Proceedings of the Reactor Physics Asia 2015 (RPHA15) Conference
Jeju, Korea, Sept. 16-18, 2015

Development of variance reduction techniques in Monte Carlo code RMC

Xiao Fan [a,b,*], Kan Wang [b] and Guohui Zhang [a]
[a] *State Key Laboratory of Nuclear Physics and Technology, Peking University, Beijing, 100871, P.R. China*
[b] *Department of Engineering Physics, Tsinghua University, Beijing, 100084, P.R. China*
[*]Corresponding author: x.fan@pku.edu.cn

1. Introduction

In the field of nuclear engineering, deterministic and stochastic methods are used to solve radiation transport problems. Deterministic methods solve the transport equation for the average particle behavior and also contain uncertainties associated with the discretization of the independent variables such as space, energy and angle of the transport equation and can admit solutions that exhibit non-physics features [1]. The Monte Carlo method obtains results by simulating individual particles and recording some aspects of their average behavior [2]. This method enables detailed, explicit geometrical, energy and angular representations and hence is considered the most accurate method presently available for solving complex radiation transport problems.

One of the difficulties associated with Monte Carlo method is the amount of computer time required to obtain sufficient precision in the simulations. Despite substantial advancements in computational hardware performance and widespread availability of parallel computers, the computer time required for analog Monte Carlo is still considered exorbitant and prohibitive for the design and analysis of many relevant real-world nuclear applications especially for the problems with complex and large geometry [1]. But there are many ways other than increasing simulation time in the Monte Carlo method by which the precision can be improved. These ways are known as Variance Reduction techniques and required enabling the Monte Carlo calculation of quantities of interest with the desired statistical uncertainty. Without the use of variance reduction techniques in complex problems, Monte Carlo code should run continuously for days or weeks and still cannot obtain statistically significant reliable results. The goal of Variance Reduction techniques is to produce more accurate and precise estimate of the expected value than could be obtained in analog calculation with the same computational efforts [3].

RMC is a Monte Carlo transport code which has been being developed by Department of Engineering Physics, Tsinghua University in China since 2008 as a tool for reactor core analysis on high-performance computing platforms [4]. To meet the requirements of reactor analysis, RMC now has such functions as criticality calculation, fixed-source calculation, burnup calculation and kinetics simulations. Some techniques for geometry treatment, new burnup algorithm, source convergence acceleration, massive tally, parallel calculation, and temperature dependent cross sections processing have been implemented in RMC to improve the efficiency and functions.

In this paper, we mainly present several variance reduction techniques developed and implemented in RMC code recently including geometry splitting/roulette and weight window. Based on weight window technique, a new strategy of inner iterative fixed source calculation is also developed.

2. Methodology

2.1 Figure-of-merit

During a Monte Carlo calculation, according to Central Limit Theorem, the estimated relative error squared R^2 should be proportional to $1/n$, where n is the number of histories. As each history will take on average, the same amount of computer time and the used computer time T in a Monte Carlo calculation should be directly proportional to n. Therefore R^2T should be approximately constant [1]. Thus the metric of efficiency for a given tally, called the figure-of-merit (FOM), is defined as:

$$\mathrm{FOM} = 1 / (R^2 T) \qquad (1)$$

where

R = relative error for the sample mean and

T = computer time for the calculation.

2.2 Geometry splitting/roulette

Geometry splitting with Russian roulette is one of the oldest and most widely used variance-reducing techniques in Monte Carlo codes. When used judiciously, it can save substantial computer time. Splitting generally decreases the history variance but increases the time per history, whereas Russian roulette generally increases the history variance but decreases the time per history [5].

Each cell in the problem geometry setup is assigned an importance I by the user and the value of I should be proportional to the estimated value that particles in the cell have for the quantity being scored.

When a particle of weight W passes from a cell of importance I to one of higher importance I', the particle is split into a number of identical particles of lower weight. If I'/I is an integer n (n>1), the particle is split into n identical particles, each weighting W/n. Weight is preserved in the integer splitting process. If I'/I is not an integer but still great than 1, splitting is done probabilistically so that the expected number of splits is

equal to the importance ratio. Denoting $n = [I'/I]$ to be the largest inter not exceeding I'/I, $p = I'/I - n$ is defined. Then with probability p, $n+1$ particles are used and with probability $1 - p$, n particles are used.

On the other hand, if a particle of weight W passes from one cell of importance I to another of lower importance I', so that $I'/I < 1$, Russian roulette is played and the particle is killed with probability $1 - (I'/I)$, or followed further with probability I'/I and weight $W \cdot I/I'$.

2.3 Weight window

The weight window shown in Fig. 1 is a phase space splitting and Russian roulette technique. For each space, a lower weight bound is supplied and the upper weight bound is a multiple of the lower weight bound. These weight bounds define a window of acceptable weights. If a particle is below the lower weight bound, Russian roulette is played and the particle's weight is either increased to a value within the window or the particle is terminated. If a particle is above the upper weight bound, it is split so that all the split particles are within the window [5]. No action is taken for particles within the window.

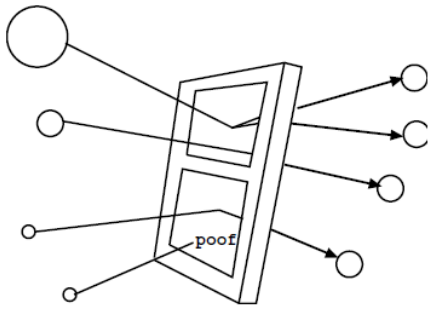

Fig. 1. Weight window [5].

2.3.1 Weight window generator

Variance reduction parameters vary with problem types and it's quite difficult to choose weight window importance functions by guessing, intuition, experience, or trial and error. The weight window generator is a method that automatically generates weight window importance functions, thus the task of choosing importance is simplified.

The importance of a particle at a point P in phase space equals the expected score generated by a unit weight particle. Imagine dividing the phase space into a number of cells or regions [6]. The importance of a cell then can be defined as the expected score generated by a unit weight particle after entering the cell. Thus, the cell's importance can be estimated as:

$$\text{Importance} = \text{score}/\text{weight} \qquad (2)$$

where

score = total score because of particles (and their progeny) entering the cell and

weight = total weight entering the cell.

Although the generator and weight window concepts are independent, they are used in a complementary fashion. By using a window inversely proportional to the importance, the mean score from any track in the problem is roughly constant. In other words, the window is chosen so that the track weight times the mean score is approximately constant [7]. Under these conditions, the variance is mostly due to the variation in the number of contributing tracks rather than the variation in track score.

2.3.2 Mesh-based weight window

The geometry of the problems must be divided when no single set of weight window parameters can be representative of a whole region. By using mesh-based weight window, a mesh importance grid map is generated and superimposed, which is a good way to subdivide the space without complicating the cell geometry. The mesh-based weight window method makes it more convenient to use weight window technique.

2.4 Inner iteration for fixed source calculation based on weight window

Traditionally, fixed source shielding calculations using weight window require trial and error and typically involve many steps and several iterations to get a set of reasonable weight window parameters.

To improve efficiency and simplify the usage of weight window, a new strategy of inner iterative fixed source calculation is developed.

A single fixed source calculation is divided into several batches. The first batch can be run with or without variance reduction techniques and generates a set of weight window parameters which are then used in the second batch. A set of more reasonable parameters are generated after the second batch run and those are then used by the next batch. The number of batches is assigned by the user. Fig. 2 shows the flow chart of this strategy when the calculation is divided into 3 batches.

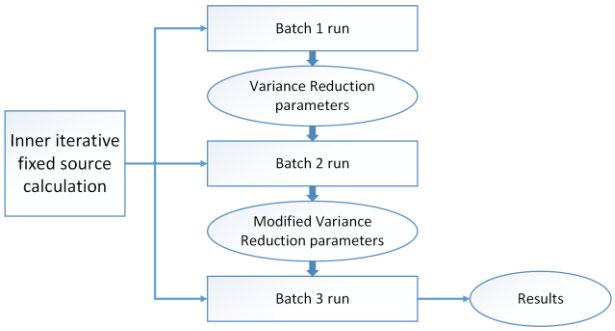

Fig. 2. Flow chart of inner iterative fixed source calculation with weight window with batch number 3.

The inner iteration method avoids repeatedly reading input and generating output, which saves time and

decreases data transmission error.

One of the most common frustrations when using the weight window generator is the output of zero weight windows [8]. There are two reasons that zero weight window is generated from a region: no particle have entered the region or particle have entered the region but none of those that entered scored. The inner iteration method can use tallies that are highly correlated with the ultimate tally desired but easier to obtain in first few batches and use the ultimate tally in the last several batches taking advantage of the weight windows generated previously. Thus the problem of zero weight windows can be optimized by inner iteration method.

By employing this new strategy, the importance map or variance reduction parameters are produced and used, and the expected scores are obtained during a single fixed source calculation.

3. Numerical Verification

In order to verify and test the performance of these updated variance reduction techniques implemented in RMC, several numerical tests are made. All the following calculations are run a laptop with an Intel Core i5-3210M CPU.

A plate model is constructed to calculate the shielding of a neutron source, the side length of which is 100cm and is divided into 10 layers or cells uniformly as shown in Fig. 3. The material inside is water and a monoenergetic 2 MeV isotropic point neutron source is placed in cell 1.

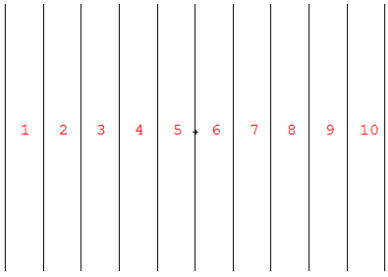

Fig. 3. Plate model.

3.1 Cell importance and weight window

The model is calculated four times by RMC code using different techniques and all these calculations are performed for 1 minute on the same computer. The relative errors of neutron flux in each cell are obtained and shown in Fig. 4.

After one minute calculation, without using specific variance reduction techniques, the relative error of neutron flux is above 20% when the distance is greater than 60 cm, as shown in the black line.

By adopting cell importance and weight window and assigning the parameters by experience, the calculation precision becomes better, as the red and blue lines show. But because these cell importance and weight window parameters are not optimized and just given by guess and experience, the calculation efficiency is improved not enough.

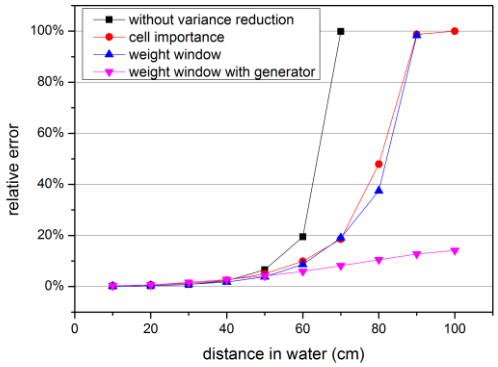

Fig. 4. Results of one minute calculations.

By using weight window with parameters from weight window generator, the results are much better and all the relative errors are under 20%, as the pink line shows. It can be said that the weight window technique is effective but the parameters are critical. Only with optimized parameters, can weight window play an efficient role.

3.2 Mesh-based weight window

While this plate model is simple and small, there are many real problems with complex and large geometry which need mesh weight window and mesh tally. For testing, this model is superimposed by a 5×20 grid and the importance of each small grid is assigned by mesh weight window generator.

Using mesh tally, the calculation is performed twice, one with mesh weight window and the other without. As no single measurement like FOM can be applied, instead comparing what fraction of voxels have less than some amount of relative error is shown in Fig. 5.

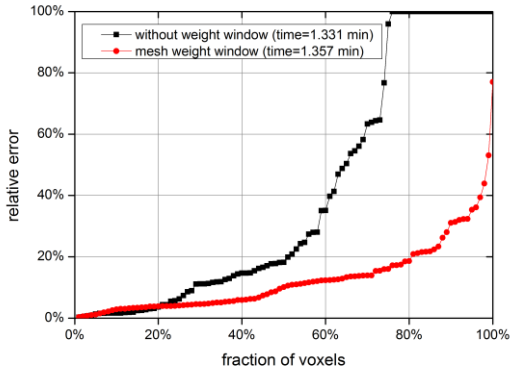

Fig. 5. Relative error histogram of mesh tally.

The calculation time is 1.331 minutes and 1.357 minutes, respectively, which are almost the same, but the efficiency is quite different. By using mesh weight window, there are about 80% of the tallies whose relative errors are under 20%. On the other hand, without

using weight window, only about 50% of the relative errors are under 20%. The comparison shows high efficiency of the use of mesh-based weight window.

3.3 Inner iterative fixed source calculation

Although using weight window with generator is quite efficient, it has to make iterative calculations to generate a set of reasonable parameters. For this model, two whole fixed source calculations are made to get a set of reasonable weight window parameters before the third calculation is performed to obtain the final results.

As comparison, an inner iterative fixed source calculation with weight window is made with batch number 3. The total number of initial neutron histories of these two strategies is the same which is 150,000.

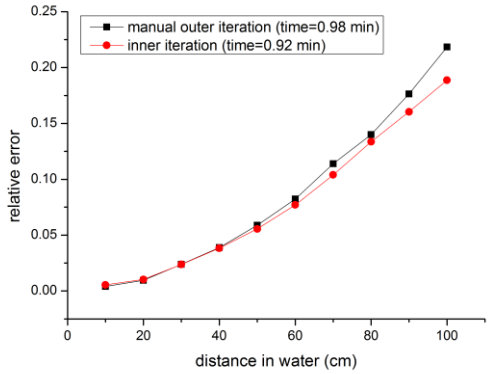

Fig. 6. Comparison of two methods.

Table I. Results of 100-cm-thick water penetration

Method	Neutron flux tally	Relative error	Time (min)	FOM
Manual outer iteration	9.76E-08	0.218	0.98	21.5
Inner iteration	9.30E-08	0.189	0.92	30.4

Fig. 6 and Table I show the results and comparison of traditional manual multistep weight window method and inner iterative fixed source calculation. The FOM (/min) for this 100-cm-thick water penetration problem is 21.5 and 30.4, respectively. The inner iteration method avoids repeatedly reading input and generating output, which saves time and decreases data transmission error. It can be said that the new inner iteration method not only makes it more convenient to use weight window but also improves the variance reduction efficiency.

4. Conclusions and future work

Several traditional and effective variance reduction methods including geometry splitting/roulette and weight window technique/weight window generator have been implemented in RMC code. These methods improve the calculation precision greatly and are of necessity for Monte Carlo codes.

Based on the weight window technique, a new inner iteration method for fixed source calculation is also developed, which improves both convenience and efficiency.

The inner iteration in fixed source calculation is still a rough strategy and method. Some aspects need more research later, such as the tradeoff between particle number per batch and the batch number.

Acknowledgements

This research work is financially supported by Key Laboratory of Neutron Physic (CAEP) - 2012AC01, National Natural Science Foundation of China (11175005 and 11475007) and the China Scholarship Council.

References

1. Chan WK, Theory and Applications of Monte Carlo Simulations, chapter 7, InTech (2013)
2. Carter LL, Cashwell ED, “Particle transport simulation with the Monte Carlo method,” ERDA Critical Review Series (1975).
3. Osnes H, “Variance reduction techniques for Monte Carlo simulation,” ISSN 0808-2839, SINTEF and University of Oslo (1997).
4. Wang K, et al., “RMC – A Monte Carlo code for reactor core analysis,” Annals of Nuclear Energy 82:121-129(2015).
5. 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(2003).
6. Brown FB, “Fundamentals of Monte Carlo Particle Transport,” Los Alamos National Laboratory, Report LA-UR-05-4983(2005).
7. Booth TE, Hendricks JS, “Importance estimation in forward Monte Carlo calculation,” Nucl. Tech. /Fusion, 5(1984).
8. Booth TE, “MCNP Variance Reduction Examples,” Los Alamos National Laboratory, Report LA-UR-12-25907(2004).