Title
Calculation of adjoint-weighted kinetic parameters with the reactor Monte Carlo code RMC

Authors
Qiu, Yishu
Wang, Zijie
Li, Kaiwen
et al.

Publication Date
2017-11-01

DOI
10.1016/j.pnucene.2017.03.023

Peer reviewed

Progress in Nuclear Energy xxx (2017) 1–11

Contents lists available at ScienceDirect

Progress in Nuclear Energy

journal homepage: www.elsevier.com/locate/pnucene

Calculation of adjoint-weighted kinetic parameters with the Reactor Monte Carlo code RMC

Yishu Qiu [a, b, *], Zijie Wang [b, c], Kaiwen Li [a], Yuan Yuan [a], Kan Wang [a], Massimiliano Fratoni [b]

[a] Department of Engineering Physics, Tsinghua University, Beijing, 100084, PR China
[b] Department of Nuclear Engineering, University of California, Berkeley, CA 94720, USA
[c] College of Science, Harbin Engineering University, Harbin, 150001, PR China

ARTICLE INFO

Article history:
Received 14 September 2016
Received in revised form
17 February 2017
Accepted 21 March 2017
Available online xxx

Keywords:
Kinetic parameters
Adjoint
Iterated fission probability
Superhistory method
RMC

ABSTRACT

In this work, the capability of computing adjoint-weighted kinetic parameters, including effective delayed neutron fraction and neutron generation time, was implemented in the Reactor Monte Carlo (RMC) code based on the iterated fission probability (IFP) method. Three algorithms, namely, the Non-Overlapping Blocks (NOB) algorithm, the Multiple Overlapping Blocks (MOB) algorithm and the super-history algorithm, were implemented in RMC to investigate their accuracy, computational efficiency and estimation of variance. The algorithms and capability of computing kinetic parameters in RMC were verified and validated by comparison with MCNP6 as well as experimental results through a set of multi-group problems and continuous-energy problems.

1. Introduction

The Reactor Monte Carlo (RMC) code (Wang et al., 2015a; 2015b), developed by the Department of Engineering Physics, Tsinghua University, is a modern Monte Carlo particle transport code mainly designed for nuclear reactor modeling and simulations. In the previous work, a capability of quasi-static dynamic calculation (Xu and Wang, 2014) was developed in RMC. However, it should be noted that RMC computes the point reactor kinetic parameters by weighting a constant instead of adjoint flux due to the difficulty of obtaining adjoint flux in continuous-energy Monte Carlo transport calculations. Recently, based on the concept of iterated fission probability (IFP), capabilities of computing adjoint-weighted kinetic parameters have been implemented in several Monte Carlo codes including MCNP5 (Kiedrowski and Brown, 2009; Nauchi and Kameyama, 2010), McCARD (Choi and Shim, 2014), TRIPOLI-4 (Truchet et al., 2013), SERPENT2 (Leppänen et al., 2014), etc. In previous works, the IFP method was adopted by RMC to compute eigenvalue sensitivity coefficients with regard to nuclear data (Qiu et al., 2015). Therefore, these experiences should be used to compute adjoint-weighted kinetic parameters and improve the dynamic simulation capability in RMC. In this work, three algorithms based on the IFP method were implemented in RMC to investigate their accuracy and computational efficiency of computing adjoint-weighted kinetic parameters. These algorithms can apply in continuous-energy Monte Carlo simulations as well as multigroup Monte Carlo simulations. Additionally, as several studies have found that the correlation of fission source between cycles may cause underestimation of variance of k_{eff} and fluxes (Gelbard and Prael, 1990), the differences in estimation of variance of k_{eff} and kinetic parameters among different algorithms were compared.

2. Theory

2.1. Adjoint-weighted kinetic parameters

The adjoint-weighted kinetic parameters this work is concerned with are the effective delayed neutron fraction (β_{eff}) and the neutron generation time (Λ_{eff}) which are defined as

* Corresponding author. Department of Engineering Physics, Tsinghua University, Beijing, 100084, PR China.
E-mail address: qys12@mails.tsinghua.edu.cn (Y. Qiu).

http://dx.doi.org/10.1016/j.pnucene.2017.03.023
0149-1970/

$$\beta_{eff} = \frac{\left\langle \Psi^*, F^D \Psi \right\rangle}{\left\langle \Psi^*, F\Psi \right\rangle}, \tag{1}$$

$$\left\langle \Psi^*, \frac{1}{\nu}\Psi \right\rangle = \int_V \int_0^\infty \int_0^{4\pi} \Psi^*(r,\Omega,E)\frac{1}{\nu}\Psi(r,\Omega',E')d\Omega dE dV, \tag{5}$$

$$\left\langle \Psi^*, F^D \Psi \right\rangle = \int_0^\infty \int_V \int_{4\pi} \Psi^*(r,\Omega,E)\beta\frac{\chi^D(r,E)}{4\pi}\int_0^\infty\int_0^{4\pi}\bar{\nu}\sum_f(r,E')\Psi(r,\Omega',E')d\Omega' dE' d\Omega dV dE, \tag{6}$$

$$\left\langle \Psi^*, F\Psi \right\rangle = \int_0^\infty \int_V \int_{4\pi} \Psi^*(r,\Omega,E)\frac{\chi(r,E)}{4\pi}\int_0^\infty\int_0^{4\pi}\bar{\nu}\sum_f(r,E')\Psi(r,\Omega',E')d\Omega' dE' d\Omega dV dE, \tag{7}$$

$$\Lambda_{eff} = \frac{\left\langle \Psi^*, \frac{1}{\nu}\Psi \right\rangle}{\left\langle \Psi^*, F\Psi \right\rangle}, \tag{2}$$

where

Ψ is the neutron flux $\Psi(r,\Omega,E)$ which is a variable of position r, direction Ω, energy E,
ν is the speed of neutron,

$F^D\Psi$ is the delayed neutron production term which can be expressed as

$$F^D\Psi = \beta\frac{\chi^D(r,E)}{4\pi}\int_0^\infty\int_0^{4\pi}\bar{\nu}\sum_f(r,E')\Psi(r,\Omega',E')d\Omega' dE', \tag{3}$$

$F\Psi$ is the total neutron production term which can be expressed as

$$F\Psi = \frac{\chi(r,E)}{4\pi}\int_0^\infty\int_0^{4\pi}\bar{\nu}\sum_f(r,E')\Psi(r,\Omega',E')d\Omega' dE', \tag{4}$$

χ^D is the delayed neutron emission spectrum,
χ is the total neutron emission spectrum,
β is the total delayed neutron fraction,
Ψ^* is the adjoint flux,
and $\langle\rangle$ is the integration over all space, angle, and energy variables.

where V means the entire space of the system in question.

As it is known, the flux Ψ is the solution of the Boltzmann Equation

$$B\Psi = \frac{1}{k}F\Psi, \tag{8}$$

where

$B\Psi$ is the transport operator which can be expressed as

$$B\Psi = \Omega\cdot\nabla\Psi(r,\Omega,E) + \sum_t(r,E)\Psi(r,\Omega,E) - \int_0^\infty \times \int_{4\pi}\sum_s(r,\Omega'\to\Omega,E'\to E)\Psi(r,\Omega',E')d\Omega' dE', \tag{9}$$

and k is the effective multiplication factor.

And the adjoint flux Ψ^* is the solution of the adjoint Equation

$$B^*\Psi^* = \frac{1}{k}F^*\Psi^*, \tag{10}$$

where

$B^*\Psi^*$ is the adjoint transport operator which can be expressed as

$$B^*\Psi^* = -\Omega\cdot\nabla\Psi^*(r,\Omega,E) + \sum_t(r,E)\Psi^*(r,\Omega,E) - \int_0^\infty\int_{4\pi}\sum_s(r,\Omega\to\Omega',E\to E')\Psi^*(r,\Omega',E')d\Omega' dE', \tag{11}$$

The terms on the right hand side of Eqs. ((1)–(2)) can be expressed in the form of

and $F^*\Psi^*$ is the adjoint neutron production term which can be expressed as

Please cite this article in press as: Qiu, Y., et al., Calculation of adjoint-weighted kinetic parameters with the Reactor Monte Carlo code RMC, Progress in Nuclear Energy (2017), http://dx.doi.org/10.1016/j.pnucene.2017.03.023

$$F^*\Psi^* = \bar{\nu}\sum_f(r,E)\int_0^\infty\int_0^{4\pi}\frac{\chi(r,E')}{4\pi}\Psi^*(r,\Omega',E')d\Omega'dE'. \quad (12)$$

For deterministic codes, the adjoint flux Ψ^* can be computed by solving Eq. (10) explicitly after which the adjoint kinetic parameters can be calculated based on Eqs. ((1)–(2)).

2.2. IFP method

To calculate adjoint-weighted parameters with a higher fidelity, the IFP method (Nauchi and Kameyama, 2010; Kiedrowski and Brown, 2009) is used to estimate adjoint flux for the Monte Carlo transport calculations.

The IFP method interprets the adjoint flux $\phi^*(r,\Omega,E)$ as the asymptotic increase of fission rates caused by a neutron with energy E flying towards Ω which is introduced at position r (Hurwitz, 1964). According to this physical meaning, adjoint flux can be computed directly in the forward transport calculations by dividing the active cycles into blocks. Each block contains an original cycle, sufficient latent cycles and an asymptotic cycle. Numerical experiments show that a block size of ten can produce a convergent adjoint flux at the asymptotic cycle for most problems. And the adjoint-weighted tally (Kiedrowski et al., 2011) method is put forward to compute adjoint-weighted reaction rates in the form of Eqs. ((5)–(7)).

To be more specific, in the original cycle, every fission neutron produced by the source neutron is taken as a progenitor and concerning reaction rates are scored between the position of the source neutron and where every progenitor was born. An ID number is assigned to every progenitor and then inherited by all corresponding progeny neutrons in the latent cycles. In the asymptotic cycle, fission production rates of all progeny neutrons having the same ID number are collected to obtain the importance of progenitor neutrons, which are used to weight the corresponding reaction rates in the original cycle. Finally, the adjoint-weighted reaction rates can be computed in every block based on the estimator

$$\widehat{R} = \sum_m T_m I_m, \quad (13)$$

where

$\widehat{R}$ is a score of concerning adjoint-weighted reaction rates at any block,
m is the index of progenitor in the original cycle,
T_m is the reaction rates produced by the progenitor m in the original cycle and
I_m is the iterated fission probability of the progenitor m, which can be computed in the asymptotic cycle by summing up the

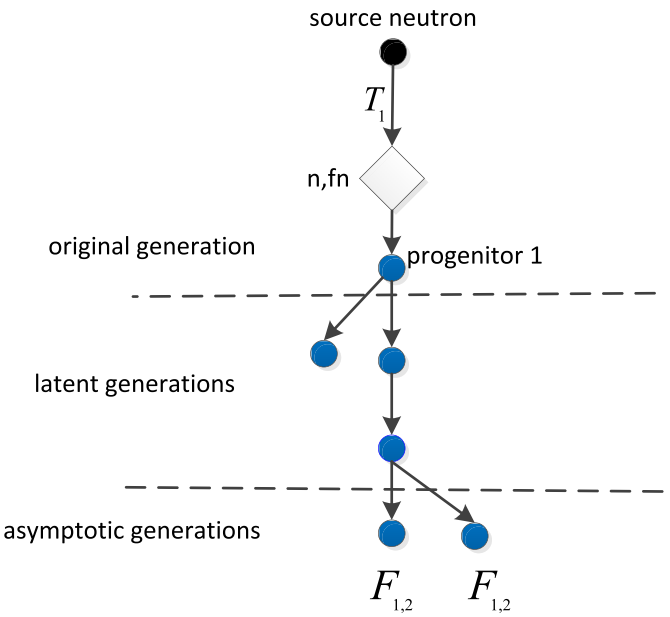

Fig. 2. Schematic diagram of adjoint-weighted tally approach for the superhistory method.

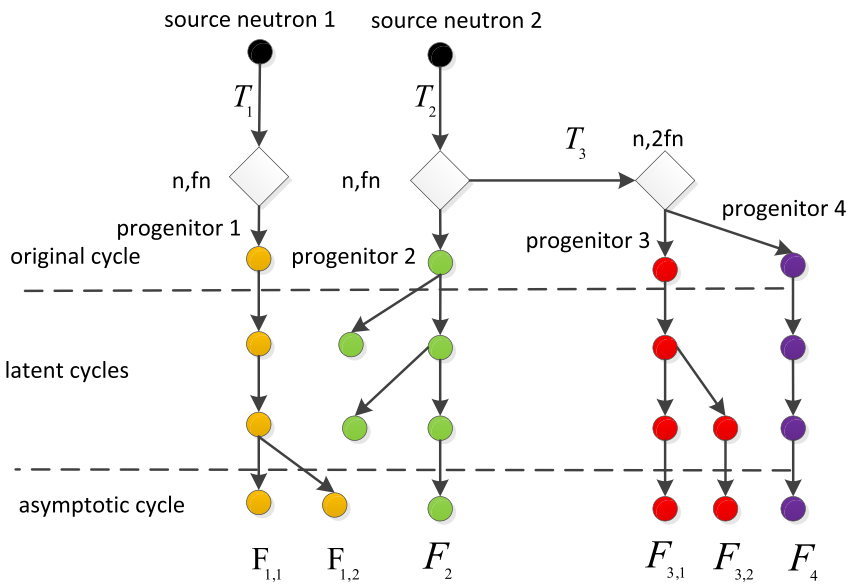

Fig. 1. Schematic diagram of adjoint-weighted tally approach for the power method.

Please cite this article in press as: Qiu, Y., et al., Calculation of adjoint-weighted kinetic parameters with the Reactor Monte Carlo code RMC, Progress in Nuclear Energy (2017), http://dx.doi.org/10.1016/j.pnucene.2017.03.023

fission production rates of all progeny neutrons produced by the progenitor m, in the form of

$$I_m = \sum_{n\in m} F_n, \tag{14}$$

where

n is the index of progeny neutron in the asymptotic cycle, and
F_n is the fission production rates of progeny neutron n in the asymptotic cycle, which can be scored by using the track length estimator, in the form of

$$F_n = \sum_{\tau\in n} \left(\bar{\nu}\Sigma_f\right)_\tau w_\tau l_\tau, \tag{15}$$

where

τ is the track index of neutron n,
l_τ is the length of track τ,
w_τ is the neutron weight at track τ and
$(\bar{\nu}\Sigma_f)_\tau$ is mean neutron emission times macroscopic fission cross section.

Take Fig. 1 as an example. The diamond mark "n, fn" and "n, 2fn" in Fig. 1 mean one fission neutron induced by one neutron and two fission neutrons induced by one neutron respectively. It should be noted that the implicit capture technique is used in this work. In this case, instead of killing a neutron immediately after it is captured, its weight is reduced. Therefore, the source neutron 2 in Fig. 2 may have a "n, 2fn" after a "n, fn". In the original cycle, two source neutrons produce four progenitors in total, and the reaction rates of these progenitors are T_1, T_2, T_2+T_3 and T_2+T_3 respectively. And the iterated fission probability of these progenitor are $F_{1,1}+F_{1,2}$, F_2, $F_{3,1}+F_{3,2}$ and F_4 respectively. Therefore, the score of adjoint-weighted reaction rates is $T_1\times(F_{1,1}+F_{1,2})+T_2\times F_2+(T_2+T_3)\times(F_{3,1}+F_{3,2})+(T_2+T_3)\times F_4$.

Eq. (13) provides a general approach to obtain a score for adjoint-weighted reaction rates. Now consider the specific scores for Eqs. ((5)–(7)). The track length estimator is used to estimate $\langle\Psi^*,\frac{1}{v}\Psi\rangle$, in the form of

$$T_{m,1} = \sum_{d\in m}\frac{1}{v_d}w_{0,m}l_d, \tag{16}$$

where

d is the index of track for neutron m,
l_d is the length of track d,
v_d is the speed of neutron at track d, and
$w_{0,m}$ is the initial weight of neutron m.

And a collision estimator is used to estimate $\langle\Psi^*,F^D\Psi\rangle$ and $\langle\Psi^*,F\Psi\rangle$ in the form of

$$T_{m,2} = \sum_m \delta_{m,D}w_{0,m}, \tag{17}$$

and

$$T_{m,3} = \sum_m w_{0,m}, \tag{18}$$

where $\delta_{m,D}$ = one if the progenitor number m is a delayed fission neutron and zero otherwise.

It should be noted that in Eqs. ((16)–(18)), the initial neutron weight instead of the current weight of each track is used since each progenitor is required to start with an equal weight.

In this way, one can obtain a score of the adjoint-weighted kinetic parameters in Eqs. ((1)–(2)) at each block, in the form of

$$\Lambda_{eff} = \frac{\sum_m \sum_{d\in m}\frac{1}{v_d}w_{0,m}l_d \sum_{n\in m}\sum_{\tau\in n}\left(\bar{\nu}\Sigma_f\right)_\tau w_\tau l_\tau}{\sum_m w_{0,m}\sum_{n\in m}\sum_{\tau\in n}\left(\bar{\nu}\Sigma_f\right)_\tau w_\tau l_\tau}, \tag{19}$$

and

$$\beta_{eff} = \frac{\sum_m \delta_{m,D}w_{0,m}\sum_{n\in m}\sum_{\tau\in n}\left(\bar{\nu}\Sigma_f\right)_\tau w_\tau l_\tau}{\sum_m w_{0,m}\sum_{n\in m}\sum_{\tau\in n}\left(\bar{\nu}\Sigma_f\right)_\tau w_\tau l_\tau}. \tag{20}$$

3. Algorithms of the IFP method

3.1. Non-Overlapping Blocks algorithm

The representative implementation of the Non-Overlapping Blocks algorithm is MCNP5 (Kiedrowski et al., 2011), which divides the active cycles into contiguous blocks. In this algorithm, each cycle belongs to one block uniquely.

The NOB algorithm can be expressed in the form of

$$\overline{R} = \frac{1}{B_1}\sum_{i=1}^{B_1}\sum_m T_m^{b(i-1)+1}I_m^{bi}, \tag{21}$$

where

i is the block index,
b is the number of cycles in each block or the block size,
B_1 is the total number of blocks for the NOB algorithm,
$T_m^{b(i-1)+1}$ is the reaction rates produced by progenitor m, which is stored in the cycle $b(i-1)+1$, i.e., the original cycle in block i and
I_m^{bi} is the iterated fission probability produced by progenitor m, which is computed in the cycle bi, i.e., the asymptotic cycle in block i.

And the total number of blocks of the NOB algorithm is

$$B_1 = \left[\frac{c}{b}\right], \tag{22}$$

where [] means to reserve the integer part of the inside value and c is the total number of active cycles.

Thus, in the NOB algorithm, the adjoint-weighted tally can only be computed in specific cycles, i.e., the asymptotic cycles in all blocks.

3.2. Multiple overlapping blocks algorithm

The representative implementations of the Multiple Overlapping Blocks algorithm are SERPENT 2 (Leppänen et al., 2014) as well as RMC (Qiu et al., 2015). In this algorithm, there are multiple overlapping blocks in each cycle and each active cycle starts with a new block. Therefore, cycle i is taken as the original cycle of block i, the latent cycle of blocks $i-b+2$ to $i-1$ as well as the asymptotic cycle of the block $i-b+1$.

The MOB algorithm can be expressed in the form of

Please cite this article in press as: Qiu, Y., et al., Calculation of adjoint-weighted kinetic parameters with the Reactor Monte Carlo code RMC, Progress in Nuclear Energy (2017), http://dx.doi.org/10.1016/j.pnucene.2017.03.023

$$\overline{R} = \frac{1}{B_2}\sum_{j=1}^{B_2}\sum_{m} T_m^j I_m^{j+b-1}, \tag{23}$$

where

j is the cycle index and also the block index for the MOB algorithm,
B_2 is the total number of blocks for the MOB algorithm,
T_m^j is the reaction rates produced by progenitor m, which is stored in the cycle j, and
I_m^{j+b-1} is the iterated fission probability produced by progenitor m, which is computed in the cycle $j+b-1$.

And the total number of blocks of the MOB algorithm is

$$B_2 = c - b + 1. \tag{24}$$

Therefore, in the MOB algorithm, the adjoint-weighted tally can be computed in $c-b+1$ cycles totally. Therefore, the MOB algorithm can achieve lower uncertainties of results than the NOB algorithm by a factor of $\sqrt{\frac{\left[\frac{c}{b}\right]}{c-b+1}}$. However, it should be noted that the memory consumption of the MOB algorithm will also increase by a factor of a block size compared to the NOB algorithm. In addition, another concern is that the MOB algorithm may cause underestimation of uncertainties due to inter-cycle correlation of the tally scores. Therefore, the differences in estimation of uncertainties among different algorithms were studied by numerical tests in this work.

3.3. Superhistory algorithm

The IFP method needs to store the concerning reaction rates for all progenitors in the original cycle until the end of the asymptotic cycle. Consequently, the memory consumption of the IFP method is proportional to the number of progenitors in the original cycle which is roughly equal to the number of source neutrons per cycle. This may produce considerable memory consumption if the number of concerning adjoint-weighted reaction rates is large. Therefore, the superhistory algorithm (Brissenden and Garlick, 1986) is adopted by RMC to reduce memory consumption of computing adjoint-weighted parameters. As opposed to the NOB algorithm and the MOB algorithm which are implemented in the power method, the superhistory algorithm of the IFP method is based on the superhistory method (Brissenden and Garlick, 1986; Blomquist and Gelbard, 2002; She et al., 2012). The superhistory method has the advantage of reducing inter-cycle correlation and biases of results because it reduces the frequency of source renormalization (Brissenden and Garlick, 1986).

In the superhistory method, the fission neutrons of each source neutron are tracked through a specific number of generations,[1] namely, a supergeneration (Blomquist and Gelbard, 2002), instead of only one generation before they are placed into the fission bank for the next cycle. The history of the source neutron and all of its fission neutrons within a supergeneration is called a superhistory. Practically, superhistories are simulated one by one. Therefore, a supergeneration can be regarded as a block just like the Non-Overlapping Blocks algorithm. The first generation in a supergeneration can be taken as the original generation while the last generation in a supergeneration can be regarded as the asymptotic generation. As a result, a score of the adjoint-weighted tally can be obtained in every superhistory, in the form of

$$\widehat{R} = \sum_{m} T_m^1 I_m^s, \tag{25}$$

where $\widehat{R}$ is a score of concerning adjoint-weighted reaction rates at any block (in the superhistory algorithm, every supergeneration can be taken as a block),

m is the index of progenitor in the first generation of a supergeneration,
T_m^1 is the reaction rates produced by progenitor m, which is stored in the first generation of a supergeneration,
I_m^s is the iterated fission probability produced by progenitor m, which is computed in the last generation of a supergeneration,
s is the number of generations in a supergeneration, which can be set to be the same value as the block size b of the Non-Overlapping Blocks algorithm and Multiple Overlapping Blocks algorithm.

Take Fig. 2 as an example. In the original generation, the source neutron has produced a progenitor whose reaction rates is T_1. And the iterated fission probability of this progenitor is $F_{1,1}+F_{1,2}$, which can be computed in the last generation of the supergeneration. Therefore, a score of the adjoint-weighted reaction rates in this superhistory is $T_1 \times (F_{1,1}+F_{1,2})$.

And the average score of the adjoint-weighted tally for the superhistory algorithm is

$$\overline{R} = \frac{1}{hc}\sum_{i=1}^{c}\sum_{j=1}^{h}\sum_{m\in j} T_m^{1,i,j} I_m^{s,i,j}, \tag{26}$$

where

i is the cycle index,
j is the index of particle history in cycle i,
h is the total number of particle superhistories in cycle i,
m is the index of progenitor in the first generation within superhistory j in cycle i,
$T_m^{1,i,j}$ is the reaction rates produced by progenitor m and is stored in the first generation within superhistory j in cycle i, and
$I_m^{s,i,j}$ is the iterated fission probability produced by progenitor m and is computed in the last generation within superhistory j in cycle i.

Instead of being proportional to the number of particle histories in every cycle, the memory consumption of the superhistory algorithm depends on the number of progenitors produced in the first generation of each supergeneration, which is roughly equal to one. Furthermore, since every superhistory may contribute to the concerning adjoint-weighted tally, it can be inferred that the superhistory algorithm can achieve the same uncertainties as the Non-Overlapping Blocks algorithm, as long as the number of generations in a supergeneration is set to be equal to the block size in the Non-Overlapping Blocks algorithm and both algorithms simulate the same number of effective particle histories.

[1] It should be noted that the superhistory method is also a powering strategy. Therefore, interpretation of "cycle", an iteration for updating the k_{eff} and the fission source distribution, should be suitable for the conventional method as well as the superhistory method. For the conventional power method, since the fission bank for the next cycle is made of the 1st generation fission neutrons of the source neutrons, "cycle" and "generation" have the same meaning. However, for the superhistory method, a "cycle" is a "supergeneration", which has L generations (L being some fixed input integer).

Please cite this article in press as: Qiu, Y., et al., Calculation of adjoint-weighted kinetic parameters with the Reactor Monte Carlo code RMC, Progress in Nuclear Energy (2017), http://dx.doi.org/10.1016/j.pnucene.2017.03.023

4. Verification and results

In this work, three algorithms based on the IFP method, namely, the Non-Overlapping Blocks (NOB) algorithm, the Multiple Overlapping Blocks (MOB) algorithm, the superhistory algorithm, have been implemented in the Reactor Monte Carlo (RMC) code to compute adjoint-weighted kinetic parameters. While the NOB and MOB algorithms are implemented in the power method, the superhistory algorithm is implemented in the superhistory method. The new capability in RMC has been verified and validated by comparison to MCNP6 (Goorley et al., 2013) and experimental results through a set of multi-group problems and continuous-energy problems (Kiedrowski, 2010; Mosteller and Kiedrowski, 2011) covering a wide range of uranium, plutonium and U-233 fueled experiments from fast to thermal spectrum, which are summarized in Table 1.

All calculations were performed on the Tsinghua High Performance Computational Platform, the Inspur TS10000 HPC Server having 740 nodes total. Each node is composed of 12 CPUS (Intel Xeon X5670 at 2.93 GHz) sharing 32 Gb or 48 Gb memory. The MCNP6 version in this work utilizes shared-memory platform, OpenMP, while RMC uses the Message Passing Interface (MPI) for parallelism.

As discussed above, the adjoint-weighted kinetic parameters could only be computed at the asymptotic generations. In order to investigate how many generations are sufficient to establish a stable population of fission neutrons, the effective delayed neutron fractions and neutron generation times with different block sizes for the FLAT23 problem, with ID 13 in Table 1, are computed and presented in Fig. 3 and Fig. 4 respectively. As can be seen, a block size of 5 is enough for the FLAT23 problem. And numerical tests

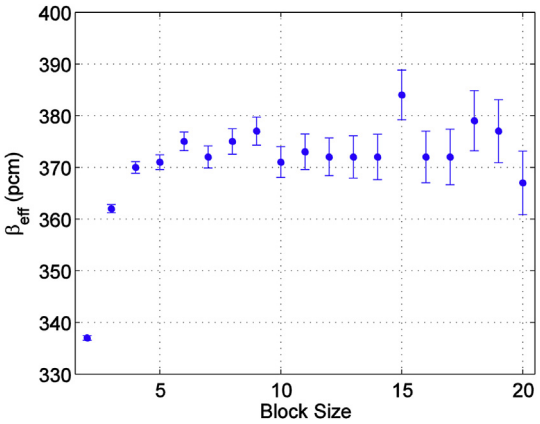

Fig. 3. Effective delayed neutron fractions with different block sizes for the FLAT23 problem.

Table 1
Description of test problems.

ID	Problems	Energy structure	Short description
1	ONEINF	One group	Infinite medium problem
2	THRESLAB	Two group	A three region slab which is composed of metallic core, thermal neutron shield and moderator
3	TWOINF	Four group	Infinite medium problem
4	BARESLAB	Four group	Bare fast slab
5	BARESPHR	Four group	Bare fast slab
6	REFLSLAB	Four group	Metallic slab surrounded by moderating material
7	REFLSPHR	Four group	Metallic sphere and reflector
8	SUBCSLAB	Four group	Subcritical bare fast slab
9	SUPCSLAB	Four group	Supercritical bare fast slab
10	INTRSLAB	Eight group	Intermediate spectrum bare slab
11	BIG TEN (1)	Continuous-Energy	Large all-uranium-metal cylindrical core surrounded by a thick reflector of natural uranium
12	BIG TEN (2)	Continuous-Energy	Four intermediate-enriched uranium cylinders surrounded by annuli of normal uranium or a homogeneous mixture of highly enriched uranium and normal uranium and enclosed by a depleted-uranium reflector
13	FLAT23	Continuous-Energy	An inner sphere of U-233 enclosed in an annulus of normal uranium
14	Flattop-25	Continuous-Energy	Solid, homogeneous sphere of highly enriched uranium reflected by an annulus of normal uranium
15	Flattop-Pu	Continuous-Energy	Sphere of delta-phase plutonium reflected by an annulus of normal uranium
16	GODIVA	Continuous-Energy	Bare, homogeneous sphere of highly enriched uranium
17	JEZPU	Continuous-Energy	Bare sphere of plutonium
18	Jezebel-233	Continuous-Energy	Bare, homogeneous sphere of U-233
19	STACY29	Continuous-Energy	Water-reflected cylindrical tank with uranyl nitrate solution
20	STACY-30	Continuous-Energy	Unreflected cylindrical tank with uranyl nitrate solution
21	STACY-46	Continuous-Energy	Water-reflected cylindrical tank with uranyl nitrate solution
22	THOR	Continuous-Energy	Sphere of plutonium enclosed in the center of a right circular cylinder of thorium
23	Zeus-1	Continuous-Energy	Cylindrical highly enriched uranium platters separated by graphite platters and enclosed in a copper reflector
24	Zeus-4	Continuous-Energy	Cylindrical highly enriched uranium platters separated by graphite platters and enclosed in a copper reflector
25	Zeus-6	Continuous-Energy	Plates of highly enriched uranium reflected by copper

Please cite this article in press as: Qiu, Y., et al., Calculation of adjoint-weighted kinetic parameters with the Reactor Monte Carlo code RMC, Progress in Nuclear Energy (2017), http://dx.doi.org/10.1016/j.pnucene.2017.03.023

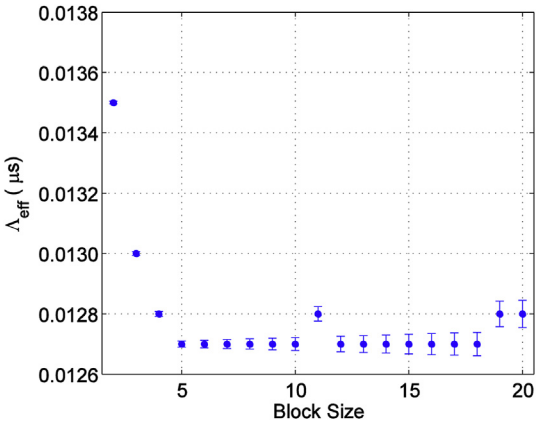

Fig. 4. Effective neutron generation times with different block sizes for the FLAT23 problem.

suggest that a block size of ten is sufficient to obtain accurate estimate of the asymptotic population for most cases. Therefore, a block size of ten is applied for all the calculations with MCNP6 and RMC in this work.

All MCNP6, the NOB algorithm, MOB algorithm calculations were run using 100 inactive and 50,000 active cycles of 10,000 source neutrons while those of the superhistory algorithm were performed with 10,000 superhistories × 100 inactive cycles and 5000 active cycles since every supergeneration is set to contain ten generations. Both MCNP6 and RMC use the ENDF/B-VII.0 based cross section library. And all calculations were performed with 12 processors in parallel.

Table 2 lists the effective multiplication factors computed by the power method of MCNP6, the power method of RMC and the superhistory method of RMC. And the standard deviations (Std) and the differences (Diff) in Table 2 are given in pcm, i.e., 10^{-5}. It should be noted that the relative difference, i.e., $\Delta k/k$, is traditionally used to quantify the difference of metrics measured in pcm. However, Δk is used to show the absolute difference in this work. Therefore, the differences in Table 2 are defined as the results of RMC minus the corresponding results of MCNP6. As can be seen, except the twenty-third case where the differences are about ten standard deviations (but still within 60 pcm), the differences in k_{eff} between MCNP6 and either method of RMC, i.e., the power method or the superhistory method, are within three standard deviations.

Table 2
Comparison of effective multiplication factors between MCNP6 and RMC.

ID	MCNP6		RMC-superhistory method			RMC-power method			EF
	k_{eff}	Std	k_{eff}	Std	Diff	k_{eff}	Std	Diff	
1	1.00003	1	0.99998	1.5	−5[a]	1.00003	1.5	−1	1.00
2	1.00043	3	1.00041	2.5	−2	1.00047	2.5	4	1.00
3	1.00001	3	1.00000	3.0	−1	1.00008	3.1	7	1.03
4	0.99998	3	1.00002	3.0	4	1.00000	3.0	2	1.00
5	0.99999	3	0.99993	2.8	−7	0.99995	2.6	−4	0.93
6	1.00012	4	1.00012	4.2	0	1.00016	4.1	4	0.98
7	1.00080	3	1.00081	2.8	1	1.00085	2.7	5	0.96
8	0.77966	3	0.77971	2.8	5	0.77974	2.7	8	0.96
9	1.14007	3	1.14010	3.1	3	1.14012	3.1	5	1.00
10	1.00036	3	1.00039	2.9	3	1.00044	2.9	8	1.00
11	0.99498	2	0.99499	2.7	1	0.99498	2.3	0	0.85
12	1.00491	2	1.00481	2.7	−10	1.00483	2.3	−8	0.85
13	0.99943	3	0.99925	3.6	−18	0.99932	3.1	−11	0.86
14	1.00290	3	1.00289	3.3	−1	1.00290	2.9	0	0.88
15	1.00009	3	1.00010	3.6	1	1.00009	3.1	0	0.86
16	0.99983	3	0.99990	3.1	7	0.99980	2.7	−3	0.87
17	0.99993	3	0.99986	2.9	−7	0.99989	2.6	−4	0.90
18	0.99962	3	0.99961	2.9	−1	0.99958	2.5	−4	0.86
19	1.00191	3	1.00196	2.9	5	1.00199	2.7	8	0.93
20	1.00241	3	1.00242	2.7	1	1.00244	2.4	3	0.89
21	0.99826	3	0.99826	3.1	−1	0.99832	2.8	6	0.90
22	0.99802	3	0.99799	3.0	−3	0.99801	2.8	−1	0.93
23	0.99362	3	0.99306	3.6	−56	0.99304	3.4	−58	0.94
24	1.00732	3	1.00736	3.5	4	1.00735	3.3	3	0.94
25	1.00882	3	1.00880	3.1	−2	1.00884	2.9	2	0.94

[a] It should be noted that the standard deviations (Std) and the differences (Diff) are given in pcm, i.e., 10^{-5}. Therefore, a difference of 5 should be read as 5×10^{-5}.

Among the different methods, one of the main concerns is the level of underestimation of variance. Therefore, in order to investigate the underestimation of variance, a term, Estimation Factor (EF) is defined. As it is known, the relationship between the relative uncertainties of a tally score, R, and the effective particle histories contributing to the tally score, N, is

$$R^2 = \frac{1}{f_0 N}, \tag{27}$$

where f_0 is some constant for a specific problem with a specific method. Because the superhistory method reduces the frequency of fission source normalization (Brissenden and Garlick, 1986), the apparent variance is closer to the real variance and it may have smaller f_0 than the power method. Assuming the variance of the superhistory method is the reference value for the real variance, the estimation factor is defined as

$$\frac{f_{0,s}}{f_{0,p}} = \frac{N_p R_p^2}{N_s R_s^2}, \tag{28}$$

where the subscript index s represents the superhistory method and the subscript index p represents the power method.

Therefore, the estimation factor in Eq. (28) describes the level of underestimation of variance of the power method. Since a supergeneration is set to have ten generations, the number of effective particle histories contributing to tally scores of k_{eff} for the superhistory method, is the same as that for the power method, i.e.,

Table 3
Computational time (minutes) for the test problems.

ID	superhistory	NOB	MOB
1	1.9	2.0	3.2
2	200.1	211.9	256.5
3	0.9	1.0	1.5
4	12.2	10.3	12.7
5	0.6	0.8	1.3
6	311.8	258.4	305.4
7	0.9	1.0	2.3
8	8.5	9.4	10.4
9	11.9	11.5	14.3
10	10.8	9.0	11.3
11	76.5	72.3	89.1
12	83.9	79.8	99.9
13	32.4	31.3	32.4
14	35.5	35.0	35.8
15	41.0	39.7	41.1
16	5.8	4.3	6.1
17	2.9	2.9	3.5
18	2.9	3.0	3.6
19	119.8	115.2	131.3
20	117.4	112.0	120.4
21	78.2	75.5	80.0
22	21.6	21.7	22.2
23	71.7	67.9	88.3
24	78.8	148.6	158.5
25	213.5	197.5	214.6

Please cite this article in press as: Qiu, Y., et al., Calculation of adjoint-weighted kinetic parameters with the Reactor Monte Carlo code RMC, Progress in Nuclear Energy (2017), http://dx.doi.org/10.1016/j.pnucene.2017.03.023

$N_p = N_s$. As shown in Table 2, the average EF is 0.93, which means the variance of k_{eff} obtained by the power method in RMC is underestimated by a factor of 0.93 compared to the superhistory method in RMC.

Table 3 lists the computational time of different algorithms in RMC for the test problems. The superhistory algorithm consumes the same time as the NOB algorithm on average while the MOB algorithm increases the run time by a factor of 30% on average.

Table 4 presents the effective delayed neutron fractions computed by MCNP6 and the different algorithms in RMC and their differences. The MCNP6 results are regarded as the reference and the differences are defined as the RMC results minus by the corresponding MCNP6 results. All values and differences in Table 4 and the standard deviations in Table 5 are given in pcm. The differences in the effective delayed neutron fractions between MCNP6 and any of the three algorithms of RMC, i.e., the superhistory algorithm, the NOB algorithm and the MOB algorithm, are within three standard deviations.

Similar to Table 2, the estimation factors of effective delayed neutron fractions are given in Table 5, where the variances obtained from the superhistory method are taken to be the reference values for the real variances. For convenience, the variance are the sum of squares computed for the ratio, i.e., the left hand side of Eq. (20) and they are not computed for the numerator and denominator separately along with the correlations, although the latter is a more statistically sound method for computing them. It should be noted that the number of effective particles contributing to scores of β_{eff} for both the NOB algorithm and the superhistory algorithm are 10,000 × 5000 whilst that for the MOB algorithm is 10,000 × (50000-9). As shown in Table 5, the average EF of both the NOB and MOB algorithms are 1.00, which means both algorithms avoid underestimation of the variances of effective delayed neutron fractions compared to the superhistory algorithm.

Moreover, in order to compare efficiencies of different algorithms in RMC, a term, Figure of Merit (FOM), is introduced, which is defined as

$$\frac{1}{R^2 T}, \tag{29}$$

where T is the computational time which is defined as the run time multiplied by the parallel processors (i.e., 12) in this work. As shown in Table 6, the ratio of the FOMs of the NOB algorithm to those of the superhistory algorithm is 1.01 on average while the ratio of the FOMs of the MOB algorithm to those of the superhistory algorithm is 8.92 on average, indicating in terms of computing effective delayed neutron fractions, the efficiency of the NOB algorithm is almost the same as that of the superhistory algorithm while the MOB algorithm is more efficient than the other two algorithms by a factor of 8.92.

Table 7 displays the effective neutron generation times computed by MCNP6 and the different algorithms of RMC and their differences. All the effective neutron generation times, the differences in Table 7 and the standard deviations in Table 8 are given in μs. Again, the MCNP6 results are considered to be the reference results. Except the twenty-forth and twenty-fifth cases where the differences are about ten standard deviations, the differences in the effective neutron generation times between MCNP6 and any of the three algorithms of RMC, i.e., the superhistory algorithm, the NOB algorithm and the MOB algorithm, are within three standard deviations.

The estimation factors of effective neutron generation times are also given in Table 8, where the variances obtained from the superhistory method are taken to be the reference values for the real variances. For convenience, the variance are the sum of squares computed for the ratio, i.e., the left hand side of Eq. (19) and they are not computed for the numerator and denominator separately along with the correlations, although the latter is a more statistically sound method for computing them. The average EF of the NOB algorithm is 0.99 while that of the MOB algorithm is 1.01, which suggests both algorithms nearly avoid underestimation of the variances of effective neutron generation times compared to the superhistory algorithm. It should be noted that in some cases the

Table 4
Comparison of effective delayed neutron fractions between MCNP6 and RMC.

ID	MCNP6	superhistory		NOB		MOB	
	β_{eff}	β_{eff}	Diff	β_{eff}	Diff	β_{eff}	Diff
11	723	722	−1	723	0	722	−1
12	716	719	3	722	6	725	9
13	372	375	3	371	−1	372	0
14	686	686	0	690	4	689	3
15	279	278	−1	274	−5	278	−1
16	644	652	8	654	10	650	6
17	187	187	0	187	0	183	−4
18	291	296	5	296	5	295	4
19	732	735	3	728	−4	728	−4
20	716	718	2	713	−3	713	−3
21	740	732	−8	734	−6	735	−5
22	210	209	−1	210	0	209	−1
23	741	735	−6	741	0	738	−3
24	724	729	5	728	4	730	6
25	687	685	−2	679	−8	684	−3

Table 5
Standard deviations and estimation factor of effective delayed neutron fractions.

ID	MCNP	superhistory	NOB		MOB	
	Std	Std	Std	EF	Std	EF
11	4	3.9	4.0	1.02	1.2	1.00
12	4	3.8	3.9	1.02	1.2	1.02
13	3	3.1	3.0	0.99	1.0	1.00
14	4	4.1	4.2	1.01	1.3	1.00
15	3	2.7	2.7	1.01	0.8	0.99
16	4	4.0	4.0	1.01	1.3	1.01
17	2	2.1	2.1	1.02	0.6	1.01
18	3	2.6	2.6	1.00	0.8	1.00
19	6	3.4	3.4	1.03	1.1	1.02
20	5	3.2	3.2	1.01	1.0	1.00
21	6	3.5	3.4	0.98	1.1	0.99
22	2	2.2	2.1	0.97	0.7	0.99
23	4	4.1	4.2	1.01	1.3	1.00
24	4	4.2	4.3	1.02	1.3	1.01
25	4	4.1	3.9	0.98	1.3	0.98

Table 6
Figure of merits of effective delayed neutron fractions.

ID	superhistory	NOB	MOB
11	3.69E+01	3.77E+01	3.15E+02
12	3.54E+01	3.55E+01	2.85E+02
13	3.82E+01	4.08E+01	3.83E+02
14	6.51E+01	6.48E+01	6.44E+02
15	2.20E+01	2.25E+01	2.24E+02
16	3.86E+02	5.06E+02	3.63E+03
17	2.39E+02	2.30E+02	1.91E+03
18	3.72E+02	3.66E+02	2.99E+03
19	3.31E+01	3.23E+01	2.91E+02
20	3.49E+01	3.60E+01	3.38E+02
21	4.70E+01	5.06E+01	4.69E+02
22	3.52E+01	3.69E+01	3.46E+02
23	3.67E+01	3.82E+01	3.00E+02
24	3.22E+01	1.63E+01	1.57E+02
25	1.12E+01	1.25E+01	1.16E+02

Please cite this article in press as: Qiu, Y., et al., Calculation of adjoint-weighted kinetic parameters with the Reactor Monte Carlo code RMC, Progress in Nuclear Energy (2017), http://dx.doi.org/10.1016/j.pnucene.2017.03.023

Table 7
Comparison of effective neutron generation times between MCNP6 and RMC.

ID	MCNP6	superhistory		NOB		MOB	
		Λ_{eff}	Diff	Λ_{eff}	Diff	Λ_{eff}	Diff
1	1.00E-02	1.00E-02	−2.85E-06	1.00E-02	−2.02E-06	1.00E-02	2.80E-07
2	4.93E-02	4.92E-02	−8.80E-05	4.92E-02	−1.16E-04	4.92E-02	−7.80E-05
3	1.42E-02	1.42E-02	8.76E-06	1.42E-02	4.26E-06	1.42E-02	5.66E-06
4	9.79E-03	9.79E-03	2.91E-06	9.79E-03	−4.22E-06	9.79E-03	1.33E-06
5	1.72E-03	1.72E-03	−2.52E-06	1.72E-03	−6.80E-07	1.72E-03	−1.15E-06
6	1.35E+02	1.35E+02	1.05E-01	1.35E+02	1.05E-01	1.35E+02	5.75E-02
7	1.02E-02	1.02E-02	−1.41E-05	1.02E-02	−1.89E-05	1.02E-02	−5.47E-06
8	1.02E-02	1.02E-02	−1.81E-06	1.02E-02	−1.51E-06	1.02E-02	−6.10E-07
9	9.67E-03	9.68E-03	8.48E-06	9.68E-03	7.48E-06	9.67E-03	−1.00E-07
10	1.13E-01	1.12E-01	−5.46E-04	1.12E-01	−4.83E-04	1.12E-01	−5.71E-04
11	6.25E-02	6.25E-02	−1.60E-05	6.24E-02	−4.24E-05	6.24E-02	−5.83E-05
12	6.15E-02	6.15E-02	1.10E-06	6.15E-02	8.16E-05	6.15E-02	8.65E-05
13	1.27E-02	1.27E-02	−4.38E-05	1.27E-02	−3.43E-05	1.27E-02	−3.01E-05
14	1.74E-02	1.74E-02	−2.25E-05	1.74E-02	−1.16E-05	1.74E-02	−2.24E-05
15	1.33E-02	1.33E-02	−2.27E-05	1.33E-02	−2.07E-05	1.33E-02	−2.47E-05
16	5.70E-03	5.70E-03	−5.40E-07	5.70E-03	2.24E-06	5.69E-03	−5.35E-06
17	2.88E-03	2.87E-03	−1.85E-06	2.87E-03	−2.31E-06	2.87E-03	−3.43E-06
18	2.75E-03	2.75E-03	5.77E-06	2.75E-03	2.01E-06	2.75E-03	2.96E-06
19	5.97E+01	5.97E+01	2.45E-02	5.97E+01	−1.21E-02	5.97E+01	3.31E-02
20	6.72E+01	6.72E+01	−9.90E-03	6.72E+01	−3.12E-02	6.72E+01	3.70E-03
21	5.86E+01	5.86E+01	−6.74E-03	5.86E+01	−4.65E-02	5.86E+01	−3.84E-02
22	9.99E-03	1.00E-02	8.26E-06	1.00E-02	2.50E-05	1.00E-02	2.92E-05
23	2.04E+00	2.03E+00	−9.30E-03	2.03E+00	−1.21E-02	2.03E+00	−7.77E-03
24	2.26E-01	2.21E-01	−4.71E-03	2.22E-01	−4.08E-03	2.22E-01	−4.41E-03
25	1.67E-01	1.64E-01	−2.82E-03	1.64E-01	−2.67E-03	1.64E-01	−2.86E-03

Table 8
Standard deviations and estimation factors of effective neutron generation times.

ID	MCNP6	superhistory	NOB		MOB	
	Std	Std	Std	EF	Std	EF
1	8.50E-07[a]	4.30E-06	4.34E-06	1.01	1.37E-06	1.01
2	1.02E-04	4.32E-05	3.83E-05	0.89	1.83E-05	1.34
3	5.25E-06	5.03E-06	5.14E-06	1.02	1.66E-06	1.04
4	5.94E-06	5.92E-06	5.98E-06	1.01	1.89E-06	1.01
5	1.02E-06	1.03E-06	1.02E-06	0.99	3.22E-07	0.99
6	1.07E-01	1.03E-01	1.07E-01	1.03	3.40E-02	1.04
7	7.37E-06	7.36E-06	7.34E-06	1.00	2.33E-06	1.00
8	7.30E-06	7.33E-06	7.23E-06	0.99	2.31E-06	1.00
9	5.26E-06	5.28E-06	5.27E-06	1.00	1.67E-06	1.00
10	4.40E-04	4.41E-04	4.27E-04	0.97	1.37E-04	0.98
11	5.10E-05	4.90E-05	4.89E-05	1.00	1.54E-05	1.00
12	4.99E-05	4.84E-05	4.73E-05	0.98	1.52E-05	0.99
13	2.15E-05	2.14E-05	2.15E-05	1.00	6.79E-06	1.00
14	2.39E-05	2.35E-05	2.39E-05	1.01	7.48E-06	1.01
15	2.22E-05	2.20E-05	2.18E-05	0.99	6.93E-06	1.00
16	4.72E-06	4.69E-06	4.73E-06	1.01	1.49E-06	1.01
17	2.52E-06	2.54E-06	2.48E-06	0.98	7.94E-07	0.99
18	2.24E-06	2.25E-06	2.26E-06	1.01	7.06E-07	0.99
19	4.02E-02	2.55E-02	2.53E-02	0.99	8.01E-03	0.99
20	4.18E-02	2.70E-02	2.67E-02	0.99	8.49E-03	0.99
21	3.79E-02	2.45E-02	2.44E-02	1.00	7.63E-03	0.99
22	1.77E-05	1.80E-05	1.78E-05	0.99	5.61E-06	0.98
23	3.67E-03	3.70E-03	3.68E-03	0.99	1.17E-03	1.00
24	3.01E-04	2.90E-04	2.94E-04	1.01	9.27E-05	1.01
25	1.85E-04	1.82E-04	1.80E-04	0.99	5.70E-05	0.99

[a] It should be noted that the standard deviations (Std) are given in μs. Therefore, a Std. of 8.50E-07 should be read as 8.50×10^{-13} s.

Table 9
Figure of merits of effective neutron generation times.

ID	superhistory	NOB	MOB
1	2.36E+05	2.24E+05	1.36E+06
2	5.41E+02	6.48E+02	2.36E+03
3	7.54E+05	6.40E+05	4.03E+06
4	1.87E+04	2.17E+04	1.75E+05
5	3.89E+05	3.13E+05	1.86E+06
6	4.59E+02	5.18E+02	4.32E+03
7	1.78E+05	1.55E+05	6.89E+05
8	1.88E+04	1.75E+04	1.56E+05
9	2.36E+04	2.44E+04	1.96E+05
10	5.00E+02	6.37E+02	4.93E+03
11	1.77E+03	1.88E+03	1.53E+04
12	1.60E+03	1.77E+03	1.36E+04
13	9.02E+02	9.26E+02	8.99E+03
14	1.29E+03	1.27E+03	1.26E+04
15	7.43E+02	7.80E+02	7.44E+03
16	2.14E+04	2.82E+04	2.01E+05
17	3.69E+04	3.89E+04	3.08E+05
18	4.26E+04	4.17E+04	3.49E+05
19	3.81E+03	4.02E+03	3.53E+04
20	4.38E+03	4.72E+03	4.34E+04
21	6.11E+03	6.36E+03	6.14E+04
22	1.19E+03	1.21E+03	1.19E+04
23	3.48E+02	3.72E+02	2.85E+03
24	6.17E+02	3.19E+02	3.01E+03
25	3.17E+02	3.49E+02	3.20E+03

EFs are significantly different from the average value, such as the second case where the EF for the NOB algorithm is 0.89 and that for the MOB algorithm is 1.34. For these cases, the real uncertainties should be examined by repeating the calculations many times with different random seeds.

Moreover, the Figure of merits of effective neutron generation times are shown in Table 9. The ratio of the FOMs of the NOB algorithm to those of the superhistory algorithm is 1.03 on average whilst the ratio of the FOMs of the MOB algorithm to those of the superhistory algorithm is 8.18 on average, indicating in terms of computing effective neutron generation times, the efficiency of the NOB algorithm is almost the same as that of the superhistory algorithm, while the MOB algorithm is more efficient than the other two algorithms by a factor of 8.18.

Since some of the experimental results are given in Rossi-alpha, α,

Please cite this article in press as: Qiu, Y., et al., Calculation of adjoint-weighted kinetic parameters with the Reactor Monte Carlo code RMC, Progress in Nuclear Energy (2017), http://dx.doi.org/10.1016/j.pnucene.2017.03.023

Table 10
Comparison of Rossi-alphas between MCNP6 and RMC.

ID	MCNP6	superhistory		NOB		MOB	
	α	α	Diff.	α	Diff.	α	Diff.
11	−1.16E-01	−1.17E-01	−9.48E-04	−1.17E-01	−1.08E-03	−1.17E-01	−1.04E-03
12	−1.17E-01	−1.17E-01	−2.99E-04	−1.17E-01	−5.92E-04	−1.18E-01	−1.08E-03
13	−2.92E-01	−3.01E-01	−8.41E-03	−2.96E-01	−4.12E-03	−2.97E-01	−4.66E-03
14	−3.93E-01	−3.96E-01	−2.97E-03	−3.98E-01	−5.08E-03	−3.98E-01	−4.73E-03
15	−2.10E-01	−2.12E-01	−2.12E-03	−2.09E-01	4.83E-04	−2.12E-01	−2.04E-03
16	−1.13E+00	−1.15E+00	−1.70E-02	−1.15E+00	−2.03E-02	−1.15E+00	−1.54E-02
17	−6.49E-01	−6.54E-01	−5.49E-03	−6.51E-01	−1.99E-03	−6.41E-01	8.23E-03
18	−1.06E+00	−1.08E+00	−1.87E-02	−1.08E+00	−2.16E-02	−1.08E+00	−1.75E-02
19	−1.23E-04	−1.23E-04	−2.57E-07	−1.22E-04	7.33E-07	−1.22E-04	7.95E-07
20	−1.07E-04	−1.07E-04	−2.09E-07	−1.06E-04	4.73E-07	−1.06E-04	5.38E-07
21	−1.26E-04	−1.25E-04	1.06E-06	−1.26E-04	6.89E-07	−1.26E-04	5.56E-07
22	−2.10E-01	−2.13E-01	−2.39E-03	−2.13E-01	−2.85E-03	−2.12E-01	−1.64E-03
23	−3.64E-03	−3.71E-03	−7.22E-05	−3.74E-03	−1.02E-04	−3.72E-03	−8.57E-05
24	−3.20E-02	−3.29E-02	−8.96E-04	−3.28E-02	−7.54E-04	−3.30E-02	−9.10E-04
25	−4.13E-02	−4.18E-02	−5.10E-04	−4.13E-02	−2.80E-05	−4.16E-02	−3.89E-04

Table 11
Standard deviations and estimation factors of Rossi-alphas.

ID	MCNP6	superhistory	NOB		MOB	
	Std	Std	Std	EF	Std	EF
11	6.91E-04	6.44E-04	6.50E-04	1.01	2.04E-04	1.00
12	6.99E-04	6.26E-04	6.44E-04	1.03	2.04E-04	1.02
13	2.56E-03	2.55E-03	2.47E-03	0.98	7.90E-04	0.99
14	2.56E-03	2.45E-03	2.50E-03	1.01	7.77E-04	1.00
15	2.14E-03	2.09E-03	2.08E-03	1.01	6.53E-04	0.99
16	7.41E-03	7.10E-03	7.18E-03	1.01	2.26E-03	1.01
17	7.60E-03	7.20E-03	7.33E-03	1.02	2.26E-03	1.02
18	9.83E-03	9.50E-03	9.52E-03	1.00	3.00E-03	1.00
19	9.38E-07	5.64E-07	5.80E-07	1.04	1.81E-07	1.02
20	7.89E-07	4.83E-07	4.84E-07	1.01	1.52E-07	1.00
21	9.74E-07	5.98E-07	5.87E-07	0.98	1.88E-07	0.99
22	2.33E-03	2.28E-03	2.23E-03	0.97	7.12E-04	0.99
23	2.25E-05	2.22E-05	2.22E-05	0.99	6.98E-06	0.99
24	2.01E-04	1.94E-04	1.96E-04	1.02	6.18E-05	1.01
25	2.57E-04	2.52E-04	2.44E-04	0.98	7.78E-05	0.98

$$\alpha = -\frac{\beta_{eff}}{\Lambda_p}, \tag{30}$$

where Λ_p is the effective generation time for prompt neutrons, this parameter should also be computed for verification and validation.

Table 10 exhibits the Rossi-alphas computed by MCNP6 and different algorithms of RMC and their differences. And the values of Rossi-alpha, the differences in Table 10 and the standard deviations in Table 11 are all given in μs^{-1}. Again, the MCNP6 results are taken as the reference results. As presented in Table 10, the differences in the Rossi-alpha between MCNP6 and any of the three algorithms of RMC, i.e., the superhistory algorithm, the NOB algorithm and the MOB algorithm, are within five standard deviations, indicating all three algorithms of RMC agree with MCNP6.

Additionally, as shown in Table 11, the average EF for both the NOB and MOB algorithms are 1.00, which suggests both algorithms avoid underestimation of the variances of Rossi-alphas compared to the superhistory algorithm.

Furthermore, according to Table 12, the ratio of the FOMs of the NOB algorithm to those of the superhistory algorithm is 1.02 on average while the ratio of the FOMs of the MOB algorithm to those of the superhistory algorithm is 8.95 on average, which indicates in terms of computing Rossi-alphas, the efficiency of the NOB algorithm is almost the same as that of the superhistory algorithm, while the MOB algorithm is more efficient than the other two algorithms by a factor of 8.95.

Table 12
Figure of merits and estimation factors and of Rossi-alphas.

ID	superhistory	NOB	MOB
11	3.57E+01	3.71E+01	3.07E+02
12	3.45E+01	3.46E+01	2.78E+02
13	3.57E+01	3.82E+01	3.63E+02
14	6.16E+01	6.06E+01	6.10E+02
15	2.10E+01	2.13E+01	2.14E+02
16	3.78E+02	4.98E+02	3.55E+03
17	2.38E+02	2.29E+02	1.89E+03
18	3.66E+02	3.63E+02	2.96E+03
19	3.30E+01	3.20E+01	2.88E+02
20	3.46E+01	3.57E+01	3.36E+02
21	4.67E+01	5.06E+01	4.66E+02
22	3.36E+01	3.52E+01	3.32E+02
23	3.23E+01	3.48E+01	2.69E+02
24	3.05E+01	1.57E+01	1.50E+02
25	1.07E+01	1.21E+01	1.11E+02

Table 13 exhibits the difference in Rossi-alphas among MCNP6, RMC and experimental results (Nuclear Science Committee, 2010; Brookhaven National Laboratory, 1986; Tonoike et al., 2002). The experimental results are taken as the reference results for the test problems. As can be seen, the differences in the Rossi-alpha between the experiments and either MCNP6 or any of the three algorithms of RMC, i.e., the superhistory algorithm, the NOB algorithm and the MOB algorithm are within nine standard deviations.

It should be noted that since the memory consumption of the adjoint-weighted kinetic parameters is not significant compared to the memory consumption of k_{eff} sensitivity coefficients to nuclear data. Therefore, the memory usage for different methods were not investigated in terms of the adjoint-weighted kinetic parameters. For the memory consumption of k_{eff} sensitivity coefficients to nuclear data, please refer to the reference (Qiu et al., 2016).

5. Conclusions

In this work, three algorithms, namely, the Non-Overlapping Blocks algorithm, the Multiple Overlapping Blocks algorithm and the superhistory algorithm were implemented in the Reactor Monte Carlo code RMC to compute adjoint-weighted kinetic parameters. These algorithms were compared in terms of accuracy, efficiency and estimation of variance. All three algorithms in RMC agree well with both MCNP6 and experimental results. In terms of

Please cite this article in press as: Qiu, Y., et al., Calculation of adjoint-weighted kinetic parameters with the Reactor Monte Carlo code RMC, Progress in Nuclear Energy (2017), http://dx.doi.org/10.1016/j.pnucene.2017.03.023

Table 13
Comparison of Rossi-alphas between experimental results and calculation results.

ID	Experimental results		MCNP6	Superhistory	NOB	MOB
	α	Std				
11	−1.17E-01	1.00E-03	1.38E-03	4.28E-04	3.00E-04	3.40E-04
12	−1.17E-01	1.00E-03	4.85E-04	1.86E-04	−1.07E-04	−5.93E-04
13	−2.67E-01	5.00E-03	−2.53E-02	−3.37E-02	−2.94E-02	−3.00E-02
14	−3.82E-01	2.00E-03	−1.13E-02	−1.43E-02	−1.64E-02	−1.60E-02
15	−2.14E-01	5.00E-03	4.03E-03	1.92E-03	4.52E-03	1.99E-03
16	−1.10E+00	2.00E-02	−3.06E-02	−4.75E-02	−5.08E-02	−4.59E-02
17	−6.40E-01	1.00E-02	−9.00E-03	−1.45E-02	−1.10E-02	−7.75E-04
18	−1.00E+00	1.00E-02	−6.03E-02	−7.90E-02	−8.19E-02	−7.77E-02
19	−1.22E-04	4.00E-06	−6.15E-07	−8.72E-07	1.18E-07	1.80E-07
21	−1.27E-04	2.90E-06	5.16E-07	1.58E-06	1.21E-06	1.07E-06
22	−1.97E-01	1.00E-02	−1.33E-02	−1.57E-02	−1.61E-02	−1.49E-02
23	−3.38E-03	7.40E-05	−2.56E-04	−3.28E-04	−3.57E-04	−3.41E-04
25	−3.73E-02	4.77E-04	−3.98E-03	−4.49E-03	−4.01E-03	−4.37E-03

efficiency, the Non-Overlapping Blocks algorithm is as efficient as the superhistory algorithm while the Multiple Overlapping Blocks algorithm are more efficient than the other two algorithms by almost a factor of a block size. Moreover, while the variances of k_{eff} obtained by the power method are underestimated by 7% compared to those obtained by the superhistory method, the differences in variances of adjoint-weighted kinetic parameters among the three algorithms are ignorable, suggesting the three algorithms almost produce the same level of estimation of variance. Efforts should be made to understand the reason why there are not differences in estimation of variances of adjoint-weighted kinetic parameters among different algorithms in the future work. Additionally, the superhistory algorithm requires much lower memory than the other two algorithms and can reduce the large consumption of memory needed for sensitivity coefficient calculations (Qiu et al., 2016). Future work will focus on developing a capability of computing generalized sensitivity coefficients based on the superhistory method.

Acknowledgements

This research is partially supported by National Natural Science Foundation of China (Grant No: 11475098) and Science and Technology on Reactor System Design Technology Laboratory. The authors wish to thank Tsinghua National Laboratory for Information Science and Technology for providing the access to Inspur TS10000 HPC Cluster.

References

Blomquist, R.N., Gelbard, E.M., 2002. Alternative implementations of the Monte Carlo power method. Nucl. Sci. Eng. 141, 85–100.
Brissenden, R.J., Garlick, A.R., 1986. Biases in the estimation of k_{eff} and its error by Monte Carlo methods. Ann. Nucl. Energy 113, 63–83.
Brookhaven National Laboratory, 1986. Cross Section Evaluation Working Group Benchmark Specifications. BNL-19302, ENDF-202(Rev., September 1986).
Choi, S.H., Shim, H.J., 2014. Efficient Estimation of Adjoint-weighted Kinetics Parameters in the Monte Carlo Wielandt Calculations. PHYSOR 2014 – the Role of Reactor Physics toward a Sustainable Future. The Westin Miyako, Kyoto, Japan. September 28 – October 3, 2014, on CD-ROM.
Gelbard, E.M., Prael, R., 1990. Computation of standard deviations in Eigenvalue calculations. Prog. Nucl. Energy 24 (1–3), 237–241.
Goorley, J.T., James, M.R., Booth, T.E., et al., 2013. Initial MCNP6 Release Overview - MCNP6 Version 1.0. Los Alamos National Laboratory. LA-UR-13–22934.
Hurwitz, H., 1964, 864, Naval Reactors. In: Radkowsky, A. (Ed.), Naval Reactor Physics Handbook, vol. I. U.S. Atomic Energy Commission.
Kiedrowski, B.C., 2010. Theory, Interface, Verification, Validation, and Performance of the Adjoint-weighted Point Reactor Kinetics Parameter Calculations in MCNP. Los Alamos National Laboratory. LA-UR-10–01700.
Kiedrowski, B.C., Brown, F.B., 2009. Adjoint-weighted Kinetics Parameters with Continuous Energy Monte Carlo. Los Alamos National Laboratory. LA-UR-09–00544.
Kiedrowski, B.C., Brown, F.B., Wilson, P.P.H., 2011. Adjoint-weighted tallies for k-eigenvalue calculations with continuous-energy Monte Carlo. Nucl. Sci. Eng. 168 (3), 226–241.
Leppänen, J., Aufiero, M., Fridman, E., et al., 2014. Calculation of effective point kinetics parameters in the Serpent 2 Monte Carlo code. Ann. Nucl. Energy 65, 272–279.
Mosteller, R.D., Kiedrowski, B.C., 2011. The Rossi Alpha Validation Suite for MCNP. International Conference on Nuclear Criticality. Scotland, Edinburgh, 19–22 September 2011.
NEA Nuclear Science Committee, 2010. International Handbook of Evaluated Criticality Safety Benchmark Experiments. OECD Nuclear Energy Agency. NEA/NSC/DOC(95).
Nauchi, Y., Kameyama, T., 2010. Development of calculation technique for Iterated Fission Probability and reactor kinetic parameters using continuous-energy Monte Carlo method. J. Nucl. Sci. Technol. 47, 977–990.
Qiu, Y., Liang, J., Wang, K., Yu, J., 2015. New strategies of sensitivity analysis capabilities in continuous-energy Monte Carlo code RMC. Ann. Nucl. Energy 81, 50–61.
Qiu, Y., Shang, X., Tang, X., Liang, J., Wang, K., 2016. Computing eigenvalue sensitivity coefficients to nuclear data by adjoint superhistory method and adjoint Wielandt method implemented in RMC code. Ann. Nucl. Energy 87, 228–241.
She, D., Wang, K., Yu, G., 2012. Asymptotic Wielandt method and superhistory method for source convergence in Monte Carlo criticality calculation. Nucl. Sci. Eng. 172, 127–137.
Tonoike, K., Miyoshi, Y., Kikuchi, T., et al., 2002. Kinetic parameter β_{eff}/ℓ measurement on low enriched uranyl nitrate solution with single unit cores (600φ, 280T, 800φ) of STACY. J. Nucl. Sci. Tech. 39, 1227–1236.
Truchet, G., Leconte, P., Peneliau, Y., et al., 2013. Continuous-Energy adjoint flux and perturbation calculation using the iterated fission probability method in Monte Carlo code tripoli-4 and underlying applications. In: Joint International Conference on Supercomputing in Nuclear Applications and Monte Carlo 2013 (SNA + MC 2013). Paris, France, October 27–31.
Wang, K., Li, Z., She, D., et al., 2015a. RMC – a Monte Carlo code for reactor core analysis. Ann. Nucl. Energy 82, 121–129.
Wang, K., Liang, J., Yu, J., et al., 2015b. New Features and Enhancements of Reactor Monte Carlo Code RMC. ANS MC2015-joint International Conference on Mathematics and Computation (M&C), Supercomputing in Nuclear Applications (SNA) and the Monte Carlo (MC) Method, Nashville, TN, April 19-23, 2015, on CD-ROM. American Nuclear Society, LaGrange Park, IL (2015).
Xu, Q., Wang, K., 2014. Development of Quasi-static Dynamic Simulation Capability in Monte Carlo Code RMC, vol. 110. Transactions of the American Nuclear Society. Reno, Nevada, June 15–19, 2014.

Please cite this article in press as: Qiu, Y., et al., Calculation of adjoint-weighted kinetic parameters with the Reactor Monte Carlo code RMC, Progress in Nuclear Energy (2017), http://dx.doi.org/10.1016/j.pnucene.2017.03.023