7th International Conference on Modelling and Simulation in Nuclear Science and Engineering (7ICMSNSE)
Ottawa Marriott Hotel, Ottawa, Ontario, Canada, October 18-21, 2015

7ICMSNSE-008

Comparison of different strategies for global tallying in Monte Carlo criticality calculation

DanHua ShangGuan[1], BaoYin Zhang[1] , Gang Li[1] , Li Deng[1] , Yan Ma[1], YuanGuan Fu[2], Rui Li[2] and XiaoLi Hu[2]

[1] Institute of Applied Physics and Computational Mathematics, Beijing, China
[2] CAEP Software Center for High Performance Numerical Simulation, Beijing, China
sgdh@iapcm.ac.cn, zby@iapcm.ac.cn, li_gang@iapcm.ac.cn, deng_li@iapcm.ac.cn, mayan2301@126.com, fyg1989@126.com, li_rui@iapcm.ac.cn, hu_xiaoli@iapcm.ac.cn

Abstract

We propose the uniform tally density algorithm and the uniform track number density algorithm for biasing the fission secondary neutron number in active cycles of the Monte Carlo criticality calculation when the target is seeking high performance of some global tally and compare these strategies with the original uniform fission site algorithm. Using the global volume averaged cell flux tally and the global energy deposition tally of the pin-by-pin model of Dayawan nuclear reactor as examples, the efficiencies of these strategies are compared carefully. All the strategies are realized in a recently developed parallel Monte Carlo particle transport code JMCT.

Keywords: Monte Carlo Method, Criticality Calculation, Global Tally.

1. Introduction

It is well known that the power iteration method is the most important technique in general Monte Carlo simulation codes as a powerful tool for criticality calculations [1]. For obtaining an accurate multiplication factor and exact global tallies, the so called inactive iteration cycles must be discarded until the source distribution has converged. Only then, the tallying action should be invoked in the following active cycles. Some techniques, which are still in progress [2-6], have been developed to accelerate the convergence rate. But if we want to get some tallies in all cells, this global tallying problem can suffer one difficulty even when the source distribution has converged perfectly. Based on the fact that relative uncertainties of local tallies tend to be large in low-power regions and small in higher-power regions, reducing most uncertainties to an acceptable level simply by running a large number of histories is often prohibitively expensive. At the same time, it is widely accepted that the goal of global tallying should be put on the decrease of statistical errors of tallies in most cells. Some authors propose 95% of all regions should have a relative error less than 1% based on a 95% confidence interval [7]. So, the uniform fission site method (UFS algorithm) has been developed in MC21 code and gets better results when tested with some benchmark models [8-10]. The main idea of this algorithm is biasing the fission secondary neutron number in active cycles based on specifically data obtained from past active cycles.

In this paper, we propose two different strategies for biasing the fission secondary neutron number based on other data obtained from past active cycles. One is called the uniform track number density algorithm (UTND algorithm) and relies on the volume averaged density of neutrons' track number

in each cell. The other is called the uniform tally density algorithm (UTD algorithm) and relies on the volume averaged density of target tally in each cell. Although these strategies are stimulated by the UFS algorithm, they may be more efficient intuitively and will be illustrated in detail in Sec.3.

All these strategies (including the UFS algorithm) have been realized in the JMCT (Jointed Monte Carlo Transport) code [11], which is a parallel Monte Carlo particle transport code developed by a team in Institute of Applied Physics and Computational Mathematics (IAPCM). This code is constructed on the JCOGIN (J COmbinatorial Geometry Monte Carlo transport INfrastructure) framework [12]. The basic idea underlying this development route is to lett JCOGIN deals with geometry modelling and high performance parallelization. Utilizing the tools supplied by this infrastructure, JMCT code can focus on physics and simulation techniques. By comparing the numerical results of many benchmark models obtained by JMCT with results from other programs and experiments, this code has been validated and verified adequately [13].

The remainder of this paper is organized as follows. In Sec.2, we review the modified UFS algorithm according to the power iteration scheme of JMCT code. In Sec.3, we explain the main idea of the UTND algorithm and the UTD algorithm. Numerical results of these strategies, tested by the pin-by-pin model of Dayawang nuclear reactor, are compared carefully in Sec.4. Sec.5 gives a summary of this paper.

2. Modified UFS algorithm in JMCT code

Unlike the situation of MC21 code, the standard procedure of JMCT code for computing the expected value of fission secondary neutron number m_{JMCT}^{std} in criticality calculation is given by [11]

$$m_{JMCT}^{std} = w_{in} \frac{\gamma\Sigma_f}{\hat{k}_{eff}\Sigma_t}, \tag{1}$$

where

$\hat{k}_{eff}$ =the multiplication factor estimated in last cycle,

w_{in} =the scoring weight for the neutron undergoing collision,

$\gamma\Sigma_f$ =the macroscopic neutron production for the region in which the collision occurs ,

Σ_t =the macroscopic total cross section for the region in which the collision occurs .

At the beginning of next cycle, all sites created in last cycle are tracked with an initial weight which equals N_{hist} / N_{bank}. Here, N_{hist} is the history number and N_{bank} is the number of all created sites in last cycle. Note that the actual tracked particle number in next cycle is N_{bank}, N_{hist} is only used to set the weight of source particle.

Apparently, according to the idea of original UFS algorithm, the expected value of fission secondary neutron number m_{JMCT}^{UFS} in UFS algorithm of JMCT code should be given by [11]

$$m_{JMCT}^{UFS} = w_{in} \frac{\gamma\Sigma_f}{\hat{k}_{eff}\Sigma_t} \frac{v_k}{s_k}. \tag{2}$$

Here，

v_k = the fraction of V occupied by cell k in which the collision occurs,

V = the volume of problem domain that comprises all fissionable material,

s_k = the fraction of fission source contained in cell k.

But if we track all the fission neutrons created according to Eq(2) in next cycle (with an initial weight $\frac{N_{hist}}{N_{bank}}\frac{s_k}{v_k}$), the calculation is too time-consuming and the benefit of UFS algorithm will be canceled. So, alternatively, we track every fission site with probability $p = N_{hist} / N_{bank}$ and the weight of surviving fission site should be adjusted from $\frac{N_{hist}}{N_{bank}}\frac{s_k}{v_k}$ to $\frac{N_{hist}}{N_{bank}}\frac{s_k}{v_k p}$.

3. The UTND algorithm and UTD algorithm

As explained before, the target of UFS algorithm is to reduce most statistical errors to some acceptable level. However, the number v_k / s_k, in which s_k is explained as the fraction of fission source contained in cell k, is not the only candidate for biasing the fission secondary neutron number. Intuitively, if the object is to increase the global performance of some specific global tally, the corresponding tally density and track number density, which is defined by the result of the fraction of target tally in cell k divided by v_k and the result of the fraction of neutrons' track number in cell k divided by v_k, may be more suitable. For the UTD algorithm, the expected value of fission secondary neutron number m_{JMCT}^{UTD} is given by

$$m_{JMCT}^{UTD} = w_{in}\frac{\gamma\Sigma_f}{\hat{k}_{eff}\Sigma_t}\frac{v_k}{t_k}. \tag{3}$$

Here, t_k is the fraction of target tally in cell k. For the UTND algorithm, the expected value of fission secondary neutron number m_{JMCT}^{UTND} is given by

$$m_{JMCT}^{UTND} = w_{in}\frac{\gamma\Sigma_f}{\hat{k}_{eff}\Sigma_t}\frac{v_k}{d_k}. \tag{4}$$

Here, d_k is the fraction of neutrons' track number in cell k. Other procedures are similar with the UFS algorithm.

As a substitute of setting mesh cell to decrease the fluctuation of s_k in small cell k, in our scheme, a combination scale number M will be set by the user. The number s_k, t_k and d_k should be counted by considering M true cells which contain fissionable materials as a whole. This is the only parameter which should be set by experience. The user does not need to specify the true cells in each group. The program will automatically chooses M true cells as a whole. Furthermore, considering M fissionable cells as a whole has another meaning. Because the fissionable cell number of pin-by-pin model is usually huge and the parallel programming language of JMCT code is MPI, reducing the number s_k, t_k and d_k of all fissionable cells is expensive when compared with reducing the corresponding number of M_s virtual cells. Here, $M_s = M_{cell} / M$ and M_{cell} is the total fissionable cell number. Another point which is noticeable in our scheme is, the number s_k, t_k and d_k is not the number counted in last finished active cycle, but the fraction of the sum of corresponding numbers counted in all past active cycles, this choice will decrease the fluctuation remarkably, as suggested by Kelly [8]. If some s_k, t_k or d_k are zero

in the first active cycle, then Eq(1) will be used to generate secondary neutrons and the weight of the corresponding source particle will not be adjusted.

4. Comparison of numerical results

For numerical tests, the UFS , UTND and UTD strategy are implemented into the Monte Carlo code named JMCT and tested on the Dayawan pin-by-pin nuclear reactor model. Because it is unfair to give the same up-limit and down-limit of survival bias to different strategies, all algorithms are tested with no survival bias technique.

4.1 The Dayawan pin-by-pin nuclear reactor model

The pin-by-pin model from the Dayawan Nuclear Power Station was taken as a test model. As shown in Fig.1, there are totally 157 repeated assemblies in the center of a cylinder tank. Each assembly has 25 control rods and 264 fuel rods in some 17×17 array. In radial direction each fuel rod is divided into two coaxial layers. The outer layer is made of Zirconium and the inner of Uranium. In axial direction each fuel rod is divided into 16 segments. There are totally 758973 cells in this model and the fissionable cell number exceeds 600000.

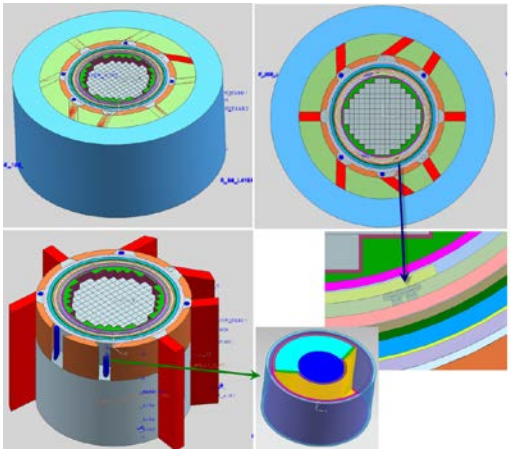

Figure 1 The Dayawan pin-by-pin nuclear reactor model

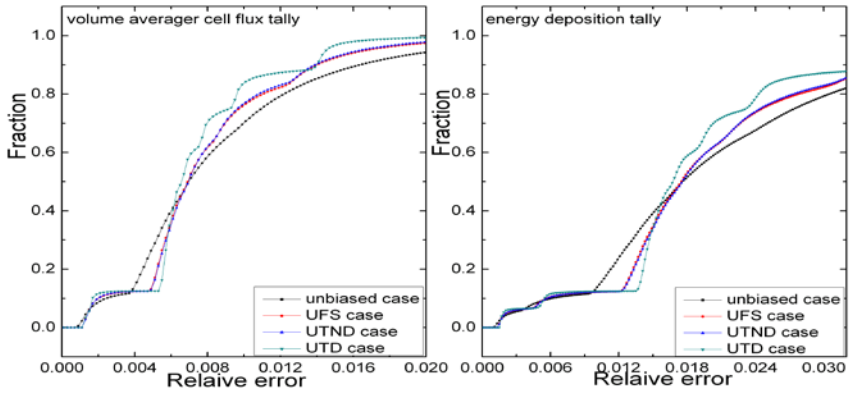

Figure 2 Cumulative distributions of relative uncertainties

4.2 Numerical results

Here, we mainly focus on the global volume averaged cell flux tally and energy deposition tally. There are totally 600 generations and 5000000 particles per generation in each calculation. The first 300 generations are discarded. The final number of virtual cells equals 600(Actually, we first make the final number equals 600 and then calculate the corresponding scale number M. Because the total cell number can not be divided exactly by this M, the last group will contains less true cells) and 300 CPUs are utilized. All the calculations use the same initial neutron source distribution.

It must be mentioned that, for the UTD algorithm, the flux tally density will be used to bias the fission secondary neutron number for the global volume averaged cell flux tally and the energy deposition tally density will be used to bias the fission secondary neutron number for the global energy deposition tally. We do not hope to get better results for the global volume averaged cell flux tally by using the energy deposition tally density, and vice versa. That is to mean, the UTD bias works for only one tally type at a time.

From Table I, we can conclude all strategies do not bias the eigenvalue. Because of the absence of survival bias, the start weights of source particles of UFS, UTND and UTD case have more fluctuations than those of unbiased case. So, the uncertainties of the multiplication factor of all these cases are bigger but still at low levels (because we have modified the original material composition, this eigenvalue is slightly bigger than one. But the geometry is true). In paper[10], it was shown that when the UFS algorithm was utilized with suitable survival bias, it has a little effect on the uncertainty of the engenvalue.

TABLE Ⅰ
The multiplication factor and its uncertainty

	multiplication factor	**uncertainty**
Unbiased case	1.01342	1.67738E-5
UFS case	1.01339	2.31931E-5
UTND case	1.01343	2.41103E-5
UTD case	1.01337	2.38019E-5

From Table Ⅱ and Table III, we can see that the UTD strategy is the most efficient strategy according to the FOM-MAX index and the FOM-95 index, which is the inverse of the product of run time with the square of maximum uncertainty and 95th percentile value (for example, if there are 200 cells in total, this value will be the tenth largest one), respectively. The FOM-MAX index of UTD strategy is about 100%~200% higher than that of UFS strategy and the FOM-95 index of UTD strategy is about 25%~35% higher than that of UFS case. However, these two index values are almost the same for the UTND algorithm as for the UFS algorithm. This phenomenon means the UTND algorithm and UFS algorithm may have similar performance. Note that the UFS algorithm and UTND algorithm use the same quantity(fission site density and track number density, respectively) for all these two tally types, but the UTD algorithm uses different tally density for them. It can also be deduced that, because the volume averaged cell flux tally and energy deposition tally are different tallies, the latter has a much bigger maximum uncertainty than the former. This means it is more difficult for the energy deposition tally to reach the 95/95 standard.

The above-mentioned conclusions can be deduced more clearly from Fig.2, which plots the cumulative distribution of relative uncertainties for all cases. The abscissa represents the magnitude of relative

uncertainty and the ordinate means the fraction of all cells whose relative uncertainties are less than the indicated value. Although in unbiased case there are more cells whose relative uncertainties lie in some

TABLE Ⅱ
The comparison of efficiency of the global volume averaged cell flux tally

Unbiased case (Run time: 4792 seconds**)**	**Relative Uncertainty**			**FOM**	
	min	**max**	**95th percentile**	**max**	**95**
	2.90612E-4	0.07608	0.02096	0.03605	0.47501
UFS case (Run time: 4856 seconds**)**	**Relative Uncertainty**			**FOM**	
	min	**max**	**95th percentile**	**max**	**95**
	2.25274E-4	0.06255	0.01681	0.05263	0.72876
UTD case (Run time: 5167 seconds**)**	**Relative Uncertainty**			**FOM**	
	min	**max**	**95th percentile**	**max**	**95**
	1.68955E-4	0.04143	0.01460	0.11275	0.90794
UTND case (Run time: 5013 seconds**)**	**Relative Uncertainty**			**FOM**	
	min	**max**	**95th percentile**	**max**	**95**
	2.14960E-4	0.05657	0.01649	0.06233	0.73302

interval near zero point, which means these cells are in higher-power regions, the UTD case has more cells whose relative uncertainties are less than some bigger value, even compared with the UFS case and UTND case. The reason is apparent: all these algorithms decrease the track number in those cells whose relative uncertainties are small enough and increase the track number for a larger fraction of all cells. The UTD algorithm does even better than the UFS algorithm and UTND algorithm evidently. From Fig.2, we can find the UTND algorithm has very similar performance with the UFS algorithm.

Because we do not know the dominance ratio of this model, the uncertainties we got may be severely underestimated if this ratio is too close to one. So, we run 20 independent simulations with 250000 particles per generation. The results are shown in Table IV. From this table, we can see the uncertainties are really underestimated because of the correlation among generations. But the UTD algorithm still has a better performance than the UFS and UTND algorithm for the global volume averaged cell flux tally. Note that when eliminating the effect of the cycle-to-cycle correlation, the UTND algorithm has a better performance than the UFS algorithm, not like the conclusion from Table II and Fig.2

TABLE III
The comparison of efficiency of the global energy deposition tally

Unbiased case (Run time: 4951 seconds**)**	**Relative Uncertainty**			**FOM**	
	min	**max**	**95th percentile**	**max**	**95**
	3.33812E-4	0.23869	0.05144	0.00355	0.07633
UFS case (Run time: 4993 seconds**)**	**Relative Uncertainty**			**FOM**	
	min	**max**	**95th percentile**	**max**	**95**
	2.59528E-4	0.18372	0.04170	0.00593	0.11518
UTD case (Run time: 5046 seconds**)**	**Relative Uncertainty**			**FOM**	
	min	**max**	**95th percentile**	**max**	**95**
	1.95599E-4	0.11265	0.03667	0.01562	0.14738
UTND case (Run time: 4966 seconds**)**	**Relative Uncertainty**			**FOM**	
	min	**max**	**95th percentile**	**max**	**95**
	2.47964E-4	0.19081	0.04110	0.00553	0.11921

TABLE IV
Efficiency of the global volume average cell flux tally using 20 independent runs

UFS case (Run time: 3554 seconds)	**Relative Uncertainty**			**FOM**	
	min	**max**	**95th percentile**	**max**	**95**
	0.00152	0.07369	0.022	0.05182	0.58135
UTD case (Run time: 3424 seconds)	**Relative Uncertainty**			**FOM**	
	min	**max**	**95th percentile**	**max**	**95**
	0.00125	0.06043	0.021	0.07064	0.66226
UTND case (Run time: 3425 seconds)	**Relative Uncertainty**			**FOM**	
	min	**max**	**95th percentile**	**max**	**95**
	0.00107	0.06889	0.021	0.06152	0.66206

Fig.3 is the plot of relative difference of the global flux tally between the unbiased case and all biased case. We can see the major part of relative differences is below 8%. In principle, all algorithms are unbiased. If running more particles per generation, the relative difference can be reduced to a smaller level.

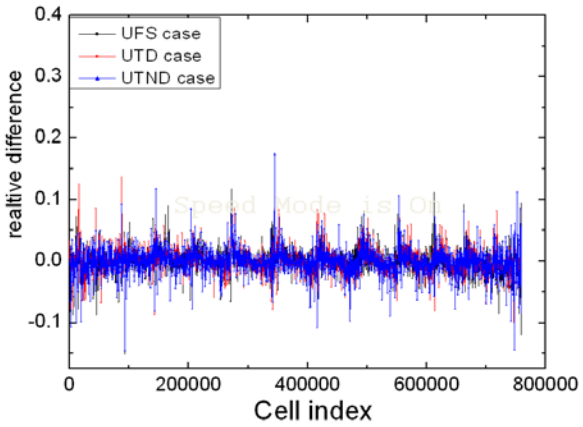

Figure 3 relative differences of the global flux tally between the unbiased case and all biased case

5. Conclusions

In this paper, we have proposed two strategies for biasing the fission secondary neutron number in active cycles of criticality calculation. The UTD strategy is based on the specified tally density and the UTND strategy is based on the track number density. All these densities are gotten from past active cycles. Although the basic idea is inspired by the uniform fission site algorithm apparently, we think the UTD algorithm is more efficient than the origin uniform fission site based strategy intuitively. The reason partly relies on the fact that the biasing rule of this new strategy is more relevant with the target global tally. Our numerical results have shown this conclusion is true. At the same time, we have shown the UTND algorithm has a slightly higher performance than the UFS algorithm.

6. Acknowledgments

The research work is supported by the sub-item of special projects of the National Energy Bureau (2015ZX06002008), the Key Program of the National Natural Science Foundation of China (91118001)，the National High Technology Research and Development Program (863) of China

(2012AA01A303)，the Nuclear Power Development Project of the Science, Technology and Industry for National Defense of China ([2012]1523) and the Science Foundation of China Academy of Engineering Physics of China (2014B0202029)

7. References

[1] F. B. Brown, "A Review of Monte Carlo Criticality Calculations – Convergence, Bias, Statistics," Proc. Int. Conf. Mathematics, Computational Methods & Reactor Physics, Saratoga Springs, New York, 2009 May 3-7.

[2] R. N. Blomquist and E. M. Gelbard, "Alternative Implementations of the Monte Carlo Power Method", Nucl. Sci. Eng., **141**, 85 (2002)

[3] T. Yamamoto and Y. Miyoshi, "Reliable Method for Fission Source Convergence of Monte Carlo Criticality Calculation with Wielandt's Method", J. Nucl. Sci. Technol., **41**, 2(2004).

[4] E. W. Larsen and J. Yang, "A Functional Monte Carlo Method for k-Eigenvalue Problems", Nucl. Sci. Eng., **159**, 107 (2008).

[5] E. R. Wolters, E. W. Larsen, and W. R. Martin, "Hybrid Monte Carlo – CMFD Method for Accelerating Fission Source Convergence", Nucl. Sci. Eng., **174**, 286 (2013).

[6] H. Kadotani et al, "Acceleration of Fission Distribution Convergence Using Eigenvectors from Matrix K Calculations in the KENO Code", ICNC 91, Oxford, UK, 1991 Sept 9-13

[7] Kord S. Smith, "Monte Carlo for Practical LWR Analysis: What's Needed to Get to the Goal", Transactions of the American Nuclear Society, **104**, 313 (2011).

[8] Daniel J Kelly, Thomas M Sutton and Stephen C Wilson, "MC21 Analysis of the Nuclear Energy Agency Monte Carlo Performance Benchmark Problem", Proceedings of PHYSOR 2012-Advances in Reactor Physics-Linking Research, Industry, and Education, Knoxville, Tennessee, USA, 2012 April 15-20.

[9] Daniel J Kelly, Brian N Aviles and Bryan R Herman, "MC21 Analysis of The MIT PWR Benchmark: Hot Zero Power Results", M&C 2013, Sun Valley, Idaho, USA, 2013 May 5-9.

[10] Jessica L Hunter, Thomas M Sutton, "A Method for Reducing the Largest Relative Errors in Monte Carlo Iterated-Fission-Source Calculations", M&C 2013, Sun Valley, Idaho, USA, 2013 May 5-9.

[11] DanHua Shangguan, et al., "Modified Uniform-Fission-Site Algorithm in Monte Carlo Simulation of Reactor Criticality Problem", Acta. Phys. Sin., **64**, 052801 (2015).

[12] Baoyin Zhang, et al., "JCOGIN: A Parallel Programming Infrastructure for Monte Carlo Particle Transport ", PHYSOR 2014 – The Role of Reactor Physics Toward a Sustainable Future, The Westin Miyako, Kyoto, Japan, 2014 Sept 28 – Oct 3.

[13] Li Deng, et al., "3-D Monte Carlo Neutron-Photon Transport Code JMCT And Its Algorithms", PHYSOR 2014 – The Role of Reactor Physics Toward a Sustainable Future, The Westin Miyako, Kyoto, Japan, 2014 Sept 28 – Oct 3.