not always a good way to minimize the second moment. The time saved is usually small and not commensurate with the dangers inherent in using such weight windows.

Though not discussed here, it is worth noting that there are methods that determine M_1, M_2 and computer time T as a function of some variance reduction parameters, V_i, and then choose the optimal V_i that minimize $(M_2 - M_1^2)T$. A nonexhaustive list of references include [25–28]. (Burn [25], in particular, has many related publications.)

11 Summary

Monte Carlo methods can be extremely useful, diverse, clever, and intricate. Viewing Monte Carlo primarily as a way to solve an equation that describes some average over the transport process is inappropiate for two reasons. First, there is a richness of information that is available in a Monte Carlo calculation but the averaging process removes much of this richness. Second, there is a tendency to think about Monte Carlo methods in terms of what one sees in the equations. Thus much of the discussion about transport Monte Carlo is in terms of simulating terms in an equation rather than simulating the transport process. This tends to result in a very restricted view about Monte Carlo methods and theoretical claims whose assumptions are not properly identified.

I once talked at great length, confusion, and frustration with somebody who did not understand the richness in the variety of ways one could apply Monte Carlo methods. The confusion finally went away when he exclaimed in astonishment "you mean that you simulate a naturally occuring Markov process with a nonMarkov process!" He was looking at the equations and the transition kernels in the equations were Markovian objects like $K(P' \to P)$, so his mind was fixed on using biased Monte carlo transition probabilities of the form $\tilde{K}(P' \to P)$ that were also Markovian. There is, of course, no reason to limit the Monte Carlo playground to the tiny space of Markovian random walks, but that seems to be many people's natural inclination.

In addition to the tendency to make unstated assumptions about what a Monte Carlo calculation must look like, these assumptions slow the progress of Monte Carlo methods development by unnecessarily limiting the Monte Carlo playground. This paper discussed some of the more common misconceptions in practical Monte Carlo calculations in hopes of correcting them. Additionally, this paper discussed some more theoretical misconceptions in hopes of broadening peoples' view of Monte Carlo.

Acknowledgement Thanks to Avneet Sood and ***** for reading and commenting on a draft of this paper.

References

[1] X-5 Monte Carlo Team, "MCNP-A General Monte Carlo N-Particle Transport Code, Version 5," Los Alamos National Laboratory Report LA-UR-03-1987, April 24, 2003

[2] FUNDAMENTALS OF MONTE CARLO PARTICLE TRANSPORT LA-UR-05-4983 by Forrest Brown

[3] T.E. Booth, "Genesis of the Weight Window and the Weight Window Generator in MCNP - A Personal History", LA-UR-06-5807 (2006).

[4] G. GREENMAN, R. PROCASSINI, and C. CLOUSE, A Monte Carlo Method for Calculating Initiation Probability, Proc. Joint Int. Topl. Mtg. Mathematics and Computations and Supercomputing in Nuclear Applications (M& C + SNA 2007), Monterey, California, April 1519, 2007, American Nuclear Society 2007!.

[5] Neutron Branching and the Probability of Initiation for Weak Sourced, Super Critical Systems Michael Rising and Forrest Brown LA-UR-09-05299 (2009)

[6] Michael E. Rising, Forrest B. Brown, and Anil K. Prinja, “The Probability of Initiation in MCNP”, *Transactions of the American Nuclear Society*, vol. 102, pp. 258-260 (2010)

[7] T.E. Booth, Comments on Monte Carlo Probability of Initiation Estimates for Neutron Fission Chains NUCLEAR SCIENCE AND ENGINEERING: 166, 175178 (2010)

[8] T.E. Booth, ”Monte Carlo probability of initiation extimates in MCNP”, Los Alamos Report LA-UR 09-05873 (2009)

[9] Monte Carlo Variance Reduction Approaches for Non-Boltzmann Tallies Thomas E. Booth Los Alamos National Laboratory Report LA-12433 (Dec. 1992)

[10] A Monte Carlo Variance Reduction Approach for Non-Boltzmann Tallies Thomas E. Booth Nuclear Science and Engineering: 116, 113-124 (1994)

[11] Pulse Height Tally Variance Reduction in MCNP Thomas E. Booth Los Alamos Report LA-13955 Jun 2002

[12] E Shuttleworth, ”The Pulse Height Distribution Tally in MCBEND”, Proc International Conference on Radiation Shielding ICRS’9, Tsukuba, Japan. (October 1999)

[13] PHYSOR 2004 -The Physics of Fuel Cycles and Advanced Nuclear Systems: Global Developments Chicago, Illinois, April 25-29, 2004, on CD-ROM, American Nuclear Society, Lagrange Park, IL. (2004) Variance Reduction Techniques for the Monte Carlo Simulation of Neutron Noise Measurements Máté Szieberth and Jan Leen Kloosterman

[14] Máté Szieberth and Jan Leen Kloosterman, “Estimation of coincidence and correlation in non-analogous Monte Carlo Particle Transport,” PHYSOR 2010 ? Advances in Reactor Physics to Power the Nuclear Renaissance Pittsburgh, Pennsylvania, USA, May 9-14, 2010, on CD-ROM, American Nuclear Society, LaGrange Park, IL (2010)

[15] J. De Beenhouwer, S. Staelens, M. Dressel, et al, “Geometrical importance sampling and pulse height tallies in GATE,” Conf Proc IEEE Eng Med Biol Soc, Vol. 2 (2004), pp. 1349-1352.

[16] L. E. Smith et. al. “Coupling Deterministic and Monte Carlo Transport Methods for the Simulation of Gamma-Ray Spectroscopy Scenarios,” IEEE Transactions onNuclear Science, Oct. 2008, Volume: 55 Issue: 5, p 2598 - 2606

[17] Jacob Benz and Todd S. Palmer, “Pulse height distributions from deterministic radiation transport simulations,” Annals of Nuclear Energy, Volume 37, Issue 11, November 2010, Pages 1486-1493

[18] Iván Lux and László Koblinger, "*Monte Carlo Particle Transport Methods: Neutron and Photon Calculations*," CRC Press, Inc. (1991).

[19] Thomas E. Booth "A Transport Process Approach to Understanding Monte Carlo Transport Methods" Los Alamos Report LA-UR-04-1426.pdf, June 3, 2004

[20] Simultaneous Monte Carlo Zero-Variance Estimates of Several Correlated Means, Thomas E. Booth, Nuclear Science and Engineering: vol 129, June 1998

[21] J. Eduard Hoogenboom, "Zero-Variance Monte Carlo Schemes Revisited," Nuclear Science and Engineering,160, 122 (2008)

[22] H. KAHN, Applications of Monte Carlo, RM-1237- AEC, Rand Corporation (1956).

[23] Zero-Variance Solutions for Linear Monte Carlo, Thomas E. Booth, Nuclear Science and Engineering: 102, 332-340 (1989)

[24] Alireza Haghighat and John Wagner, "Monte Carlo Variance Reduction with Deterministic Importance Functions," *Progress in Nuclear Energy*, Vol. 42, No. 1, pp. 25-53 (2003)

[25] K.W. Burn, "Complete optimization of space/energy cell importances with the dsa cell importance model," *Ann. Nucl. Energy*, 19:65-98, 1992.

[26] A.J. van Wijk and J.E. Hoogenboom, "A priori efficiency calculations for monte carlo applications in neutron transport", *Joint International Conference on Supercomputing in Nuclear Applications and Monte Carlo*, Tokyo, Japan, Oct. 2010.

[27] C.J. Solomon, A. Sood, T.E. Booth, and J.K. Shultis. "An Sn Approach to Predicting Monte Carlo Cost with Weight Dependent Variance Reduction," *Trans. Amer. Nuc. Soc. 2010 ANS Winter Meeting. 2010*

[28] C.J. Solomon, A. Sood, T.E. Booth, and J.K. Shultis, "TWO-DIMENSIONAL MONTE CARLO COST MINIMIZATION OF WEIGHT-WINDOW VARIANCE REDUCTION PARAMETERS WITH DETERMINISTIC CALCULATIONS," International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2011) Rio de Janeiro, RJ, Brazil, May 8-12, 2011.

Figure 1 Typical Estimation Process

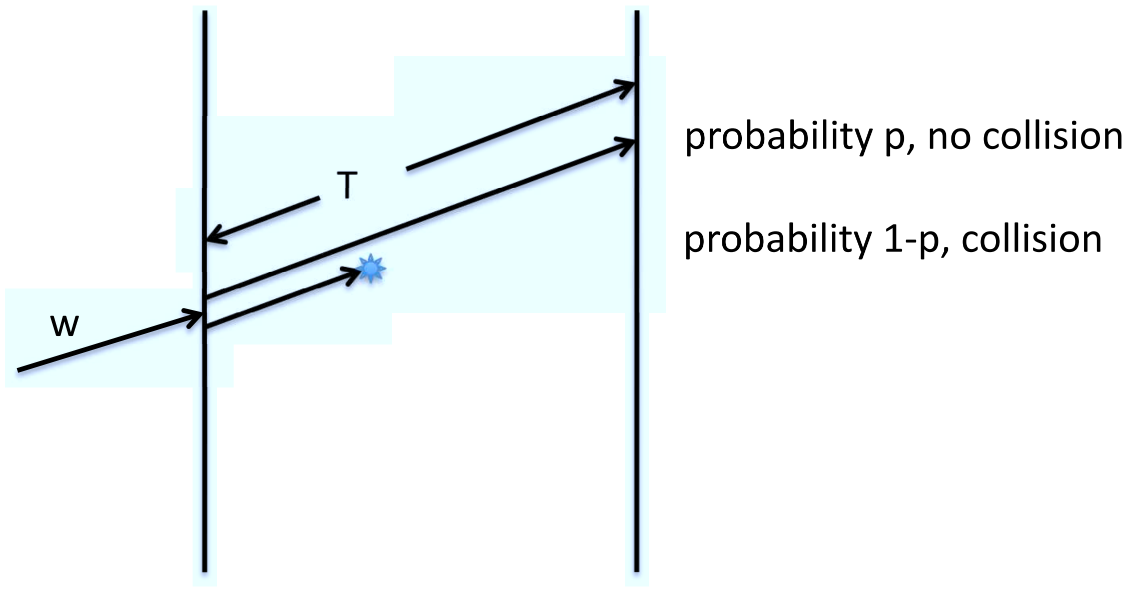

Figure 1: Typical estimation process

Figure 2 Another Estimation Process

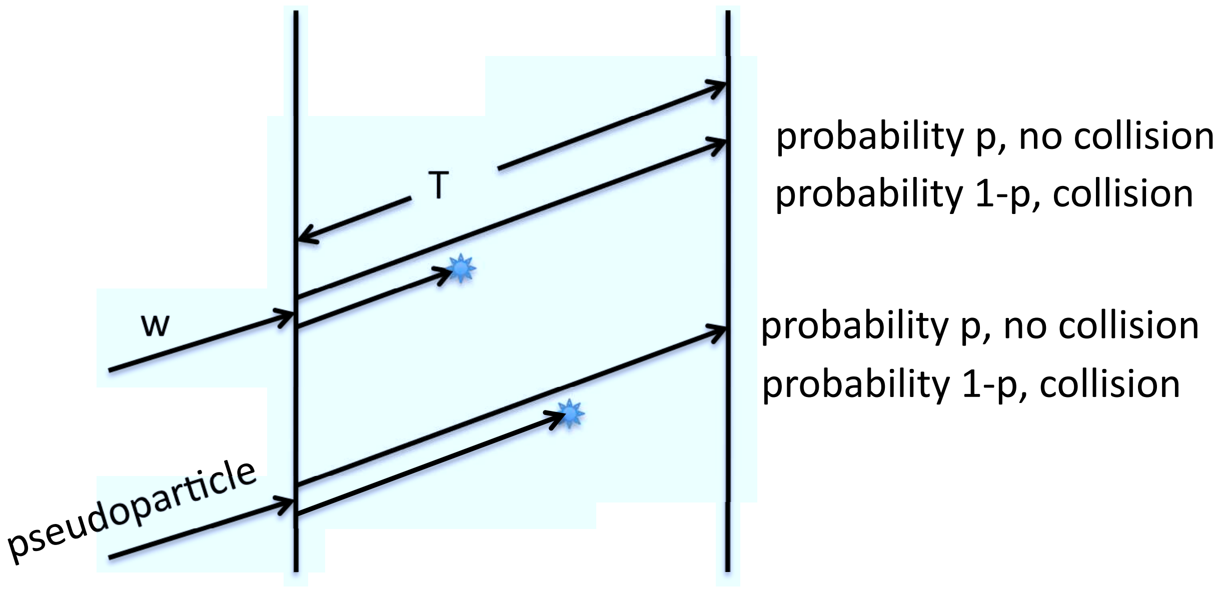
T
probability p, no collision
probability 1-p, collision
w
probability p, no collision
probability 1-p, collision
pseudoparticle

Figure 2: Another estimation process