Genesis of the Weight Window and the Weight Window Generator in MCNP - A Personal History

Thomas E. Booth

July 28, 2006

Los Alamos Report LA-UR-06-5807, Los Alamos National Laboratory, Mail Stop A143, Los Alamos, New Mexico, 87545 USA   email teb@lanl.gov

Abstract

The weight window and weight window generator have proven to be effective tools for variance reduction in Monte Carlo particle transport calculations. From time to time, people have contacted the MCNP team seeking historical reference information about the weight window and weight window generator. This report supplies the initial history of the weight window and the weight window generator. Additionally, this report gives the reasons for the window parameters that the author initially recommended (and are still recommended today) as well as a brief list of the improvements others at Los Alamos have made over time.

1 Introduction

The weight window and weight window generator have been in MCNP[1] since the early 1980's. The first reference to the weight window (outside of Los Alamos internal memorandums) occurs in 1980 [2, 3]. The weight window generator is not mentioned in [2, 3], but substantial numbers of users were testing it. The first reference to the weight window generator (outside of Los Alamos internal memorandums) occurs in 1982[4], but the methods were in heavy use in Los Alamos by 1980 in various developmental "patched" versions of MCNP. (Note that reference [3, page 280] indicates that MORSE [5] also seems to have had some type of weight bounding scheme, though apparently its intended use was not for narrow importance-based weight bounding as in MCNP.)

From time to time a number of questions have arisen about the genesis of the weight window and weight window generator, particularly among authors needing to cite references to properly credit the ideas for both. The reference situation can be confusing. The first difficulty is that there is no way one would know who to give credit for the weight window idea. The weight window simply appears in publications about the weight window generator. The second difficulty is that the full paper[6] describing the weight window generator has two authors and it is not specified in the paper who is responsible for which ideas in the paper. The short answer is that Booth is responsible for the weight window and weight window generator methods described in the paper. Hendricks is responsible for the "on the fly generator" method described in the paper. Both authors share responsibility for the specific test problems illustrating the methods.

Many years have passed since the initial idea and early implementation of the weight window and generator. During these years the capability has been significantly enhanced and extended by many people. These efforts are described in Section 4. Also we have been asked over the years why certain parameters are recommended or defaulted. An explanation is given in Section 3. My personal recollection of the early history is given in Section 2

2 Initial History of the Weight Window and Generator

In 1978, I was still a graduate student with extremely limited Monte Carlo experience when Buck Thompson (then group leader of the Monte Carlo transport group) asked me to study the exponential transform in MCNP. In particular, why did MCNP's exponential transform sometimes give wildly unreliably results in shielding calculations even though the sample variance estimates looked good? In fact, Buck indicated that the exponential transform had the reputation as a "dial an answer technique" because the mean estimate seemed to depend on the user's choice of exponential transform parameter.

I looked at how the random walks behaved on an event by event basis as a function of the exponential transform parameter. (The "event log" option in MCNP allowed me to examine easily the outcome of every sampling along a particle's random walk.) I found that the higher the transform parameter, the larger the possible spread of weights was in a given phase-space region of the problem. In particular, large weight particles sometimes occurred near the tally region. The particles typically accumulated the large weight because they had many collisions, and at each collision there was a weight multiplication. A track could then occasionally pick up very large weight. (I gave an analysis in the MCNP[1, Chapter 2 section 7] manual of the exponential transform behavior in the absence of a weight window.)

I reasoned that in any given region of the phase-space it made no computational sense to be following particles of widely differing weights, so the weights should be bounded within a narrow range. If the weight was too small, a roulette game was played such that if the particle survived the roulette, its weight would be in the correct range. If the weight was too large then the particle would be split by the minimum integer that put the post-split weight into the correct range. I termed this range the "weight window."

This left open the question of what the correct range was in any given region. I noted that an ideal situation would be to have every track in the problem contribute approximately the same amount to the tally, although this was impractical in MCNP. On the other hand, it was possible to ensure that the *expected* contribution of any track was approximately the same by choosing the track's weight at phase-space location P to be inversely proportional to the expected tally produced by a unit weight track at P (i.e. the importance).

In 1978 MCNP users typically did shielding calculations either with the geometry splitting and Russian roulette technique or with the exponential transform technique. For the geometry splitting and Russian roulette technique, the user divided the geometry into cells and manually supplied “cell importance” parameters (IMP card in MCNP) for each cell, based on physical intuition with a few short Monte Carlo runs to confirm and/or modify that intuition. Lacking any expected tally information, I took a set of cell importances that empirically appeared to provide a fairly optimum calculation and produced a weight window using these cell importances as a substitute for the expected tally produced by a unit weight track in the cells. That is, the weight windows were chosen inversely proportional to the manually set cell importances. The constant of proportionality was set so that the unit weight source particles would start within the window.

The result of adding the weight window to the exponential transform was dramatic. Even with extreme exponential transform parameter values, the Monte Carlo results were reliable. The “dial an answer” phenomenon completely disappeared. Furthermore, the combination of the exponential transform and the weight window produced better results than either the geometry splitting and Russian roulette technique or the exponential transform technique. The weight window was then tested by itself and it produced results that were usually a little better than the geometry splitting and Russian roulette technique.

Because the importance in a shielding problem is highly energy-dependent, I decided to make the weight windows space-energy dependent. This decision immediately caused two problems

1. No manually set cell importances could be used to calculate the space-

energy window because the cell importances were spatial only.

2. For 20 spatial cells and 5 energy ranges, this required the user to set 100 weight windows.

Item 2 was a serious problem. First, being new to Monte Carlo transport, I had little intuition about what to guess for an importance function to base the weight windows on. Second, there were 100 windows to manually set, so that even if one did know how to set the windows, doing so was going to be exceptionally tedious.

At this point I decided to make up for my lack of intuition about the importance function by modifying MCNP to estimate the importance function for me. That is, the expected score per unit weight (importance) could be estimated as:

$$region\ importance = \frac{total\ score\ because\ of\ tracks\ entering\ the\ region}{total\ weight\ entering\ the\ region} \quad (1)$$

Note that Eq. 1 permits the use of any variance reduction techniques while the importance is being estimated. This was easy to code and test in MCNP. In doing the bookkeeping indicated in Eq. 1, a space-energy weight window was generated and the "weight window generator" was born.

For Los Alamos' larger problems, there was not enough fast core memory (65K I believe) on the CDC-7600 computer, to keep the information required for the bookkeeping of Eq. 1, so I also devised a modification of Eq. 1. The denominator of Eq. 1 was replaced by an unbiased estimate of the denominator. This modified generator simply set a "flag" bit to 1 if a track passed through the region. When a track terminated (by physical process), the termination weight was used as an estimate of the weight the track had when it passed through the flagged regions. This memory trick was removed sometime later (by someone else after computer memories had expanded) in favor of Eq. 1.

I reported the results of my exponential transform study to Buck Thompson, showing him how a space-energy weight window not only made the exponential transform results reliable, but also substantially increased the efficiency of the calculation. Buck seemed impressed with the results, but his questions were focused on how I chose 100 space-energy weight windows rather than how the windows made the exponential transform reliable. As I was new to Monte Carlo at that time, I explained to Buck that I did not have enough experience yet to guess what the importance function would look like in such a problem, so I had the computer estimate the importance function instead.

Buck was more enthusiastic about the weight window generator than about the solution to the exponential transform puzzle. When it came time to publish, I published what Buck (and others in my group) perceived to be the truly important result of my study, the development of the weight window generator[6]. No paper was ever published on the weight window idea itself. The weight window and the weight window generator ideas basically were developed together.

3 Weight Window Definition in 1978

From time to time, people have inquired about the reason for some of the features and/or recommended parameters for the weight window. In particular,

1. The window always does an integer split.
2. The window must be at least a factor of 2 wide.
3. The upper window bound was recommended to be 5 times the lower window bound.
4. The roulette survival weight was recommended to be 3 times the lower window bound.
5. The maximum split/roulette factor was recommended to be 5.

The first two items are easily justifiable whereas the rest are mostly based on empirical experience in 1978 backed by some plausibility arguments.

1. The window always does an integer split because an integer split introduces no variance in the total post-split weight. (In contrast, the expected value splitting that MCNP's geometry splitting/Russian roulette uses does introduce variance in the total post-split weight.)
2. The window must be at least a factor of 2 wide because otherwise a particle just above the window could be split and its post-split weight would be below the window. Making the window at least a factor of two wide ensures that the split particles will always be within the window.
3. The upper window bound is recommended to be 5 times the lower window bound so that a particle repeatedly crossing back and forth between two regions is not repeatedly subjected to "thrashing", i.e., roulette followed by splitting.

Adjacent window regions often have windows that differ by as much as a factor of 4, so such thrashing is a real possibility if the problem has substantial scattering. Note that the roulette game always introduces variance in the post roulette weight because the particle either is assigned 0 weight (i.e., it is killed) or the particle is assigned increased weight commensurate with the survival probability. Once this variance has been introduced, no splitting game can remove it. It thus seems intuitive to avoid thrashing situations.

4. The roulette survival weight is recommended to be 3 times the lower window bound to avoid playing roulette for small changes in particle weight. For instance, if the roulette survival weight were the lower window bound, then every time an implicit capture technique was used in a region, the window's roulette game would effectively undo the implicit capture game. That is, if the particle's weight is currently at the lower bound ($w = w_l$) and there is a capture probability of 0.1 upon collision, then the implicit capture game will capture $0.1w$ at the collision and assign weight $0.9w$ to the surviving particle. But now $0.9w < w_l$ so that a roulette game will be played by the window with survival probability 0.9. That is, with probability 0.1 the particle disappears and with probability 0.9 the particles weight is increased to w_l. Note that this is the same distribution that occurs if the particle's survival is simply sampled rather than split into captured and surviving fractions. Although this does not increase the sample variance like the thrashing problem alluded to in the previous item,

it increases the time required to obtain a sample because extra games are played that have no effect on the sampled distributions but consume time to play them.

5. The maximum split/roulette factor is recommended to be 5 in an attempt to keep from over splitting when a large importance change is encountered on a single step. For example, the particle might stream up a tiny void duct and change weight window regions by 10^4. A 10^4 split is unlikely to be justifiable even if the particle starts in importance region 1 and ends in importance region 10^4. That is there is a 10^4 importance ratio between the beginning and end of the transport step. Note that the essential variance problem is that a biased sampling procedure should have been used to get the particle up the duct. If a good biased sampling procedure is used, then sampling up the duct will occur far more often and with a correspondingly lower weight (because of the biased sampling) that will then be commensurate with the weight window at the end of the step. Note that an $\infty : 1$ split will eliminate any variance produced *after* the split, but it cannot eliminate the variance produced by the poor sampling up the duct in the first place. Such a biasing up the duct is often not practical for a variety of reasons, so the maximum split parameter is an attempt to save computer time associated with following huge numbers of highly correlated random walks which are unlikely to significantly reduce the variance commensurate with the time that they consume.

   Experience since 1978 has shown that although this feature helps significantly in some cases, there are many other cases in which it does not help very much.

One further note is worthwhile. In most cases, the weight window method is relatively insensitive to the choice of any of these defaulted parameters above.

4 Beyond Genesis

It is worth pointing out the important work John Hendricks and Tom Godfrey did to implement and integrate the initial capability into a production quality code for a final released version of MCNP. John helped test the generator as well as supplying the theory and coding to uniformly populate the geometry. John's uniform population coding was used in our paper (see [6] below) as a way of ensuring that at least some tracks got to the tally region to give the generator some tracks to work with. John's innovative method worked well on the paper's test problems and thus was a useful contribution to the paper.

My contribution largely ended with the publication [6] in 1984. Since then, the basic window and generator ideas have been implemented in MCNP (and other codes) in ever more useful ways to produce ever more sophisticated tools in MCNP (and other codes).

In particular, Hendricks points out that many others have made significant contributions to MCNP's capability since then: Todd Urbatsch, Tom Evans, Jeff Favorite, John Hendricks, Franz Gallmeier, and Gregg McKinney have added the rectangular and cylindrical mesh superimposed weight windows and weight window generators, have enabled the color plotting and other displays of the windows, have extended the windows to other particle types, and more.

Thus, although I alone developed the weight window and generator ideas, it is also obvious that the current window and generator methods in MCNP are much more useful and sophisticated because of the contributions made by many others over many years.

5 Acknowledgement

The author wishes to thank John Hendricks for reading this report and supplying historical information on the subsequent improvements in the weight window and weight window generator at the Los Alamos National Laboratory.

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, http://mcnp-green.lanl.gov/manual.html

[2] W. L. Thompson, O. L. Deutsch, and T. E. Booth, "Deep-Penetration Calculations," A Review of the Theory and Application of Monte Carlo Methods, Proceedings of a Seminar-Workshop, Oak Ridge Tennessee April 21-23, 1980 ORNL/RSIC-44

[3] W. L. Thompson and Edmond D. Cashwell, “The Status of Monte Carlo at Los Alamos,” A Review of the Theory and Application of Monte Carlo Methods, Proceedings of a Seminar-Workshop, Oak Ridge Tennessee April 21-23, 1980 ORNL/RSIC-44

[4] T. E. Booth, Automatic importance estimation in forward Monte Carlo calculations, Transactions of the American Nuclear Society, vol. 41, pp. 308 - 309, 1982.

[5] M. B. Emmett, “The MORSE Monte Carlo Radiation Transport System,” ORNL-4972

[6] Importance Estimation in Forward Monte Carlo Calculations Thomas E. Booth and John S. Hendricks Nuclear Technology/Fusion, vol. 5, Jan. 1984