LA-UR 90-4386

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

TITLE UNBIASED COMBINATIONS OF NONANALOG MONTE CARLO TECHNIQUES AND FAIR GAMES (U)

AUTHOR(S) Thomas E. Booth
Shane P. Pederson

SUBMITTED TO Nuclear Science and Engineering

By acceptance of this article, the publisher recognizes that the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes.

The Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy

Los Alamos National Laboratory
Los Alamos, New Mexico 87545

FORM NO. 836 R4
ST. NO. 2629 5/81

UNBIASED COMBINATIONS OF NONANALOG MONTE CARLO TECHNIQUES AND FAIR GAMES (U)

by

Thomas E. Booth, X–6, and Shane P. Pederson, A–1
Los Alamos National Laboratory
P. O. Box 1663
X–6, MS B226; A–1, MS F600
Los Alamos, NM 87545

9 1 8 1 1 4 0 0

Pages – 17
Figures – 1

ABSTRACT

Historically, Monte Carlo variance reduction techniques have developed one at a time in response to calculational needs. This paper provides the theoretical basis for obtaining unbiased Monte Carlo estimates from all possible *combinations* of variance reduction techniques. Hitherto, the techniques have not been proven to be unbiased in arbitrary combinations. The authors are unaware of any Monte Carlo techniques (in any linear process) that are not treated by the theorem herein.

9 1 8 1 1 4 0 1

I. INTRODUCTION

The Los Alamos computer code MCNP[1] allows the user to simulate transport events according to the natural event probabilities. When an event is sampled according to its naturally occurring probability, the sampling is called an *analog event sampling*. If every event on a random walk is sampled according to the naturally occurring probabilities, the random walk is called an *analog random walk*. If every random walk is analog, then the estimates produced by these walks are *analog estimates*.

Analog sampling is often computationally impractical when estimates associated with rare events are desired. For this reason, MCNP (like most general-purpose Monte Carlo transport codes) allows nonanalog samplings for which there are no corresponding natural processes. Nonanalog Monte Carlo techniques are essential to many calculations and historically they have been developed one at a time as needed. Each new nonanalog technique has, at best, been proven to preserve the expected tallies (i.e., be unbiased) when used by itself. The techniques have not been proven to be unbiased in arbitrary combinations.

This paper provides the theoretical foundation for using any arbitrary combinations of nonanalog Monte Carlo techniques currently in MCNP. Furthermore, the proofs contained herein are designed to cover all current nonanalog techniques in any Monte Carlo transport code as well as future techniques. The authors are unaware of any techniques, in any linear Monte Carlo process, that are not covered by the general theorems herein.

II. PHYSICAL INTERPRETATION OF MONTE CARLO PARTICLE TRANSPORT

An analog Monte Carlo simulation can be interpreted as a direct imitation of nature. A computer model of the physical transport process is produced by specifying the stochastic physical laws governing particle behavior. The computer then applies these laws to a computer particle on a direct one-for-one basis. That is, each computer particle is made to represent one physical particle. To the extent that computer particles obey the same stochastic laws as physical particles, the behavior of the computer particles will be statistically identical to the behavior of the physical particles. Thus one can infer the statistical properties (e.g., average number of boundary crossings) in a physical system by observing the statistical properties in the computer model.

The purpose of a nonanalog simulation is to provide the same mean estimates as the analog simulation, but with less variance. In a nonanalog Monte Carlo simulation, each particle is assigned a statistical weight ("weight") that multiplies the tallies (contributions to the estimates) made by the particle. For example, when a particle of weight 2 crosses a surface, the particle's tally will be the same as if 2 particles had crossed the surface. The intuitive notion is that the weight is the number of physical particles represented by the computer particle. Common nonanalog techniques can nonphysically redistribute weight (e.g., splitting), destroy weight (e.g., lose Russian roulette), or create weight (e.g., win Russian roulette). To convince people that the nonanalog techniques will produce the same expected tallies as analog Monte Carlo, it is usually demonstrated that, on average, the number of particles (i.e., weight) is preserved.

For example, replacing a particle of weight w at phase-space point $P = (r, v, t)$ with two particles at P, each of weight $w/2$ (i.e,. splitting), still represents w physical particles at P (note that the expected weight is trivially preserved). Similarly, a particle of weight w at P can be replaced , with probability p by a particle of weight w/p at P, or with

probability $(1-p)$ the particle is terminated (Russian roulette). The expected weight at P afterwards is $p(w/p) + (1-p)0 = w$, the same as without Russian roulette. Thus the expected number of particles at P has been preserved.

If the expected number of particles (weight) is preserved with every nonanalog sample, it seems plausible that all (nonanalog) tallies have the same expected values as the analog calculation. Weights are, in fact, analogous to a "fair game" or martingale (Doob,[2] p. 299). In a martingale, the expected value of the next element in a sequence is simply the current value of the sequence. Likewise, for particle transport, it will be shown that the expected weight following the next nonanalog game is that of the corresponding analog process, so that the difference between the expected weight following the next nonanalog game, and the corresponding analog weight, is the same as the current difference, i.e., 0.

III. REASON FOR THE DEMON

Before proceeding to the proof, it is worth pointing out that special combinations of variance reduction techniques have been proven to preserve the expected tallies. For instance, Spanier and Gelbard[3] (p. 113) show that for a subcritical medium each step of the random walk can be sampled from a nonphysical probability. To preserve the expected weight at each step the particle's weight is multiplied by the ratio of the true physical probability to the nonphysical probability actually sampled. In fact, they explicitly state "the particles may be thought of as carrying weights in order to adjust the expected weight undergoing collision to the analog value." Their proof requires explicit knowledge of the particular variance reduction techniques being used. Their proof will need to be modified if, for instance, a weight cutoff game is added.

Unlike the aforementioned proof, the proof herein does not take a fixed collection of nonanalog techniques and prove that, used together, they preserve the expected tallies. Instead, the proof herein considers all possible nonanalog techniques that preserve the expected weight executing each physical sequence of events. It is not sufficient simply to

show that the particle density is preserved, because some tallies depend on more than just the present particle density. For instance, one may be interested in tallying the number of particles entering region B that had previously crossed through region A. (This is called cell flagging in MCNP.) One needs to show that the expected weight undergoing any possible event string is preserved, just in case a user decides to ask for a tally on that event string. Because tallies are made on event strings, event strings assume central importance in the proof.

The proof herein is a bit abstract because the authors wanted a proof that would apply to all possible nonanalog techniques and not just the techniques currently in practice. It may help the reader to think about the problem the way that the authors did before the authors encountered some technical difficulties. Initially there was no demon. One simply imagined a Monte Carlo code with many possible nonanalog techniques available. Imagine that a user selects some set, V, of nonanalog techniques. Will using V preserve the expected tallies (produced by analog sampling)? To answer this question, a sequence of nonanalog calculations was considered whose limit was the original (user specified) nonanalog calculation. The kth nonanalog calculation was defined to be identical to the original nonanalog calculation through the first k random numbers; afterwards, all remaining samplings were analog. With this definition, $k = 0$ implies a completely analog calculation and $k = \infty$ is the original nonanalog calculation. An inductive proof showed that the expected behavior at $k = \infty$ was the same as the analog game.

The difficulty with this approach was that it became exceedingly complex to rigorously define what "analog after the first k random numbers" meant. In particular, the kth random number might be in the middle of a nonanalog calculational step! The authors next tried to define the k game as "identical to the original nonanalog calculation through the first k random numbers with no new nonanalog techniques started." That is, nonanalog techniques already started could complete, but the calculation would "revert to analog as quickly as possible." This too became exceedingly difficult to define precisely. The authors

then realized that what was really important was not that the remainder of the calculation had to be analog, but that the remainder of the calculation had to have a fixed set of nonanalog techniques that had the same expected behavior as the analog calculation. With this realization, the authors redefined the kth nonanalog calculation as indicated below.

To ensure that no clever Monte Carlo practitioner can provide a nonanalog technique that preserves the expected weight but that is not covered by the theorem herein, a demon is created. This demon is free to alter the sampling in any way that preserves the expected weight executing all physical event strings. The demon may replace any analog sampling with any nonanalog sampling. The demon may also replace any nonanalog sampling with a different nonanalog sampling. Furthermore, the demon is free to make these changes through the first k random numbers. After k random numbers the demon may not change the sampling and the sampling is done according to the demon's last specification. This is called the k *game* and the sampling rules specified on the kth random number are called the k *game rules* and remain fixed throughout the remainder of the particle's random walk. This is shown in Fig. 1.

The particle's history starts with analog rules because the demon has made no alterations. Thus the 0 game rules are analog and because all subsequent samplings are done with the same rules, then the entire 0 game is analog. The proof shows by induction that the expected weight executing all event strings in the k game is the same as the analog game (0 game). Taking the limit as $k \rightarrow \infty$ shows that the nonanalog random walk (under specified conditions) preserves the expected weight executing all event strings.

IV. DISCUSSIONS AND DEFINITIONS

The random numbers are central to the proof herein because the random numbers *completely determine* a particle's history. (A particle history is one independent sample followed from the source to termination of the source particle and its progeny.) A few definitions involving the random numbers are required. (For what follows, assume all integrals and sums exist and their values are finite.)

Definition 1: Let $(r_1, r_2, r_3, \cdots, r_k)$ be k random numbers used by a history. (The r_j are independent uniform random numbers on (0,1].)

Definition 2: $\vec{r}_k \triangleq (r_1, r_2, r_3, \cdots, r_k)$

Definition 3: $\vec{r} \triangleq (r_1, r_2, r_3, \cdots)$

Definition 4: $\int (\ )\, d\vec{r}_k \triangleq \int dr_1 \int dr_2 \cdots \int dr_k (\ )$

Definition 5: $\int (\ )\, d\vec{r} \triangleq \int dr_1 \int dr_2 \cdots \int dr_j \cdots (\ ) = \lim_{j \to \infty} \int (\ ) d\vec{r}_k$

Definition 6: An *event string* is a sequence of physical events tracing a track back to the source. (Tallies are made upon event string occurrences.)

Definition 7: $S_d(\vec{r}_k)$ = total weight executing event string d *through* the first k random numbers in the nonanalog game.

Definition 8: $B_d(\vec{r}_k)$ = expected total weight executing event string d *after* the first k random numbers if the history were followed subsequently with no changes in rules.

Notes:

1) For $k = 0$, then $S_d(\ ) = 0$ because no physical event strings have been executed.

2) For $k = 0$, then $B_d(\ \ ) =$ expected total weight executing event string d under analog rules.

V. CONDITIONS FOR PRESERVATION OF EXPECTED WEIGHT

Suppose that just before each random number, (r_{k+1}), is determined, a demon decrees what Monte Carlo techniques will be in use until his next decree (see Fig. 1). The only restrictions placed on the demon's decrees will be that the expected weight executing all event strings must satisfy

$$B_d(\vec{r}_k) = \int \left[S_d(\vec{r}_{k+1}) + B_d(\vec{r}_{k+1})\right] dr_{k+1} - S_d(\vec{r}_k) \quad ,$$

for all event strings d and all k.

Theorem 1: If all samplings have the property (for all k, d)

$$B_d(\vec{r}_k) = \int \left[S_d(\vec{r}_{k+1}) + B_d(\vec{r}_{k+1})\right] dr_{k+1} - S_d(\vec{r}_k) \quad , \tag{1}$$

and if

$$\lim_{k \to \infty} B_d(\vec{r}_k) = 0 \quad , \tag{2}$$

then the expected weight executing all event strings d is preserved.

Proof (by induction)

Suppose that for some k (and *all* d)

$$B_d(\ ) = \int \left[S_d(\vec{r}_k) + B_d(\vec{r}_k)\right] d\vec{r}_k \quad . \tag{3}$$

Substituting Eq. (1) into Eq. (3) yields

$$B_d(\ ) = \int \left[S_d(\vec{r}_k) + \left\{\int (S_d(\vec{r}_{k+1}) + B_d(\vec{r}_{k+1}))\, dr_{k+1} - S_d(\vec{r}_k)\right\}\right] d\vec{r}_k \quad . \tag{4}$$

Noting that $\int d\vec{r}_k \int dr_{k+1} \rightarrow \int d\vec{r}_{k+1}$ yields

$$B_d(\ ) = \int \left[S_d(\vec{r}_{k+1}) + B_d(\vec{r}_{k+1})\right] d\vec{r}_{k+1} \quad . \tag{5}$$

Thus, if Eq. (3) is true for some k and all d, then by induction, it is true for all $k' \geq k$ and all d. Note that Eq. (3) is true for $k = 0$ (and all d), that is

$$B_d(\ ) = S_d(\ ) + B_d(\ ) = 0 + B_d(\ ) \quad . \tag{6}$$

Thus, Eq. (3) is true for all $k \geq 0$ and all d. Taking the limit of Eq. (3) as $k \rightarrow \infty$ (noting Eq. (2)) yields

$$B_d(\ ) = \int S_d(\vec{r})\, d\vec{r} \quad , \tag{7}$$

as desired. ($B_d(\ )$ is the expected weight undergoing event string d in the analog game and the RHS is the expected weight undergoing event string d in the nonanalog game.)

VI. CONDITIONS FOR TALLY PRESERVATION

We have shown that the expected weight executing each event string d is preserved. From this it can be demonstrated that the tallies are also preserved provided that they are

proportional, in expectation, to the track weight. To do this, some additional notation is introduced:

Definition 9: $S_d^a(\vec{r})$ = total weight executing event string d in the analog game using $\vec{r}$ (corresponding to $S_d(\vec{r})$ for nonanalog games).

Definition 10: D = the set of all possible event strings d.

Definition 11: $D(\vec{r})$ = the set of all event strings produced by the nonanalog transport process with random numbers $\vec{r}$.

Definition 12: $D^a(\vec{r})$ = the set of all event strings produced by the analog transport process with random numbers $\vec{r}$.

Definition 13: $I_d(\vec{r}) = \begin{cases} 1 & \text{if event string d is produced by the nonanalog transport process generated by } \vec{r} \\ 0 & \text{otherwise.} \end{cases}$

Definition 14: $I_d^a(\vec{r}) = \begin{cases} 1 & \text{if event string d is produced by the analog process with random numbers } \vec{r} \\ 0 & \text{otherwise.} \end{cases}$

I_d and I_d^a are membership indicator functions for the sets $D(\vec{r})$ and $D^a(\vec{r})$, respectively; e.g., $I_d(\vec{r}) = 1$ iff $d \epsilon D(\vec{r})$. Because $B_d(\ ) = \int S_d^a(\vec{r})d\vec{r}$ by definition, it follows from Eq. (7) that

$$\int S_d(\vec{r})d\vec{r} = \int S_d^a(\vec{r})d\vec{r} \qquad \text{for all} \qquad d \epsilon D(\vec{r}) \cap D^a(\vec{r}) \quad , \tag{8}$$

and that

$$\int S_d(\vec{r})I_d(\vec{r})d\vec{r} = \int S_d^a(\vec{r})I_d^a(\vec{r})d\vec{r} \qquad \text{for all} \qquad d \epsilon D \quad . \tag{9}$$

There are two types of tallies, those that do not require additional random numbers, (e.g., track-length tallies), and those that may, (e.g., point detector tallies in MCNP). For the first type, let

Definition 15: $T_d(w)$ = the tally made by total weight w executing event string d.

The tally associated with the nonanalog transport process is

$$\tilde{T}(\vec{r}) = \sum_{d \epsilon D(\vec{r})} T_d(S_d(\vec{r})) \quad , \tag{10}$$

and the corresponding tally associated with the analog process is

$$\tilde{T}^a(\vec{r}) = \sum_{d \epsilon D^a(\vec{r})} T_d(S_d^a(\vec{r})) \quad . \tag{11}$$

Monte Carlo tallies are proportional to their weight, i.e.,

$$T_d(w) = wT_d(1) \quad . \tag{12}$$

With E( ) denoting expectation, the expected value of the tally in the nonanalog process is

$$E(\tilde{T}(\vec{r})) = \int \sum_{d \epsilon D(r)} T_d(S_d(\vec{r}))\, d\vec{r} \quad . \tag{13}$$

By Eq. (12) and Defn. 13, this becomes

$$= \int \sum_{d \epsilon D} S_d(\vec{r}) L_d(\vec{r}) T_d(1) d\vec{r} \tag{14}$$

$$= \sum_{d \epsilon D} T_d(1) \int S_d(\vec{r}) L_d(\vec{r}) d\vec{r} \quad , \tag{15}$$

where integration and summation can be interchanged because the integrand and summand are independent. Likewise for the analog process,

$$E\left(\tilde{T}^a(\vec{r})\right) = \sum_{d \epsilon D} T_d(1) \int S_d^a(\vec{r}) I_d^a(\vec{r}) d\vec{r} \quad , \tag{16}$$

and hence by Eq. (9),

$$E\left(\tilde{T}(\vec{r})\right) = E\left(\tilde{T}^a(\vec{r})\right) \quad . \tag{17}$$

A similar result can be shown for stochastic tallies. These tallies require additional random numbers $\vec{q} = (q_1, q_2, \ldots)$. For this case, let

Definition 16: $T_d(w, \vec{q})$ = the tally made by weight w executing event string d when random numbers $\vec{q}$ are used in the tally process.

$\tilde{T}(\vec{r}, \vec{q})$ and $\tilde{T}^a(\vec{r}, \vec{q})$ have corresponding definitions. Corresponding to Eq. (12), for stochastic tallies the following holds:

$$\int T_d(w, \vec{q}) d\vec{q} = w \int T_d(1, \vec{q}) d\vec{q} \quad . \tag{18}$$

The proof that $E(\tilde{T}(\vec{r}, \vec{q})) = E(\tilde{T}^a(\vec{r}, \vec{q}))$ is much the same as before:

$$E\left(\tilde{T}(\vec{r}, \vec{q})\right) = \int \int \sum_{d \epsilon D(r)} T_d\left(S_d(\vec{r}), \vec{q}\right) d\vec{q} d\vec{r} \quad . \tag{19}$$

$$= \int \sum_{d \epsilon D(r)} \left[\int T_d\left(S_d(\vec{r}), \vec{q}\right) d\vec{q}\right] \tag{20}$$

$$= \int \sum_{d \epsilon D(r)} S_d(\vec{r}) \int T_d(1, \vec{q}) d\vec{q} d\vec{r} \tag{21}$$

$$= \int \sum_{d \epsilon D} I_d(r) S_d(\vec{r}) \int T_d(1, \vec{q}) d\vec{q} d\vec{r} \tag{22}$$

$$= \sum_{d \epsilon D} \left[\int T_d(1, \vec{q}) d\vec{q}\right] \int I_d(\vec{r}) S_d(\vec{r}) d\vec{r} \tag{23}$$

These steps follow from the independence of $\vec{q}$ and d, and by the use of Eq. (18) and Defn. 16. Similarly,

$$E\left(\tilde{T}^a(\vec{r},\vec{q})\right) = \sum_{d\epsilon D}\left[\int T_d(1,\vec{q})d\vec{q}\right]\int I_d^a(\vec{r})S_d^a(\vec{r})d\vec{r} \quad , \tag{24}$$

and hence by Eq.(9), the expected tally is preserved:

$$E\left(\tilde{T}(\vec{r},\vec{q})\right) = E\left(\tilde{T}^a(\vec{r},\vec{q})\right) \quad . \tag{25}$$

VII. COROLLARY

If each nonanalog technique begins by throwing a random number (the $(k+1)^{st}$) that is not used, the event strings are unaffected and hence for this nonanalog technique

$$S_d(\vec{r}_{k+1}) = S_d(\vec{r}_k) \quad . \tag{26}$$

Furthermore, substituting this into Eq. (1) yields (for this nonanalog technique)

$$B_d(\vec{r}_{k+1}) = B_d(\vec{r}_k) \quad , \tag{27}$$

as the condition for preserving the expected weight executing all event strings. From now on, the nonanalog techniques will throw away one random number so that Eq. (27) applies.

Application to Integer Splitting

Definition - Splitting j-for-1 on an independently sampled track means:

1. Replacing 1 parent track (of weight w) by j tracks that are identical to the parent track except that their weights are w/j.
2. The split particles will be followed exactly the same as the parent particle would have been followed. That is, if (in the absence of the split) the parent track produces a

total weight $W_d(\vec{r})$ executing event string d when the track's subsequent random walk is sampled using $\vec{r}$, then the ith split track will produce

$$W_{id}(\vec{r}) = \frac{W_d(\vec{r})}{j} \qquad i = 1, \cdots, j \tag{28}$$

total weight executing event string d when the ith track's subsequent random walk is sampled using $\vec{r}$.

Because the random walk for parent particle track G is being sampled independently of all other particle tracks, all other tracks will be unaffected by a split of track G. Thus their distributions of random walks will be identical, independent of whether track G is split. Thus if the j split tracks have the same expected weight executing all event strings as track G, then the expected weight executing all event strings in the entire random walk will be preserved as required.

The expected total weight executing event string d in the split case is

$$\begin{aligned} E\left[\sum_{i=1}^{j} W_{id}(\vec{r}_i)\right] &= \sum_{i=1}^{j} E\left[W_{id}(\vec{r}_i)\right] = \sum_{i=1}^{j} \int W_{id}(\vec{r}_i) d\vec{r}_i \\ &= \sum_{i=1}^{j} \int \frac{W_d(\vec{r}_i)}{j} d\vec{r}_i = \int W_d(\vec{r}) d\vec{r} \quad . \end{aligned} \tag{29}$$

Note that the last term is the expected weight executing event string d for the parent track so that the j split tracks have the same expected weight executing all event strings as track G. This completes the proof of unbiasedness by Eq. (27) of the corollary.

Note that although the splitting game has required that the split tracks have the same distribution of random walks as the parent track has, the distribution of random walks of the split tracks can be changed in subsequent nonanalog games. Thus there is no loss of generality in defining splitting in the above manner.

VIII. CONCLUSION

This paper provides theoretical justification for arbitrary combinations of nonanalog Monte Carlo techniques.

REFERENCES

1. J. F. Briesmeister, editor, "MCNP – A General Monte Carlo Code for Neutron and Photon Transport Version 3A," Los Alamos National Laboratory manual LA–7396–M, Rev. 2 (September 1986).

2. J. L. Doob, "*Stochastic Processes*," John Wiley and Sons (1953).

3. Jerome Spanier and Ely M. Gelbard, "*Monte Carlo Principles and Neutron Transport Problems*," Addison-Wesley Publishing Company (1969).

FIGURE CAPTIONS

Figure 1: Specifications for Demon's Rule Changes.

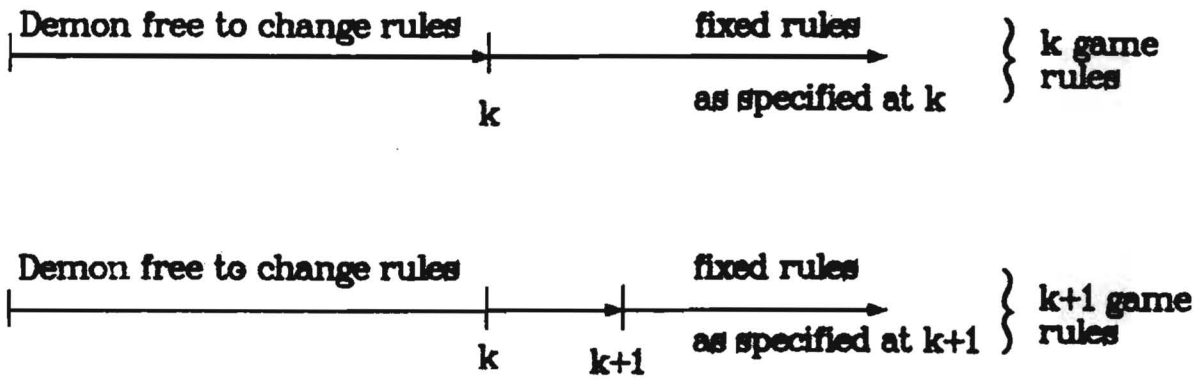

Figure 1. Specifications for Demon's Rule Changes.

9 1 8 1 1 4 1 6