Journal of Computational Physics 228 (2009) 8525–8547

Contents lists available at ScienceDirect

Journal of Computational Physics

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

Time series analysis and Monte Carlo methods for eigenvalue separation in neutron multiplication problems

Brian R. Nease [a], Taro Ueki [b,*]

[a] Commissariat á l'Énergie Atomique, Saclay, France
[b] University of New Mexico, Department of Chemical and Nuclear Engineering, 209 Farris Engineering Center, Albuquerque, NM 87131, United States

ARTICLE INFO

Article history:
Received 13 November 2008
Received in revised form 6 June 2009
Accepted 20 July 2009
Available online 19 August 2009

Keywords:
Time series
Monte Carlo
Eigenvalues
Neutron multiplication
Nuclear criticality

ABSTRACT

A time series approach has been applied to the nuclear fission source distribution generated by Monte Carlo (MC) particle transport in order to calculate the non-fundamental mode eigenvalues of the system. The novel aspect is the combination of the general technical principle of projection pursuit for multivariate data with the neutron multiplication eigenvalue problem in the nuclear engineering discipline. Proof is thoroughly provided that the stationary MC process is linear to first order approximation and that it transforms into one-dimensional autoregressive processes of order one (AR(1)) via the automated choice of projection vectors. The autocorrelation coefficient of the resulting AR(1) process corresponds to the ratio of the desired mode eigenvalue to the fundamental mode eigenvalue. All modern MC codes for nuclear criticality calculate the fundamental mode eigenvalue, so the desired mode eigenvalue can be easily determined.

This time series approach was tested for a variety of problems including multi-dimensional ones. Numerical results show that the time series approach has strong potential for three dimensional whole reactor core. The eigenvalue ratio can be updated in an on-the-fly manner without storing the nuclear fission source distributions at all previous iteration cycles for the mean subtraction. Lastly, the effects of degenerate eigenvalues are investigated and solutions are provided.

1. Introduction

The conservative transport equation is a general partial differential equation that can describe transport phenomena such as heat transfer, fluid dynamics, etc. In the nuclear engineering discipline, the transport equation is integro-differential and used to model neutron economy, i.e, the balance of neutron population in the presence of nuclear reactions and leakage out of the domain of interest [1]. In this context, it is termed the neutron transport equation, and solutions of the steady-state neutron transport equation exist for specific gauging parameters called k-eigenvalues. Monte Carlo (MC) techniques exist to compute the largest k-eigenvalue by simulating a sufficient number of realizations of the problem through iterating cycles of particle tracking and population normalization. In this way, the eigenvalue is calculated within a statistical uncertainty. In this article a novel method of calculating the k-eigenvalue separations with MC calculations is presented. It is termed the Coarse Mesh Projection Method (CMPM).

The k-eigenvalues each correspond to a particular fundamental or non-fundamental mode eigenfunction. The fundamental mode eigenfunction is everywhere non-negative over the physical space of the problem, and in MC calculations it manifests itself as the stationary nuclear fission source distribution. The non-fundamental mode eigenfunctions are those that

* Corresponding author. Tel.: +1 505 277 7964; fax: +1 505 277 5433.
E-mail addresses: bnease@gmail.com (B.R. Nease), tueki@unm.edu (T. Ueki).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.07.037

assume both positive and negative values, and in MC calculations they manifest themselves as fluctuating components. Calculating the fundamental mode eigenvalue is considered routine using power iteration methods [2], but calculating the non-fundamental mode eigenvalues is often quite difficult. Throughout this work we refer to the nuclear fission source distribution and the non-fundamental mode k-eigenvalues simply as source distribution and eigenvalues, respectively, for the sake of repetition. The application of CMPM results in the computation of eigenvalue separation in the form of the ratio of the (desired mode) eigenvalue to the fundamental mode k-eigenvalue. Since the fundamental mode k-eigenvalue is easily obtained using MC techniques, the eigenvalue follows easily.

CMPM utilizes projection techniques to transform the problem into a one-dimensional autoregressive process, which then allows the eigenvalue separation to be solved for using the standard time series technique with linear least square. Determination of the projection vector is automated and assures that only information about the desired eigenmode is extracted. CMPM has two advantages over recent work [3] on the computation of eigenvalue ratio via time series analysis. First, CMPM does not use non-linear least square iterations upon fitting and thus leads to the elimination of an initial guess of fitting coefficients. Second, the capability of CMPM is not restricted to the computation of the eigenvalue ratio of the first mode to the fundamental mode. In this article, the CMPM methodology is fully described, beginning from a general form of the transport equation.

2. Monte Carlo (MC) transport

The neutron transport equation describes the evolution of neutron distribution in a system over time, given material properties, geometry, and initial and boundary conditions. The equation is comprised of several terms that account for the creation and removal of neutrons in the system. For nuclear reactors and their fuel storage and processing facilities, creation of neutrons is governed by nuclear fission, which has a chance to occur when neutrons are absorbed in fissionable isotopes. Removal of neutrons is predominantly governed by absorption in non-fissionable isotopes and leakage out of the system. Until a neutron is either absorbed or leaked out, it simply scatters around within the system.

Due to the complexity of the transport equation, it is often written using a fission operator $\mathbf{F}\Psi$ and a transport operator $\mathbf{T}\Psi$ with Ψ standing for neutron flux. Mathematically, these operators are defined as [1]

$$\mathbf{F}\Psi = \int_0^{E_{\max}} \int_{4\pi} \nu\Sigma_f(\vec{r}, E')\,\Psi\left(\vec{r}, \vec{\Omega}', E'\right) d^2\Omega'\, dE', \tag{1}$$

$$\mathbf{T}\Psi = \vec{\Omega}\cdot\vec{\nabla}\Psi\left(\vec{r}, \vec{\Omega}, E\right) + \Sigma_t(\vec{r}, E)\Psi\left(\vec{r}, \vec{\Omega}, E\right) - \int_0^{E_{\max}} \int_{4\pi} \Sigma_s\left(\vec{r}, \vec{\Omega}', E' \to \vec{\Omega}, E\right)\Psi\left(\vec{r}, \vec{\Omega}', E'\right) d^2\Omega' dE', \tag{2}$$

where ν is the average number of neutrons generated per fission event, Σ_f is the macroscopic fission cross section, Σ_s is the macroscopic differential scattering cross section, Σ_t is the macroscopic total cross section, E is energy, $\vec{\Omega}$ is the unit vector in direction of movement, and $\vec{r}$ is spatial coordinate vector. The macroscopic cross sections represent the likelihood of a particular event per unit distance traveled. Using these operators and including the gauging parameter k_{eff}, the transport equation can be made time-independent:

$$\mathbf{T}\Psi = \frac{1}{k_{\text{eff}}}\frac{\chi(E)}{4\pi}\mathbf{F}\Psi. \tag{3}$$

In this equation, $\chi(E)$ represents the energy spectrum of fission-born neutrons, the 4π indicates that the fission neutrons are born isotropically in direction, and k_{eff} represents the fundamental mode eigenvalue termed the effective neutron multiplication factor.

To obtain the eigenvalue equation of source distribution, the operator $\mathbf{T}$ in Eq. (3) is formally inverted and both sides of the equation are operated on by $\mathbf{F}$

$$\mathbf{F}\Psi = \frac{1}{k_{\text{eff}}}\left[\mathbf{F}\mathbf{T}^{-1}\frac{\chi(E)}{4\pi}\right]\mathbf{F}\Psi. \tag{4}$$

It is common practice to rewrite the $\mathbf{F}\Psi$ as $S(\vec{r})$. This term $S(\vec{r})$ is called the source distribution since it results from the application of the fission operator $\mathbf{F}$ to the flux Ψ. Therefore, $S(\vec{r})$ is neither probability density function nor cumulative distribution function, and as is the case for the neutron flux Ψ in Eq. (2), there is freedom in normalizing $S(\vec{r})$. After defining a new integral kernel $\mathbf{H}(\vec{r}' \to \vec{r})$ as

$$\int_V \mathbf{H}(\vec{r}' \to \vec{r}) f(\vec{r}')\, dV' = \left[\mathbf{F}\mathbf{T}^{-1}\frac{\chi(E)}{4\pi}\right](f). \tag{5}$$

Eq. (4) can be rewritten as

$$S(\vec{r}) = \frac{1}{k_{\text{eff}}}\int_V \mathbf{H}(\vec{r}' \to \vec{r}) S(\vec{r}')\, dV'. \tag{6}$$

Physically, this kernel $\mathbf{H}(\vec{r}' \to \vec{r})$ is to be understood as the expected number of direct-descendent (first generation) neutrons produced via fission per unit volume at $\vec{r}$ due to a fission neutron that was produced at $\vec{r}'$. Eq. (6) is a special case of the general eigenvalue equation shown below.

$$S_j(\vec{r}) = \frac{1}{k_j}\int_V \mathbf{H}(\vec{r}' \to \vec{r}) S_j(\vec{r}')\, dV' \tag{7}$$

IIn this equation, the eigenvalues $k_j, j \geqslant 0$ are assumed to be discrete and ordered $|k_0| > |k_1| \geqslant |k_2| \geqslant \cdots$ and the corresponding eigenfunctions are denoted $S_j(\vec{r}), j = 0, 1, \ldots$ (where "0" implies the fundamental mode). The fundamental mode eigenvalue k_0 is k_{eff} and its eigenfunction is normalized as

$$k_{\text{eff}} \equiv k_0 = \int_V S_0(\vec{r})\, dV. \tag{8}$$

This normalization cannot be done for all non-fundamental mode eigenfunctions because some modes may integrate to zero over the domain. If not, the following normalization is made.

$$k_j = \int_V S_j(\vec{r})\, dV \quad \text{when} \quad \int_V S_j(\vec{r})\, dV \neq 0. \tag{9}$$

It is not known whether all eigenvalues of continuous energy cross-section problems are real and discrete. However, proofs do exist for mono-energetic transport, regardless of whether scattering is isotropic or anisotropic [4].

Instead of trying to solve for the eigenvalues of Eq. (7) directly by estimating and solving the matrix form of $\mathbf{H}(\vec{r}' \to \vec{r})$, a time series approach termed the Coarse Mesh Projection Method (CMPM) has been developed. CMPM is based on the asymptotic linearity of the MC process whose rigorous derivation is presented below. The backbone of this theory has been compiled from several sources [3,5–7] whose origin dates back to classical work by Lieberoth [8]. The theory has been significantly expanded with the proof of an assumption made in these sources.

2.1. Asymptotic linearity of the stationary monte carlo process

In modern MC calculations, batches of neutrons are generated in a specified distribution and tracked to accumulate statistics of physical properties. Simulating one batch of neutrons (one realization of the problem) is known as running one cycle. This process is repeated for many cycles, always with the same number, i.e., the same total weight of starting neutrons. The starting location for the neutrons in each cycle is dependent on the location of fission events from the previous cycle. These starting locations are updated each cycle so that they eventually match the true source distribution of the problem within statistical fluctuations. When this condition is met, the distribution is said to be stationary.

Assuming that the source distribution is stationary, a particular realization of the source distribution at the end of cycle m, denoted $\widehat{S}^{(m)}(\vec{r})$, is expressed as the sum of a deterministic part and a fluctuating part

$$\widehat{S}^{(m)}(\vec{r}) = NS(\vec{r}) + \sqrt{N}\widehat{e}^{(m)}(\vec{r}). \tag{10}$$

The hats above $\widehat{S}^{(m)}(\vec{r})$ and $\widehat{e}^{(m)}(\vec{r})$ are used to indicate a stochastic realization of the raw (unnormalized) source distribution representing the collection of fission sites and statistical weights after simulating all particles in cycle m and N represents the number of particles per cycle. The terms N and $\sqrt{N}$ are included as scaling factors, as will be explained next.

The deterministic part $S(\vec{r})$ is the expectation of source distribution defined as

$$S(\vec{r}) \equiv \frac{1}{N} E\left[\widehat{S}^{(m)}(\vec{r})\right] \tag{11}$$

Eqs. (10) and (11) imply that

$$E\left[\widehat{e}^{(m)}(\vec{r})\right] = 0. \tag{12}$$

Since $\int_V \widehat{S}^{(m)}(\vec{r})\, dV \simeq O(N)$ where V is the domain of space coordinates, it is clear that $\int_V S(\vec{r})\, dV \simeq O(1)$. The variance of $\int_V \widehat{S}^{(m)}(\vec{r})\, dV$ is assumed to be $O(N)$ taking into account particle population dynamics as in previous work [3,5–7]. Therefore,

$$O(N) = \text{var}\left[\int_V \widehat{S}^{(m)}(\vec{r})\, dV\right] \equiv E\left[\left\{\int_V \left[\widehat{S}^{(m)}(\vec{r}) - NS(\vec{r})\right] dV\right\}^2\right]. \tag{13}$$

Applying Eq. (10) yields

$$O(N) = E\left[\left\{\int_V \sqrt{N}\widehat{e}^{(m)}(\vec{r})\, dV\right\}^2\right] \tag{14}$$

indicating that $\int_V \widehat{e}^{(m)}(\vec{r})\, dV \simeq O(1)$. Since the expected number of fission born neutrons in the domain V is proportional to N, the number of fission born neutrons may be considered the Poisson random variable with the expectation proportional to N

if each starting neutron is equated to a radioactive entity. The variance of a Poisson random variable is equal to its mean. Therefore, the population dynamics formulation of scaling, $\mathrm{var}\left[\int_V \widehat{S}^{(m)}(\vec{r})\,dV\right] \simeq O(N)$, is the only assumption in the theory presented below unlike previous work [3,5–7] which additionally made an assumption about the expectation of $\widehat{S}^{(m+1)}$ conditional upon a realization of $\widehat{S}^{(m)}$. After each cycle m, the fundamental mode eigenvalue is estimated as

$$\widehat{k}^{(m)} = \frac{1}{N}\int_V \widehat{S}^{(m)}(\vec{r})\,dV. \tag{15}$$

Using Eq. (11), the expected eigenvalue is

$$k \equiv E\left[\widehat{k}^{(m)}\right] = \int_V S(\vec{r})\,dV. \tag{16}$$

This is the source normalization condition independent of N and corresponding to Eq. (8).

Next, a model is developed that describes how the source distribution evolves from one cycle to the next. The fission source distribution after simulating one batch of particles can be written as

$$\widehat{S}^{(m)}(\vec{r}) = \sum_{i=1}^{C^{(m)}} w_i^{(m)}\delta(\vec{r}-\vec{r}_i), \tag{17}$$

where $C^{(m)}$ is the number of fission sites during the mth stationary cycle, $w_i^{(m)}$ is the statistical weight assigned to the ith fission site, and δ is a Dirac delta function such that $\int \delta(\vec{r}-\vec{r}_i)(f)\,dV = f(\vec{r}_i)$ and $\delta(\vec{r}-\vec{r}_i) = 0$ when $\vec{r} \neq \vec{r}_i$. A vector may be introduced to represent the collection of weights that correspond to the specific fission sites $\vec{r}_1, \vec{r}_2, \ldots, \vec{r}_{C^{(m)}}$:

$$\vec{W}^{(m)} \equiv \left(w_1^{(m)}, w_2^{(m)}, \ldots, w_{C^{(m)}}^{(m)}\right),$$

where using a sufficiently small volume ΔV_j containing only $\vec{r}_j$,

$$w_j^{(m)} = \left(\vec{W}^{(m)}\right)_j = \int_{\Delta V_j} \widehat{S}^{(m)}(\vec{r})\,dV, \quad \vec{r}_j \in \Delta V_j,\ \vec{r}_i \notin \Delta V_j,\ i \neq j.$$

The particle weights are normalized such that

$$\sum_{j=1}^{C^{(m)}} w_j^{(m)} = \int_V \widehat{S}^{(m)}(\vec{r})\,dV. \tag{18}$$

The probability that a particular fission site $\vec{r}_j$ is chosen as a neutron source location at the beginning of cycle $m+1$ is determined from the weight distribution of the previous cycle m as

$$p^{(m+1)}\left(\vec{r} = \vec{r}_j\right) = \frac{w_j^{(m)}}{\sum_{i=1}^{C^{(m)}} w_i^{(m)}}. \tag{19}$$

If N total starter neutrons are used in cycle $m+1$, then it is expected that $Np^{(m+1)}\left(\vec{r} = \vec{r}_j\right)$ will start at that particular fission site $\vec{r}_j$.

Since the kernel $\mathbf{H}(\vec{r}_j \to \vec{r})$ in Eq. (6) can be interpreted as the expected next-generation fission-born-neutron weight per unit volume at $\vec{r}$ given the unit fission-born-neutron weight at $\vec{r}_j$, the expected weight distribution for neutrons that will start at $\vec{r}$ after simulating N neutron histories in cycle $m+1$, assuming that the unit weight at $\vec{r}_j$ is the only weight produced in cycle m, is formally expressed as

$$E\left[\widehat{S}^{(m+1)}(\vec{r})|\vec{W}^{(m)} = \left(0,\ldots,0,\underbrace{1}_{j\text{th element}},0,\ldots,0\right)\right] = E\left[\widehat{S}^{(m+1)}(\vec{r})|\widehat{S}^{(m)}(\vec{r}) = \delta\left(\vec{r}-\vec{r}_j\right)\right] = N\mathbf{H}\left(\vec{r}_j \to \vec{r}\right). \tag{20}$$

If Eq. (20) is multiplied by $p^{(m+1)}\left(\vec{r} = \vec{r}_j\right)$ from Eq. (19) and is summed over all fission sites $\left(\vec{r}_1, \vec{r}_2, \ldots, \vec{r}_{C^{(m)}}\right)$, then the expected weight distribution given the distribution of weights $\vec{W}^{(m)}$ in Eq. (17) is obtained as

$$\begin{aligned} E\left[\widehat{S}^{(m+1)}(\vec{r})|\widehat{S}^{(m)}(\vec{r}) = \sum_{i=1}^{C^{(m)}} w_i^{(m)}\delta(\vec{r}-\vec{r}_i)\right] &= E\left[\widehat{S}^{(m+1)}(\vec{r})|\vec{W}^{(m)} = \left(w_1^{(m)}, w_2^{(m)}, \ldots, w_{C^{(m)}}^{(m)}\right)\right] \\ &= \sum_{j=1}^{C^{(m)}} E\left[\widehat{S}^{(m+1)}(\vec{r})|\vec{W}^{(m)} = (0,\ldots,0,\underbrace{1}_{j\text{th element}},0,\ldots,0)\right] p^{(m+1)}\left(\vec{r} = \vec{r}_j\right) \\ &= N\frac{\sum_{j=1}^{C^{(m)}} w_j^{(m)}\mathbf{H}\left(\vec{r}_j \to \vec{r}\right)}{\sum_{j=1}^{C^{(m)}} w_j^{(m)}}. \end{aligned} \tag{21}$$

The numerator of Eq. (21) can be manipulated as

$$\sum_{j=1}^{C^{(m)}} w_j^{(m)} \mathbf{H}(\vec{r}_j - \vec{r}) = \sum_{j=1}^{C^{(m)}} w_j^{(m)} \mathbf{H}(\vec{r}_j \to \vec{r}) \int_V \delta(\vec{r}' - \vec{r}_j)\, dV' \tag{22}$$

$$= \sum_{j=1}^{C^{(m)}} \int_V w_j^{(m)} \delta(\vec{r}' - \vec{r}_j) \mathbf{H}(\vec{r}_j \to \vec{r})\, dV' \tag{23}$$

$$= \sum_{j=1}^{C^{(m)}} \int_V w_j^{(m)} \delta(\vec{r}' - \vec{r}_j) \mathbf{H}(\vec{r}' \to \vec{r})\, dV'. \tag{24}$$

$$= \int_V \sum_{j=1}^{C^{(m)}} w_j^{(m)} \delta(\vec{r}' - \vec{r}_j) \mathbf{H}(\vec{r}' \to \vec{r})\, dV'. \tag{25}$$

Eq. (24) is obtained from the relation $\delta(x-y)f(x) = \delta(x-y)f(y)$. Apply Eq. (17) to Eq. (25) to obtain

$$\sum_{j=1}^{C^{(m)}} w_j^{(m)} \mathbf{H}(\vec{r}_j - \vec{r}) = \int_V \widehat{S}^{(m)}(\vec{r}') \mathbf{H}(\vec{r}' \to \vec{r})\, dV'. \tag{26}$$

Using Eq. (18) and (26), Eq. (21) is rewritten as

$$E\left[\widehat{S}^{(m+1)}(\vec{r}) | \widehat{S}^{(m)}(\vec{r})\right] = N \frac{\int_V \widehat{S}^{(m)}(\vec{r}') \mathbf{H}(\vec{r}' \to \vec{r})\, dV'}{\int_V \widehat{S}^{(m)}(\vec{r}')\, dV'}. \tag{27}$$

This is the conditional distribution that represents the expected source distribution at the end of cycle $m+1$ given the source distribution $\widehat{S}^{(m)}(\vec{r})$ at the end of cycle m. Eq. (27) and the corresponding discrete version were assumed in previous work [3,5–7]. Here Eq. (27) has been formally derived. The stochastic equation describing $\widehat{S}^{(m+1)}$ can then be written as

$$\widehat{S}^{(m+1)}(\vec{r}) = N \frac{\int_V \widehat{S}^{(m)}(\vec{r}') \mathbf{H}(\vec{r}' \to \vec{r})\, dV'}{\int_V \widehat{S}^{(m)}(\vec{r}')\, dV'} + \sqrt{N} \widehat{\varepsilon}^{(m+1)}(\vec{r}). \tag{28}$$

The fluctuating term $\widehat{\varepsilon}^{(m+1)}(\vec{r})$ is a noise component resulting from population normalization of starting neutrons and subsequent transport tracking in cycle $m+1$. As before, the N and $\sqrt{N}$ are scaling terms. Eqs. (27) and (28) imply that $\widehat{\varepsilon}^{(m+1)}(\vec{r})$ satisfies

$$E\left[\widehat{\varepsilon}^{(m+1)}(\vec{r}) | \widehat{S}^{(m)}(\vec{r})\right] = 0. \tag{29}$$

This result further implies that

$$E\left[\widehat{\varepsilon}^{(m+1)}(\vec{r})\right] = E\left[E\left[\widehat{\varepsilon}^{(m+1)}(\vec{r}) | \widehat{S}^{(m)}(\vec{r})\right]\right] = 0. \tag{30}$$

It is important to note that Eqs. (10) and (29) yield,

$$E\left[\widehat{\varepsilon}^{(m+1)}(\vec{r}) | \widehat{e}^{(m)}(\vec{r})\right] = 0. \tag{31}$$

Substituting (10) into (28) and dividing through by N results in

$$S(\vec{r}) + \frac{1}{\sqrt{N}} \widehat{e}^{(m+1)}(\vec{r}) = \frac{\int_V \left[NS(\vec{r}') + \sqrt{N} \widehat{e}^{(m)}(\vec{r}')\right] H(\vec{r}' \to \vec{r})\, dV'}{\int_V \left[NS(\vec{r}') + \sqrt{N} \widehat{e}^{(m)}(\vec{r}')\right] dV'} + \frac{1}{\sqrt{N}} \widehat{\varepsilon}^{(m+1)}(\vec{r}). \tag{32}$$

To reduce Eq. (32) further, the first term on the right hand side (RHS) must be manipulated. First, N^{-1} is multiplied to the numerator and denominator. Next, Eq. (16) is applied to the denominator resulting in

$$\frac{\int_V H(\vec{r}' \to \vec{r}) S(\vec{r}')\, dV' + \frac{1}{\sqrt{N}} \int_V H(\vec{r}' \to \vec{r}) \widehat{e}^{(m)}(\vec{r}')\, dV'}{k\left(1 + \frac{1}{k\sqrt{N}} \int_V \widehat{e}^{(m)}(\vec{r}')\, dV'\right)}. \tag{33}$$

At this point, the denominator is in the form $1+x$, which can be written in series form as $1-x+x^2-x^3+\cdots$. Rewriting the denominator this way yields

$$\left[\frac{1}{k}\int_V H(\vec{r}'\to\vec{r})S(\vec{r}')\,dV'+\frac{1}{k\sqrt{N}}\int_V H(\vec{r}'\to\vec{r})\widehat{e}^{(m)}(\vec{r}')\,dV'\right] \times\left(1-\frac{1}{k\sqrt{N}}\int_V \widehat{e}^{(m)}(\vec{r}')\,dV'+\frac{1}{k^2N}\left(\int_V \widehat{e}^{(m)}(\vec{r}')\,dV'\right)\left(\int_V \widehat{e}^{(m)}(\vec{r}'')\,dV''\right)-\cdots\right). \tag{34}$$

All terms are multiplied out and the terms of order N^{-1} or smaller (higher powers of $N^{-1/2}$) are combined into a leading order term $O\left(N^{-1}\right)$, yielding

$$\frac{1}{k}\int_V H(\vec{r}'\to\vec{r})S(\vec{r}')\,dV'+\frac{1}{k\sqrt{N}}\int_V H(\vec{r}'\to\vec{r})\widehat{e}^{(m)}(\vec{r}')\,dV'-\frac{1}{k^2\sqrt{N}}\int_V H(\vec{r}''\to\vec{r})S(\vec{r}'')\,dV''\int_V \widehat{e}^{(m)}(\vec{r}')\,dV'+O\left(N^{-1}\right). \tag{35}$$

A kernel term is defined

$$\mathbf{A}(\vec{r}'\to\vec{r})=\frac{1}{k}\left[H(\vec{r}'\to\vec{r})-\frac{1}{k}\int_V H(\vec{r}''\to\vec{r})S(\vec{r}'')\,dV''\right] \tag{36}$$

allowing Eq. (35) to be simplified to

$$\frac{1}{k}\int_V H(\vec{r}'\to\vec{r})S(\vec{r}')\,dV'+\frac{1}{\sqrt{N}}\int_V \mathbf{A}(\vec{r}'\to\vec{r})\widehat{e}^{(m)}(\vec{r}')\,dV'+O\left(N^{-1}\right). \tag{37}$$

Now, Eq. (37) is substituted back into the first term on the RHS of Eq. (32) to obtain

$$S(\vec{r})+\frac{1}{\sqrt{N}}\widehat{e}^{(m+1)}(\vec{r})=\frac{1}{k}\int_V H(\vec{r}'\to\vec{r})S(\vec{r}')\,dV'+\frac{1}{\sqrt{N}}\int_V \mathbf{A}(\vec{r}'\to\vec{r})\widehat{e}^{(m)}(\vec{r}')\,dV'+\frac{1}{\sqrt{N}}\widehat{\varepsilon}^{(m+1)}(\vec{r})+O\left(N^{-1}\right). \tag{38}$$

Taking the expectation of Eq. (38) and using Eqs. (12) and (30) yield

$$S(\vec{r})=\frac{1}{k}\int_V H(\vec{r}'\to\vec{r})S(\vec{r}')\,dV'+O\left(N^{-1}\right). \tag{39}$$

Recall that the exact fundamental mode solution is

$$S_0(\vec{r})=\frac{1}{k_0}\int_V H(\vec{r}'\to\vec{r})S_0(\vec{r}')\,dV'. \tag{40}$$

Subtracting Eq. (40) from Eq. (39) yields

$$S(\vec{r})-S_0(\vec{r})-\int_V H(\vec{r}'\to\vec{r})\left[\frac{S(\vec{r}')}{k}-\frac{S_0(\vec{r}')}{k_0}\right]dV'=O\left(N^{-1}\right). \tag{41}$$

Suppose that a bias of order N^{-a} exists in the expected source distribution

$$S(\vec{r})-S_0(\vec{r})=O\left(N^{-a}\right), \tag{42}$$

which also implies that

$$k-k_0=O\left(N^{-a}\right), \tag{43}$$

by Eqs. (8) and (16). Using Eqs. (42) and (43), Eq. (41) can be rewritten

$$O\left(N^{-a}\right)=O\left(N^{-1}\right). \tag{44}$$

It is clear that the order of bias must be equivalent, i.e., $a=1$, otherwise the RHS and LHS of Eq. (44) will differ by orders of magnitude as $N\to\infty$. Therefore, we obtain

$$S(\vec{r})-S_0(\vec{r})=O\left(N^{-1}\right), \tag{45}$$

$$k-k_0=O\left(N^{-1}\right). \tag{46}$$

Now, we can return to the derivation of the cycle-wise representation of the source distribution. Subtracting Eq. (39) from Eq. (38) and multiplying through by $\sqrt{N}$ yields

$$\widehat{e}^{(m+1)}(\vec{r})=\int_V \mathbf{A}(\vec{r}'\to\vec{r})\widehat{e}^{(m)}(\vec{r}')\,dV'+\widehat{\varepsilon}^{(m+1)}(\vec{r})+O\left(N^{-1/2}\right). \tag{47}$$

A new operator $\mathbf{A}_0(\vec{r}' \to \vec{r})$ is introduced:

$$\mathbf{A}_0(\vec{r}' \to \vec{r}) = \frac{1}{k_0}[H(\vec{r}' \to \vec{r}) - S_0(\vec{r})]. \tag{48}$$

From Eqs. (36) (40), (45) and (46), it is clear that

$$\mathbf{A}(\vec{r}' \to \vec{r}) = \mathbf{A}_0(\vec{r}' \to \vec{r}) + O\left(N^{-1}\right). \tag{49}$$

Substituting Eq. (49) in Eq. (47) and using operator notation such that

$$\mathbf{A}_0\widehat{e}^{(m)} = \int \mathbf{A}_0(\vec{r}' \to \vec{r})\widehat{e}^{(m)}(\vec{r}')\,dV', \tag{50}$$

we obtain

$$\widehat{e}^{(m+1)} = \mathbf{A}_0\widehat{e}^{(m)} + \widehat{\varepsilon}^{(m+1)} + O\left(N^{-1/2}\right). \tag{51}$$

The operator $\mathbf{A}_0$ is termed the Noise Propagation (NP) operator.

Two important properties of fluctuation and noise are worth mentioning:

$$E\left[\widehat{\varepsilon}^{(p)}\widehat{e}^{(q)}\right] = 0, \quad p > q, \tag{52}$$

$$E\left[\widehat{\varepsilon}^{(p)}\widehat{\varepsilon}^{(q)}\right] = 0, \quad p > q. \tag{53}$$

Eq. (52) can be derived by considering for cycles $p > q$. First,

$$E\left[\widehat{\varepsilon}^{(p)}\widehat{e}^{(q)}\right] = E\left[E\left[\widehat{\varepsilon}^{(p)}\widehat{e}^{(q)}|\widehat{e}^{(p-1)}, \widehat{e}^{(q)}\right]\right] \tag{54}$$

by a theorem concerning conditional expectation in probability theory [9]. Since $\widehat{e}^{(q)}$ is fixed in the conditional expectation, it can be pulled out:

$$E\left[E\left[\widehat{\varepsilon}^{(p)}\widehat{e}^{(q)}|\widehat{e}^{(p-1)}, \widehat{e}^{(q)}\right]\right] = E\left[\widehat{e}^{(q)}E\left[\widehat{\varepsilon}^{(p)}|\widehat{e}^{(p-1)}, \widehat{e}^{(q)}\right]\right]. \tag{55}$$

Since Eqs. (27) and (28) imply that $\widehat{\varepsilon}^{(p)}$ is the noise in cycle p generated after fixing $\widehat{e}^{(p-1)}$ ($\widehat{S}^{(p-1)}$) and the rules for population normalization and particle tracking, which govern the law of the noise, are the same for all cycles, the conditioning on $\widehat{e}^{(q)}$ $(q \leqslant p-1)$ can be dropped in Eq. (55):

$$E\left[\widehat{e}^{(q)}E\left[\widehat{\varepsilon}^{(p)}|\widehat{e}^{(p-1)}, \widehat{e}^{(q)}\right]\right] = E\left[\widehat{e}^{(q)}E\left[\widehat{\varepsilon}^{(p)}|\widehat{e}^{(p-1)}\right]\right]. \tag{56}$$

Apply Eq. (31) to finally obtain

$$E\left[\widehat{\varepsilon}^{(p)}\widehat{e}^{(q)}\right] = E\left[\widehat{e}^{(q)}E\left[\widehat{\varepsilon}^{(p)}|\widehat{e}^{(p-1)}\right]\right] = 0, \quad p > q. \tag{57}$$

Eq. (53) is derived similarly:

$$E\left[\widehat{\varepsilon}^{(p)}\widehat{\varepsilon}^{(q)}\right] = E\left[E\left[\widehat{\varepsilon}^{(p)}\widehat{\varepsilon}^{(q)}|\widehat{e}^{(p-1)}, \widehat{\varepsilon}^{(q)}\right]\right] = E\left[\widehat{\varepsilon}^{(q)}E\left[\widehat{\varepsilon}^{(p)}|\widehat{e}^{(p-1)}, \widehat{\varepsilon}^{(q)}\right]\right] = E\left[\widehat{\varepsilon}^{(q)}E\left[\widehat{\varepsilon}^{(p)}|\widehat{e}^{(p-1)}\right]\right] = 0, \quad p > q. \tag{58}$$

3. Eigenvalues of the noise propagation operator

The properties of the NP operator $\mathbf{A}_0$ are discussed in terms of the eigenfunctions S_j of Eq. (7), since the source fluctuation $\vec{e}^{(m)}$ can be expanded as

$$\widehat{e}^{(m)}(\vec{r}) = \widehat{\alpha}_0^{(m)}S_0(\vec{r}) + \widehat{\alpha}_1^{(m)}S_1(\vec{r}) + \widehat{\alpha}_2^{(m)}S_2(\vec{r}) + \cdots \tag{59}$$

where $\widehat{\alpha}_j^{(m)}$ are regarded as stochastically realized constants. Two cases are examined: $\mathbf{A}_0S_0$ and $\mathbf{A}_0S_j, j \geqslant 1$. In the first case,

$$[\mathbf{A}_0S_0](\vec{r}) = \frac{1}{k_0}\int [H(\vec{r}' \to \vec{r}) - S_0(\vec{r})]S_0(\vec{r}')\,dV' = \frac{1}{k_0}\left[\int H(\vec{r}' \to \vec{r})S_0(\vec{r}')\,dV' - \int S_0(\vec{r})S_0(\vec{r}')\,dV'\right] = S_0(\vec{r}) - S_0(\vec{r}) = 0, \tag{60}$$

where Eqs. (7) and (8) were used at the third equality. In the second case,

$$\begin{aligned}[\mathbf{A}_0S_j](\vec{r}) &= \frac{1}{k_0}\int [H(\vec{r}' \to \vec{r}) - S_0(\vec{r})]S_j(\vec{r}')\,dV' = \frac{1}{k_0}\int H(\vec{r}' \to \vec{r})S_j(\vec{r}')\,dV' - \frac{1}{k_0}\int S_0(\vec{r})S_j(\vec{r}')\,dV' \\ &= \frac{k_j}{k_0}S_j(\vec{r}) - \frac{S_0(\vec{r})}{k_0}\int S_j(\vec{r}')\,dV',\end{aligned} \tag{61}$$

where Eq. (7) was used at the third equality. Depending on the integration of the jth eigenfunction over the domain (whether or not it is zero), we have one of two possibilities for Eq. (61):

$$\left[\mathbf{A}_0 S_j\right](\vec{r}) = \frac{k_j}{k_0} S_j(\vec{r}) \quad \text{if} \quad \int_V S_j(\vec{r})\, dV = 0, \quad j \geqslant 1 \tag{62}$$

$$\left[\mathbf{A}_0 S_j\right](\vec{r}) = \mathbf{A}_0 \left[S_j(\vec{r}) - S_0(\vec{r})\right] = \frac{k_j}{k_0} \left[S_j(\vec{r}) - S_0(\vec{r})\right] \quad \text{if} \quad \int_V S_j(\vec{r})\, dV \neq 0, \quad j \geqslant 1, \tag{63}$$

where in Eq. (63), Eq. (60) was used at the first equality and Eq. (9) at the second equality. According to Eq. (60), the fundamental mode eigenfunction $S_0(\vec{r})$ is mapped identically to zero. This indicates that the fluctuation associated with the fundamental mode is always corrected by population normalization. Eqs. (62) and (63) combined with Eq. (51) imply that the eigenvalues of NP process are $k_i/k_0, i = 1, 2, \ldots$, if the number of particles per cycle is sufficiently large.

4. Discrete representation

We quickly cover the discrete representation of the MC NP process corresponding to Eqs. (51), (12), (30), (52) and (53), assuming N is sufficiently large:

$$\vec{e}^{(m+1)} = \mathbf{A}_0 \vec{e}^{(m)} + \vec{\varepsilon}^{(m+1)}, \tag{64}$$

$$E\left[\vec{e}^{(m)}\right] = 0, \tag{65}$$

$$E\left[\vec{\varepsilon}^{(m)}\right] = 0, \tag{66}$$

$$E\left[\vec{\varepsilon}^{(m)} \otimes \vec{e}^{(n)}\right] = E\left[\vec{\varepsilon}^{(m)} (\vec{e}^{(n)})^T\right] = 0, \quad m > n, \tag{67}$$

$$E\left[\vec{\varepsilon}^{(m)} \otimes \vec{\varepsilon}^{(n)}\right] = E\left[\vec{\varepsilon}^{(m)} (\vec{\varepsilon}^{(n)})^T\right] = 0, \quad m > n. \tag{68}$$

In these equations, $\vec{e}^{(m)}$ and $\vec{\varepsilon}^{(m)}$ are $p \times 1$ matrices (column vectors with p entries), $\mathbf{A}_0$ is the NP matrix of size $p \times p$ corresponding to the operator in Eq. (51), and $\otimes$ signifies an outer product (T in superscript signifies a transpose). The number of entries (p) stands for the number of spatial bins where the source distribution is tallied. Hereafter, $\mathbf{A}_0$ is to be understood as matrix if it appears with the vector notation with arrow.

The reason for spatially discretized representation is stated as follows. "If the eigenvalue sought after is extracted statistically at the final step of calculation, the answer does not suffer from discretization error in the same sense as deterministic approaches. The critical issue of statistical extraction is not the size of bins but the cancellation of eigenmodes." Strictly speaking, there is no mathematical proof of this statement in the nuclear engineering discipline. However, it was demonstrated in previous work [3] for the dominant eigenvalue calculation with two bins, which agreed with Green's function benchmark results. In addition, if one surveys various physical sciences experiments, the parameter determination by feature extraction with signal processing techniques is found to be very popular. Therefore, our guiding principle is the effective cancellation of the eigenmodes other than the one sought after. To this end, we have developed a projection technique for creating one dimensional time series that exclusively contains the fluctuating component associated with the eigenmode sought after. This is termed Coarse Mesh Projection Method (CMPM) and is described in next section.

5. Coarse Mesh Projection Method

The steps of CMPM are as follows: (1) compute the NP matrix $\mathbf{A}_0$ and its eigenvectors, (2) apply the ith eigenvector to the source fluctuation vector $\vec{e}^{(m)}$ in order to produce a one dimensional time series if the ith mode eigenvalue is sought after, and (3) perform an AR(1) fitting to the resulting time series and calculate the autocorrelation coefficient. As will be shown, this coefficient is the desired eigenvalue ratio, i.e., k_i/k_0 from Eq. (7).

5.1. Noise propagation matrix

The first task is to obtain the formal representation of $\mathbf{A}_0$. First, Eq. (64) is multiplied throughout on the right by $\vec{e}^{(m)}$ as an outer product yielding

$$\vec{e}^{(m+1)} \otimes \vec{e}^{(m)} = \mathbf{A}_0 \vec{e}^{(m)} \otimes \vec{e}^{(m)} + \vec{\varepsilon}^{(m+1)} \otimes \vec{e}^{(m)}. \tag{69}$$

Then, taking the expectation and using Eq. (67), we obtain

$$E\left[\vec{e}^{(m+1)} \otimes \vec{e}^{(m)}\right] = \mathbf{A}_0 E\left[\vec{e}^{(m)} \otimes \vec{e}^{(m)}\right]. \tag{70}$$

Defining a general cross covariance matrix as

$$\mathbf{L}_i \equiv E\left[\vec{e}^{(m+i)} \otimes \vec{e}^{(m)}\right]. \tag{71}$$

$\mathbf{A}_0$ can be expressed as

$$\mathbf{A}_0 = \mathbf{L}_1 \mathbf{L}_0^{-1}. \tag{72}$$

Therefore, $\mathbf{A}_0$ can be evaluated via the computation of the sample lag zero and one cross covariance matrices.

5.2. Projection process

The eigenvalue problem of $\mathbf{A}_0$ is

$$\mathbf{A}_0 \vec{b}_i = \lambda_i \vec{b}_i, \quad i = 1, \ldots, p, \tag{73}$$

$$\mathbf{A}_0^* \vec{d}_j = \lambda_j^* \vec{d}_j, \quad j = 1, \ldots, p. \tag{74}$$

Here, λ_i corresponds to k_i/k_0 and $\mathbf{A}_0^*$ is the adjoint (or complex conjugate transpose) of $\mathbf{A}_0$. Since $\mathbf{A}_0$ is real, the adjoint is equal to the transpose

$$\mathbf{A}_0^* = \mathbf{A}_0^{\mathbf{T}}. \tag{75}$$

Since $\det(\mathbf{A}_0 - \lambda I) = \det\left(\mathbf{A}_0^T - \lambda I\right)$ (where det is the determinant and I is the identity matrix), $\mathbf{A}_0$ and $\mathbf{A}_0^{\mathbf{T}}$ share the same set of eigenvalues,

$$\lambda_i = \lambda_j^* \quad \text{for } i = j. \tag{76}$$

In addition, their eigenvectors satisfy

$$\left\langle \vec{b}_i, \vec{d}_j \right\rangle = 0 \quad \text{if } \lambda_i \neq \lambda_j^* \text{ and } i \neq j, \tag{77}$$

where $\langle \cdot, \cdot \rangle$ indicates an inner product (or dot product) of the two column vectors. In other notation, this is equivalent to $\left\langle \vec{d}, \vec{b} \right\rangle = \vec{d}^{\mathbf{T}} \vec{b}$. The proof of Eq. (77) is shown below. With Eqs. (75) and (74) becomes

$$\mathbf{A}_0^{\mathbf{T}} \vec{d}_j = \lambda_j \vec{d}_j, \quad 1 \leqslant j \leqslant p. \tag{78}$$

Taking the transpose of each side of (78) yields

$$\vec{d}_j^{\mathbf{T}} \mathbf{A}_0 = \lambda_j \vec{d}_j^{\mathbf{T}}. \tag{79}$$

First, apply $\vec{d}_j^{\mathbf{T}}$ to Eq. (73) from left to obtain

$$\vec{d}_j^{\mathbf{T}} \left(\mathbf{A}_0 \vec{b}_i \right) = \lambda_i \vec{d}_j^{\mathbf{T}} \vec{b}_i. \tag{80}$$

Next, apply $\vec{b}_i$ to Eq. (79) from right to obtain

$$\left(\vec{d}_j^{\mathbf{T}} \mathbf{A}_0 \right) \vec{b}_i = \lambda_j \vec{d}_j^{\mathbf{T}} \vec{b}_i. \tag{81}$$

Since the LHSs of Eqs. (80) and (81) are equal, one obtains

$$0 = (\lambda_j - \lambda_i) \vec{d}_j^{\mathbf{T}} \vec{b}_i \tag{82}$$

implying that either $\lambda_i = \lambda_j$ or $\vec{d}_j^{\mathbf{T}} \vec{b}_i = 0$.

If the eigenvector of the transpose of NP matrix, which is $\vec{d}_i$, is applied to Eq. (64) and then Eq. (79) is applied, we obtain

$$\left\langle \vec{d}_i, \vec{e}^{(m+1)} \right\rangle = \lambda_i \left\langle \vec{d}_i, \vec{e}^{(m)} \right\rangle + \left\langle \vec{d}_i, \vec{\varepsilon}^{(m+1)} \right\rangle. \tag{83}$$

Defining the components of a one dimensional time series as

$$y^{(m)} \equiv \left\langle \vec{d}_i, \vec{e}^{(m)} \right\rangle, \tag{84}$$

$$z^{(m)} \equiv \left\langle \vec{d}_i, \vec{\varepsilon}^{(m)} \right\rangle. \tag{85}$$

$y^{(m)}$ satisfies

$$y^{(m+1)} = \lambda_i y^{(m)} + z^{(m+1)}. \tag{86}$$

In addition, using Eqs. (65), (66) and (68), we obtain

$$E\left[y^{(m)}\right] = E\left[\left\langle \vec{d}_i, \vec{e}^{(m)} \right\rangle\right] = \left\langle \vec{d}_i, E\left[\vec{e}^{(m)}\right] \right\rangle = 0, \tag{87}$$

$$E\left[z^{(m)}\right] = E\left[\left\langle \vec{d}_i, \vec{\varepsilon}^{(m)} \right\rangle\right] = \left\langle \vec{d}_i, E\left[\vec{\varepsilon}^{(m)}\right] \right\rangle = 0, \tag{88}$$

$$E\left[z^{(m)} z^{(n)}\right] = E\left[\left\langle \vec{d}_i, \vec{\varepsilon}^{(m)} \right\rangle \left\langle \vec{d}_i, \vec{\varepsilon}^{(n)} \right\rangle\right] = \left\langle \vec{d}_i \otimes \vec{d}_i, E\left[\vec{\varepsilon}^{(m)} \otimes \vec{\varepsilon}^{(n)}\right] \right\rangle = 0, \quad m > n. \tag{89}$$

In Eq. (89), $\left\langle \vec{d}_i, \vec{\varepsilon}^{(m)} \right\rangle \left\langle \vec{d}_i, \vec{\varepsilon}^{(n)} \right\rangle = \left\langle \vec{d}_i \otimes \vec{d}_i, \vec{\varepsilon}^{(m)} \otimes \vec{\varepsilon}^{(n)} \right\rangle$ was also used.

5.3. Time series autoregressive process

To solve for the coefficient λ_i in Eq. (86), which is the eigenvalue in Eqs. (73)–(76) and corresponds to the eigenvalue ratio k_i/k_0, $y^{(m)}$ is multiplied throughout and the expectation is taken:

$$E\left[y^{(m+1)} y^{(m)}\right] = \lambda_i E\left[y^{(m)} y^{(m)}\right] + E\left[z^{(m+1)} y^{(m)}\right]. \tag{90}$$

Using Eq. (67), the last term on the RHS becomes zero:

$$E\left[z^{(m+1)} y^{(m)}\right] = E\left[\left\langle \vec{d}_i, \vec{\varepsilon}^{(m+1)} \right\rangle \left\langle \vec{d}_i, \vec{e}^{(m)} \right\rangle\right] = E\left[\left\langle \vec{d}_i \otimes \vec{d}_i, \vec{\varepsilon}^{(m+1)} \otimes \vec{e}^{(m)} \right\rangle\right] = \left\langle \vec{d}_i \otimes \vec{d}_i, E\left[\vec{\varepsilon}^{(m+1)} \otimes \vec{e}^{(m)}\right] \right\rangle = 0. \tag{91}$$

Eq. (90) then reduces to

$$E\left[y^{(m+1)} y^{(m)}\right] = \lambda_i E\left[y^{(m)} y^{(m)}\right] \tag{92}$$

and the coefficient λ_i can be solved for as

$$\lambda_i = \frac{E\left[y^{(m+1)} y^{(m)}\right]}{E\left[y^{(m)} y^{(m)}\right]}. \tag{93}$$

Eq. (93) highlights the significance of CMPM. Simple AR(1) fitting is enough for the computation of k_i/k_0; in other words, k_i/k_0 is obtained as autocorrelation coefficient. Since $|\lambda_i| < 1, i \geqslant 1$ under the stationarity assumption which traces back to Eq. (10), the AR(1) fitting is guaranteed to be stable [10]. The method requires no input other than the standard input for MC calculation for nuclear criticality and reactor analysis.

5.4. Error estimation

The variance of the lag k autocorrelation coefficient r_k of an AR(1) process driven by normal noise is given by [10]

$$\mathrm{var}[r_k] = \frac{1}{M}\left(\frac{(1+\phi^2)(1-\phi^{2k})}{1-\phi^2} - 2k\phi^{2k}\right), \tag{94}$$

where ϕ is the lag 1 autocorrelation coefficient and M is the number of stationary cycles. Since the eigenvalue ratio (k_i/k_0) corresponds to the lag 1 autocorrelation coefficient (λ_i) of the AR(1) process of Eq. (86), the error of k_i/k_0 is estimated as

$$\sqrt{\mathrm{var}[\lambda_i]} = \sqrt{\frac{1}{M}(1-\lambda_i^2)}. \tag{95}$$

Note that one can assume normality for the elements of $\vec{\varepsilon}^{(m)}$ if the number of particles per cycle (N) is sufficiently large. Practically, Eq. (95) suggests the increase of the number of active (stationary) cycles instead of the number of particles per cycle if the latter is sufficiently large. Also, Eq. (95) enables one to compute the standard deviation of an estimate of λ_i from a single run of Monte Carlo calculation.

6. Theory for practical implementation issues

There are three main steps in CMPM: (1) determination of an appropriate projection vector, (2) application of the projection vector to the source fluctuation, (3) time series analysis of the projected series. Though it is a straightforward process, specific considerations when implementing this method into production codes have not yet been discussed. In particular, there is a strong desire to reduce memory usage. Applied in a rudimentary way, the source distribution from every bin and every cycle must be stored throughout the run. After the run is complete, the source fluctuations are computed via the subtraction of the mean source distribution and the eigenvectors of the noise propagation matrix are computed and then applied to the source fluctuation. Saving all of this data throughout the run is an enormous task; the number of storage elements required for this implementation is (p source bins) × (M active cycles). It is very common for there to be thousands of active cycles per run, so it is easy to see how this number can quickly become unmanageable. Also, there is a strong demand on the code developer side for the incremental updating of λ_i along the actual progression of cycles. Therefore, we derive alternative expressions of λ_i and $\mathbf{A_0}$. To this end, the calculation of $\mathbf{A_0}$ using the actual source distribution will be derived,

from which the equivalent projection vectors are calculated. Next, the calculation of λ_i using the actual source distribution will be derived.

We rewrite Eq. (64) as follows. Let $\vec{S}_0$ be the true mean of source distribution in discrete representation. The corresponding relations to Eqs. (8) and (60) are

$$(1,1,\ldots,1)\vec{S}_0 = k_0, \tag{96}$$

$$\mathbf{A}_0\vec{S}_0 = 0. \tag{97}$$

Using Eq. (97), $\mathbf{A}_0 N\vec{S}_0$ is added to the RHS of Eq. (64) multiplied by $\sqrt{N}$ to obtain

$$\sqrt{N}\vec{e}^{(m+1)} = \mathbf{A}_0\left(N\vec{S}_0 + \sqrt{N}\vec{e}^{(m)}\right) + \sqrt{N}\vec{\varepsilon}^{(m+1)} = \mathbf{A}_0\vec{S}^{(m)} + \sqrt{N}\vec{\varepsilon}^{(m+1)}, \tag{98}$$

where $\vec{S}^{(m)} = N\vec{S}_0 + \sqrt{N}\vec{e}^{(m)}$ corresponding to Eq. (10). Now $N\vec{S}_0$ is added to both sides of Eq. (98), yielding

$$\vec{S}^{(m+1)} = \mathbf{A}_0\vec{S}^{(m)} + \vec{\eta}^{(m+1)}, \tag{99}$$

where $\vec{\eta}^{(m+1)} = N\vec{S}_0 + \sqrt{N}\vec{\varepsilon}^{(m+1)}$. Applying $\vec{S}^{(m)}$ to both sides on the right as an outer product and taking the expectation results in

$$E\left[\vec{S}^{(m+1)}\otimes\vec{S}^{(m)}\right] = \mathbf{A}_0 E\left[\vec{S}^{(m)}\otimes\vec{S}^{(m)}\right] + E\left[\vec{\eta}^{(m+1)}\otimes\vec{S}^{(m)}\right], \tag{100}$$

$$E\left[\vec{S}^{(m+1)}\otimes\vec{S}^{(m)}\right] = \mathbf{A}_0 E\left[\vec{S}^{(m)}\otimes\vec{S}^{(m)}\right] + E\left[\left(\sqrt{N}\vec{\varepsilon}^{(m+1)} + N\vec{S}_0\right)\otimes\left(\sqrt{N}\vec{e}^{(m)} + N\vec{S}_0\right)\right], \tag{101}$$

$$E\left[\vec{S}^{(m+1)}\otimes\vec{S}^{(m)}\right] = \mathbf{A}_0 E\left[\vec{S}^{(m)}\otimes\vec{S}^{(m)}\right] + N^2\vec{S}_0\otimes\vec{S}_0. \tag{102}$$

Here, Eqs. (65)–(67) are used to arrive at Eq. (102). Solving for $\mathbf{A}_0$ yields

$$\mathbf{A}_0 = \left\{E\left[\vec{S}^{(m+1)}\otimes\vec{S}^{(m)}\right] - N^2\vec{S}_0\otimes\vec{S}_0\right\}\left\{E\left[\vec{S}^{(m)}\otimes\vec{S}^{(m)}\right]\right\}^{-1}. \tag{103}$$

Since $N^2\vec{S}_0\otimes\vec{S}_0$ is estimated as the sample mean of $\vec{S}^{(m)}\otimes\vec{S}^{(m)}$, this becomes

$$\mathbf{A}_0 = E\left[\vec{S}^{(m+1)}\otimes\vec{S}^{(m)}\right]\left\{E\left[\vec{S}^{(m)}\otimes\vec{S}^{(m)}\right]\right\}^{-1} - \mathbf{I}. \tag{104}$$

where $\mathbf{I}$ is the identity matrix. Therefore, the NP matrix $\mathbf{A}_0$ and the projection vector $\vec{d}_i$ can be calculated using the actual source distribution without the subtraction of sample mean source distribution.

Next, an expression of λ_i other than Eq. (93) will be derived using only the source distributions. Multiplying Eq. (99) by $\vec{d}_i^T$ on the left and using Eq. (79), a new times series can be written as

$$y^{(m+1)} = \lambda_i y^{(m)} + z^{(m+1)}, \tag{105}$$

$$y^{(m)} = \left\langle\vec{d}_i,\vec{S}^{(m)}\right\rangle, \tag{106}$$

$$z^{(m)} = \left\langle\vec{d}_i,\vec{\eta}^{(m)}\right\rangle. \tag{107}$$

To solve for the autocorrelation coefficient, Eq. (105) is multiplied by $y^{(m)}$ throughout and the expectation is taken:

$$E\left[y^{(m+1)}y^{(m)}\right] = \lambda_i E\left[y^{(m)}y^{(m)}\right] + E\left[z^{(m+1)}y^{(m)}\right]. \tag{108}$$

In this case, the last term in (108) is

$$E\left[z^{(m+1)}y^{(m)}\right] = E\left[\left\langle\vec{d}_i,\vec{\eta}^{(m+1)}\right\rangle\left\langle\vec{d}_i,\vec{S}^{(m)}\right\rangle\right] = E\left[\left\langle\vec{d}_i\otimes\vec{d}_i,\vec{\eta}^{(m+1)}\otimes\vec{S}^{(m)}\right\rangle\right] = \left\langle\vec{d}_i\otimes\vec{d}_i,E\left[\vec{\eta}^{(m+1)}\otimes\vec{S}^{(m)}\right]\right\rangle. \tag{109}$$

Using Eqs. (65)–(67), the expectation term inside the inner product is simply

$$E\left[\vec{\eta}^{(m+1)}\otimes\vec{S}^{(m)}\right] = E\left[\left(N\vec{S}_0 + \sqrt{N}\vec{\varepsilon}^{(m+1)}\right)\otimes\left(N\vec{S}_0 + \sqrt{N}\vec{e}^{(m)}\right)\right] = N^2\vec{S}_0\otimes\vec{S}_0 \tag{110}$$

reducing Eq. (109) to

$$E\left[z^{(m+1)}y^{(m)}\right] = \left\langle\vec{d}_i\otimes\vec{d}_i,N^2\vec{S}_0\otimes\vec{S}_0\right\rangle = N^2\left\langle\vec{d}_i,\vec{S}_0\right\rangle\left\langle\vec{d}_i,\vec{S}_0\right\rangle = 0, \tag{111}$$

because

$$\lambda_i\left\langle\vec{d}_i,\vec{S}_0\right\rangle = \left\langle\lambda_i\vec{d}_i,\vec{S}_0\right\rangle = \left\langle\mathbf{A}_0^{\mathbf{T}}\vec{d}_i,\vec{S}_0\right\rangle = \left\langle\vec{d}_i,\mathbf{A}_0\vec{S}_0\right\rangle = 0, \tag{112}$$

where Eq. (78) was used at the second step and Eq. (97) was used at the last step. Thus, Eq. (108) can be manipulated as

$$\left\langle \vec{d}_i \otimes \vec{d}_i, E\left[\vec{S}^{(m+1)} \otimes \vec{S}^{(m)}\right] \right\rangle = E\left[\left\langle \vec{d}_i, \vec{S}^{(m+1)} \right\rangle \left\langle \vec{d}_i, \vec{S}^{(m)} \right\rangle\right] = E\left[y^{(m+1)} y^{(m)}\right] = \lambda_i E\left[y^{(m)} y^{(m)}\right] = \lambda_i E\left[\left\langle \vec{d}_i, \vec{S}^{(m)} \right\rangle \left\langle \vec{d}_i, \vec{S}^{(m)} \right\rangle\right]$$
$$= \lambda_i \left\langle \vec{d}_i \otimes \vec{d}_i, E\left[\vec{S}^{(m)} \otimes \vec{S}^{(m)}\right] \right\rangle. \tag{113}$$

Hence, λ_i, which is the eigenvalue in Eqs. (73)–(76) and corresponds to the eigenvalue ratio k_i/k_0, has other expression using time series output:

$$\lambda_i = \frac{\left\langle \vec{d}_i \otimes \vec{d}_i, E\left[\vec{S}^{(m+1)} \otimes \vec{S}^{(m)}\right] \right\rangle}{\left\langle \vec{d}_i \otimes \vec{d}_i, E\left[\vec{S}^{(m)} \otimes \vec{S}^{(m)}\right] \right\rangle}. \tag{114}$$

Eqs. (104) and (114) show that the sample mean subtraction for the computation of the source distribution fluctuation is unnecessary for the computation of λ_i, i.e., k_i/k_0. In addition, only the source distributions at the most recent two cycles need to be stored along with the actual progression of cycles. Practically, the use of Eqs. (104) and (114) reduces storage requirement and enables one to update λ_i in an on-the-fly manner via the incremental updating of $E\left[\vec{S}^{(m+1)} \otimes \vec{S}^{(m)}\right], E\left[\vec{S}^{(m)} \otimes \vec{S}^{(m)}\right]$ and $\vec{d}_i$.

7. Problem descriptions

The results in the subsequent sections were all generated based on four problems that are presented in this section. They are each fully described here. Problem 1 is a one-energy group, multi-region 1-D slab with vacuum boundary conditions. The make-up of Problem 1 is illustrated in Fig. 1. There are two fuel (fissile) regions on either end of the slab with scattering and absorbing material between them. This type of problem would most likely be found in criticality safety work. The first four eigenvalue ratios of this problem (considered the benchmark) were computed using the Green's Function Method (GFM) [11] with a 1800-bin mesh across the entire domain and are shown in Table 1. The effective neutron multiplication factor (k_0) was calculated to be 0.424314 by GFM and 0.424314 ± 0.000007 by MC.

Problem 2 is a 2-D checkerboard illustrated in Fig. 2. There are two types of fuel placed alternately in a checkerboard manner making the problem symmetric along the diagonals. The first eigenvalue ratio (k_1/k_0), also called dominance ratio (DR), was estimated to be 0.9581 by the analysis of the spectral radius of outer iterations in discontinuous finite element discrete ordinates methods [12] as shown in Table 2. The effective neutron multiplication factor was calculated to be 1.05450 by the discrete ordinates method and 1.05450 ± 0.00001 by MC. The higher order eigenvalues were not available by the spectral radius analysis. Therefore, they were computed by the Fission Matrix Method (FMM) [13] using the discrete ordinates method with a 2304-bin mesh (48 bins in each coordinate direction) instead, as shown in Table 3.

Problem 3 is the two-dimensional version of an initial core pressurized water reactor (PWR) problem with continuous-energy cross-sections [14]. The problem is illustrated in Fig. 3; the computation was run using reflecting top and bottom boundaries. The effective neutron multiplication factor is 1.0187 ± 0.0004 (1 σ) using the continuous-energy cross section data. Only DR was available for this problem, as shown in Table 4. It was calculated using the ARMA(2,1) Half-Domain Fitting Method (HDFM) [3].

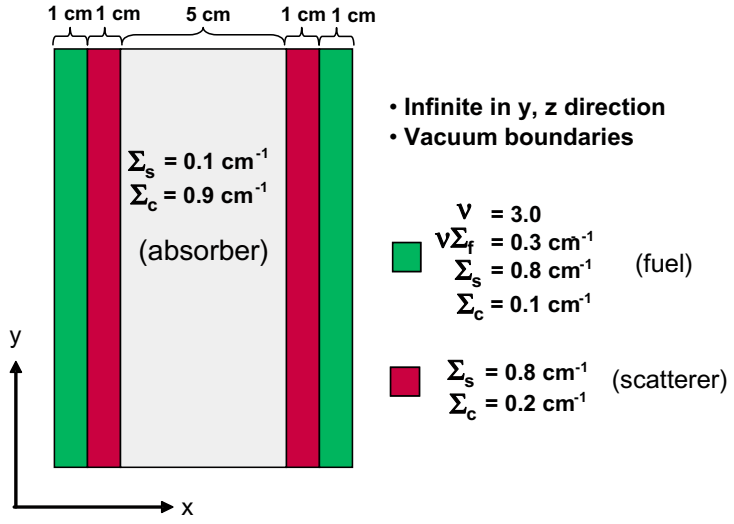

Fig. 1. Problem 1 – 1D heterogeneous slab.

Table 1
First four benchmark eigenvalue ratios of Problem 1.

	GFM using 1800-bin mesh
k_1/k_0	0.999565
k_2/k_0	0.304653
k_3/k_0	0.304635
k_4/k_0	0.167738

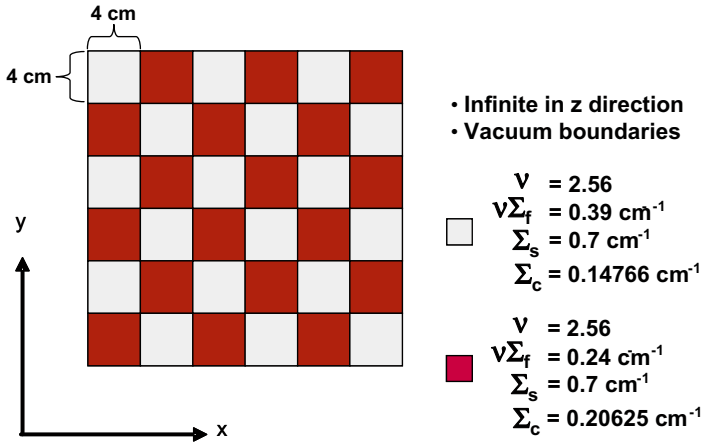

Fig. 2. Problem 2 – 2D checkerboard.

Table 2
Benchmark DR of Problem 2.

	Benchmark by discrete ordinates
k_1/k_0	0.9581

Table 3
First four eigenvalue ratios of Problem 2 using FMM.

	FMM using 2304-bin mesh
k_1/k_0	0.95740
k_2/k_0	0.95710
k_3/k_0	0.92031
k_4/k_0	0.89708

Problem 4 is a three-dimensional version of Problem 3. It is nearly the same as Problem 3 except that it includes vertical modeling of the plenum, top and bottom end plugs inside Zirconium cladding, and the top and bottom support structures. The detailed specifications are found in [14]. The effective neutron multiplication factor is 1.01249 ± 0.00055 (1σ) using the continuous-energy cross section data. A benchmark DR was calculated to be 0.990 ± 0.002 (2σ) by HFDM.

8. CMPM results

Results are presented for the three user-specified parameters that affect the performance of CMPM: the number of particles per cycle, the number of active cycles, and the mesh used to tally the source distribution. In each case, a "sufficient" parameter must be used to obtain accurate results. The definition of sufficiency for each of these parameters will be discussed in depth and results will be shown that illustrate what happens when the conditions are not met.

8.1. Number of particles per cycle

As indicated by the eigenvalue ratio calculation and its error estimate (Eqs. (93) and (95)), the number of particles per cycle does not directly influence these results. However, CMPM does assume that there are a sufficient number of neutrons

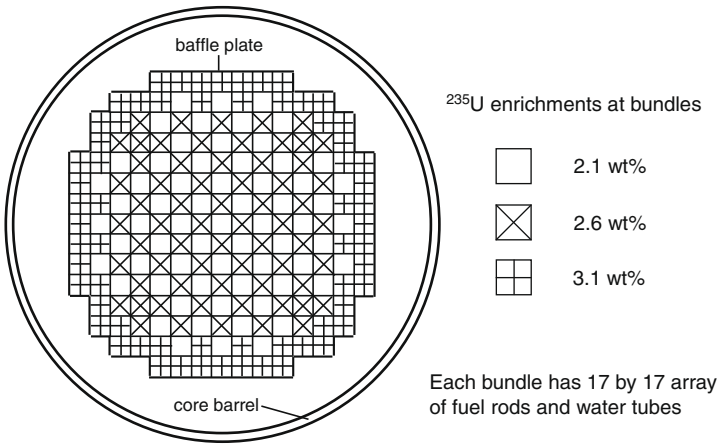

Fig. 3. Problem 3 – 2D PWR initial core.

Table 4
Benchmark DR of Problem 3.

	Benchmark by ARMA(2,1) HDFM
k_1/k_0	0.9927 ± 0.0017 (2σ)

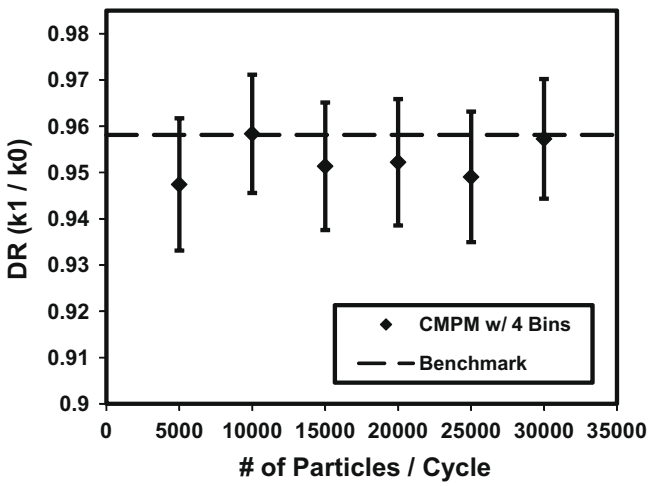

Fig. 4. Effect of particles per cycle on CMPM (Problem 2).

causing fission events in each individual tally bin, such that every binned source can also be assumed Normal. For this reason, the number of particles per cycle should be at least as large as the standard MC calculations for reactor analysis.

To prove this point that CMPM results are independent of particle population, Problem 2 was tested using varying numbers of particles per cycle. Five hundred active cycles were used as well as an appropriate source tally mesh. Results are presented in Fig. 4 with 2σ error bars. As can be seen, the accuracy and error bounds are consistent, illustrating the independence of CMPM of the number of particles per cycle and justifying the use of Eq. (95) for error estimation. Similar observations were made when testing other problems in this manner.

8.2. Number of active cycles

Since CMPM is a methodology based on time series, it can be strongly influenced by cycle correlation. The stronger the correlation, the more cycles are necessary to obtain the accurate representation of a particular eigenmode fluctuation. Without enough cycles, the results will be biased, often an underestimating of the eigenvalue. To illustrate this, Problem 1 was run using 30,000 particles per cycle and varying numbers of active cycles. Results are shown in Fig. 5 with 1σ error bars.

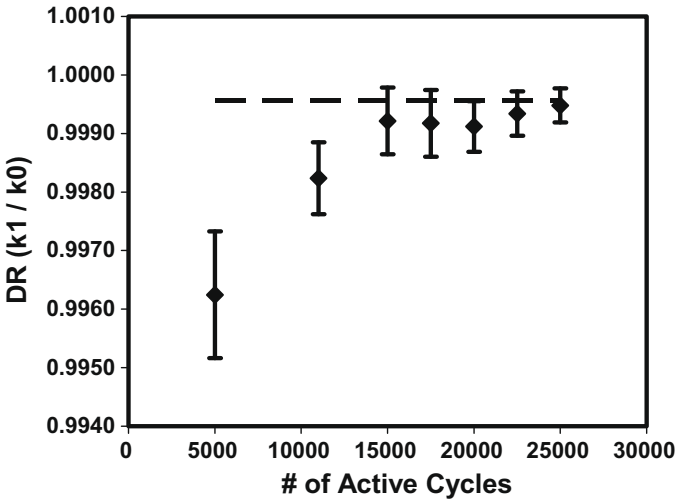

Fig. 5. Effect of # of active cycles on CMPM (Problem 1).

This trend of underestimation was observed in all problems tested, though the rate of convergence to an unbiased estimate varied. A relation has been developed to provide the user with a way of estimating the necessary number of active cycles for each problem. The decay of the correlation between cycles is upper-bounded by the exponential of DR (k_1/k_0), i.e., correlation strength $= \lambda_1^{\eta_{AC}}$, where η_{AC} is the number of active cycles [15]. We impose that there either be a minimum of 500 active cycles or less than a 0.1% correlation between the first and last cycles to ensure unbiased results. If this is the case, the minimum number of active cycles necessary to obtain unbiased results is estimated as

$$\eta_{AC} = \max[500, \ln(10^{-3})/\ln(\lambda_1)]. \tag{115}$$

If the value of k_1/k_0 in Table 1 is substituted in λ_1 in Eq. (115), $\eta_{AC} = 15,876$ is obtained. This agrees with Fig. 1. The practical utility of Eq. (115) is to check if the number of active cycles actually run is larger than the RHS with the output of CMPM substituted in λ_1.

8.3. Tally mesh of the fission source distribution

Time series methods analyze fluctuations of binned source distribution and, therefore, do not suffer from discretization error in the same sense as posing final answers directly from solving the NP matrix as discussed in Section 4. However, there are two requirements that must be met by the binning scheme: (1) the fluctuation being analyzed must exclusively contain the desired eigenmode, and (2) the binning scheme must minimize the desired eigenmode cancellations in each bin due to the simplicity of the AR(1) fitting. Here, the desired eigenmode is the jth mode if k_j/k_0 is sought after. The first requirement is met by using the eigenvector of the transposed NP matrix, guaranteeing that the effects of other modes are removed. Optimally, the second requirement can be met by ensuring that each bin boundary is set where the eigenmode changes sign.

Unfortunately, the eigenmode structure for an arbitrary problem cannot be known a priori. Despite this, several observations can be made so that the cancellations for the first several eigenvalues can be minimized while still using a coarse mesh that is easy to manipulate and solve. Consider a simple one-dimensional problem. The nth eigenfunction S_n changes sign n times over the domain. This means that *optimally* only two tally bins are needed to compute DR (λ_1) where the boundary between the two bins is placed at the point where the eigenmode is equal to zero since the corresponding eigenmode only changes sign once across the domain. Only three tally bins are *optimally* needed to compute $k_2/k_0(\lambda_2)$ for a one-dimensional problem, again with the boundaries of the bins placed at the points where the eigenmode is equal to zero. This concept extends to multi-dimensional problems as well. Only four bins (two per coordinate direction) are needed to compute DR (λ_1) for a two-dimensional problem, *optimally* and eight bins for a three-dimensional problem, optimally.

It is worthwhile to estimate how much cancellation of an eigenfunction can occur before erroneous results are obtained. The shape of the eigenfunction corresponding to the k_1 eigenvalue is available for the flux of Problem 1 from the GFM benchmark results and is shown in Fig. 6. It is clear that the eigenfunction changes sign at the middle of the problem. Note that the source distribution corresponds to the subdomain of $0 \leqslant x \leqslant 1$ and $8 \leqslant x \leqslant 9$.

To prevent any cancellations from occurring, the bin boundary of a two-bin mesh should be placed in the center. Instead, an irregular bin scheme was applied. A three-bin mesh is applied over the two fuel regions as shown in Fig. 7. Bins 1 and 3 are the same size. Bin 2 is varied in size to gauge the amount of cancellation that occurs versus accuracy. In equation form, if $\Delta_1, \Delta_2, \Delta_3$ are the widths of bins 1, 2, 3, respectively, then

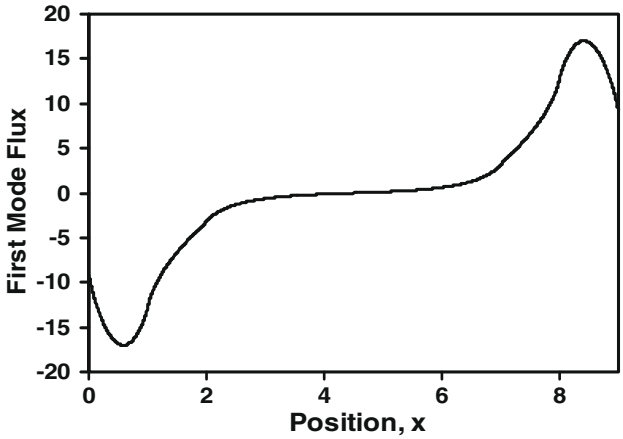

Fig. 6. First non-fundamental mode flux profile (Problem 1).

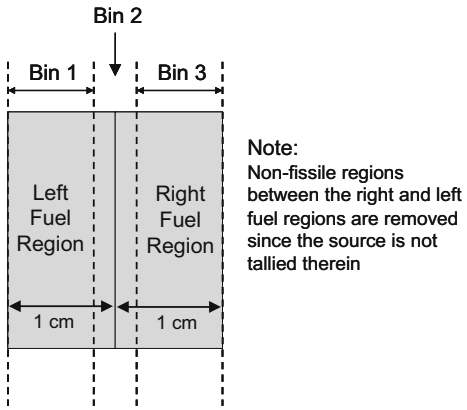

Fig. 7. Mesh scheme for perturbation analysis of Problem 1.

$$\Delta_1 = \Delta_3$$
$$\Delta_1 = 1 - \frac{\Delta_2}{2}.$$

Fig. 8 shows the results of this analysis using 40,000 active cycles and 80,000 particles per cycle with 2σ error bars.

From this result it appears that when more than 20% of the eigenfunction canceled out over Bin 2 (corresponding to >0.4 cm for the size of Bin 2), the results became erroneous. At that point, the estimates were outside of a 2σ confidence interval. Especially with higher eigenmodes that have many changes of sign, a fine mesh scheme could be applied to prevent such a large cancellation of the mode. A stronger fitting (such as ARMA(2,1)) could also be used to improve the estimate, but the goal of CMPM is to keep the fitting method as simple as possible. CMPM relies on the availability of the full eigenmode for the simple AR(1) fitting order. As long as there is minimal cancellation of the mode, the method will produce accurate results, no matter what mesh scheme is applied.

8.4. General results

General results are presented next. In each case, sufficient parameters were used as described in Sections 8.1–8.3. First, Problem 1 was run using a 30-bin mesh (15 bins across each fuel region) using 40,000 active cycles and 80,000 particles per cycle. Table 5 compares the 2σ confidence interval k_1/k_0–k_4/k_0 results of the CMPM to the benchmark GFM results.

As can be seen, the 30-bin mesh is sufficient to accurately compute all four eigenvalue ratios since there is minimal cancellation of the eigenmode. It is worth reemphasizing that a 30-bin mesh is *not* strictly required to accurately compute these first eigenvalue ratios. A coarser mesh could be used, but since the shape of the corresponding eigenmodes are generally unknown, a finer 30-bin mesh is used to prevent significant cancellations. Note that GFM used a 1800 bin mesh.

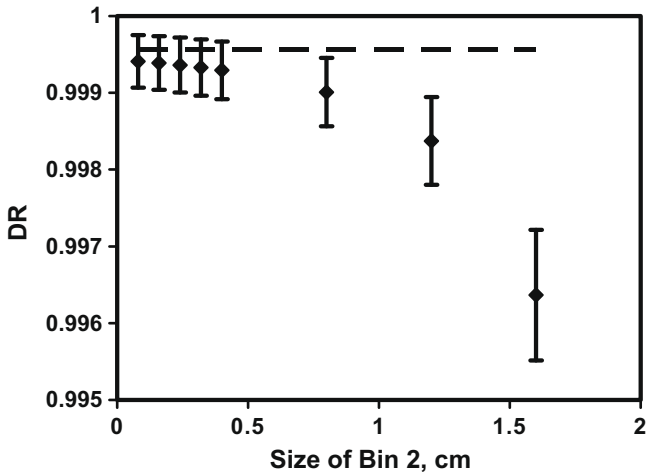

Fig. 8. DR with 2σ standard deviation vs increasing eigenmode cancellation (Problem 1).

Table 5
First four eigenvalue ratios of Problem 1 benchmark vs CMPM with 30-bin mesh and (2σ) standard deviation.

	GFM using 1800-bin mesh	CMPM 2σ interval using 30-bin mesh	Benchmark contained in 2σ?
k_1/k_0	0.999565	(0.999405, 0.999750)	Yes
k_2/k_0	0.304653	(0.292868, 0.311932)	Yes
k_3/k_0	0.304635	(0.285621, 0.304730)	Yes
k_4/k_0	0.167738	(0.161231, 0.180936)	Yes

Table 6
DR of Problem 2 benchmark vs CMPM with minimum 4-bin mesh and 2σ standard deviation.

	Benchmark by discrete ordinates	CMPM 2σ interval using 4-bin mesh	Benchmark contained in 2σ?
k_1/k_0	0.9581	(0.953156, 0.959017)	Yes

Table 7
First four eigenvalue ratios of Problem 2 benchmark vs CMPM with 36-bin mesh and 2σ standard deviation.

	FMM using 2304-bin mesh	CMPM 2σ interval using 36-bin mesh	FMM result contained in 2σ?
k_1/k_0	0.95740	(0.954088, 0.959891)	Yes
k_2/k_0	0.95710	(0.953242, 0.959098)	Yes
k_3/k_0	0.92031	(0.917194, 0.924981)	Yes
k_4/k_0	0.89708	(0.894094, 0.902874)	Yes

Table 8
DR of Problem 3 using MCNP5 benchmark vs CMPM with minimum 4-bin mesh and 2σ standard deviation.

	Benchmark by ARMA(2, 1) HDFM	CMPM 2σ interval using 4-bin mesh	Benchmark contained in 2σ?
k_1/k_0	(0.99092, 0.99438)	(0.98782, 0.99356)	Yes

Next, Problem 2 was run using 40,000 active cycles and 80,000 particles per cycle. Table 6 compares the 2σ confidence interval of k_1/k_0 results of the CMPM using a 4-bin mesh (two bins per coordinate direction) to the benchmark discrete ordinates results. Table 7 compares the 2σ confidence interval $k_1/k_0 - k_4/k_0$ results of the CMPM using a 36-bin mesh to the FMM results using the discrete ordinates method with a 2304-bin mesh. CMPM was able to accurately compute all of the first four eigenvalue ratios within the 2σ interval. It is interesting to note that since the problem is symmetric along the diagonal, it is affected by multiplicity where the first and second non-fundamental eigenvalues (and likewise, the eigenvalue ratios) are theoretically equal. This can often make problems difficult to solve and will be discussed in Section 9.

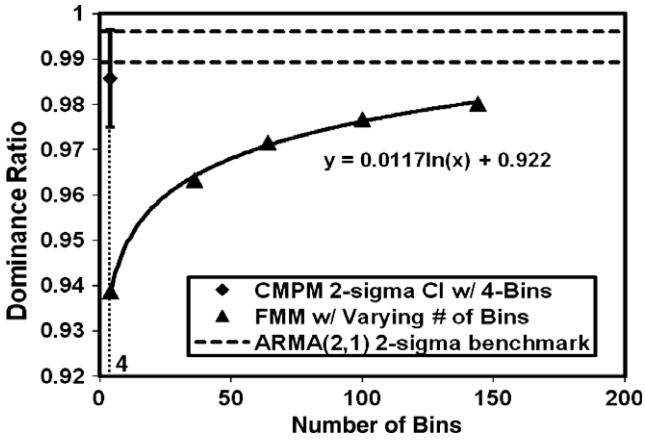

Fig. 9. Problem 3 – 2D PWR comparison FMM vs CMPM.

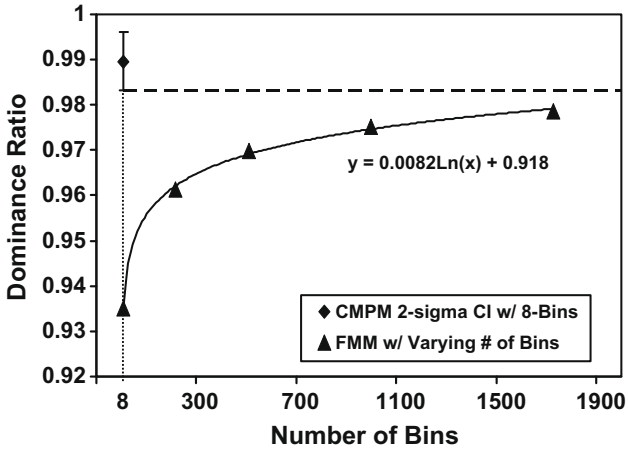

Fig. 10. Problem 4 – 3D PWR comparison FMM vs CMPM.

Problem 3 is considered next. Table 8 compares the benchmark DR computed by the ARMA(2,1) Half-Domain Fitting Method (HDFM) [3] against CMPM using 200,000 particles per cycle, 1000 inactive cycles, 9000 active cycles and a 4-bin mesh. As can be seen, CMPM performs well even for realistic problems.

The final analysis was made using Problems 3 and 4. DR was calculated using basic CMPM mesh (four bins for Problem 3 and eight bins for Problem 4) and contrasted against the FMM using MC and a varying number of bins. This was done to illustrate the advantage of CMPM over FMM, especially in three-dimensional analysis where thousands of bins may be required for FMM to calculate the eigenvalues accurately. Problem 3 was run using 5000 particles per cycle, 200 inactive cycles and 1000 active cycles and is shown in Fig. 9. Problem 4 was run using 20,000 particles per cycle, 200 inactive cycles and 2000 active cycles and is shown in Fig. 10. Trend lines were added to visually show the number of bins in FMM necessary to reach the lower ends of 2σ CI of the CMPM estimate in each case.

As can be seen, FMM needs significantly more bins to accurately estimate DR, especially for Problem 4. The number of bins necessary for FMM to compute DR within the CMPM 1σ and 2σ CI is estimated to be 93 and 146, respectively, for Problem 3 and 2815 and 4169, respectively, for Problem 4 using the trend lines. CMPM requires only four bins for Problem 3 and eight bins for Problem 4. This is a significant difference that highlights the benefit of CMPM for large three-dimensional analysis. In passing, ARMA(2,1) HFDM yields $0.990 \pm 0.002(2\sigma)$ for Problem 4 in Fig. 10.

9. Degenerate eigenvalues

The appearance of multiplicity (repeated or degenerate eigenvalues) can make some problems difficult to solve. Also the eigenvalues need only be statistically indistinguishable (within the uncertainty of each other) to be considered degenerate. There are two effects of degeneracy: slower convergence of the second of degenerate eigenvalues and the occasional appearance of complex solutions.

9.1. Degenerate eigenvalue convergence

The second of degenerate eigenvalues often converges extremely slowly. The recommended number of active cycles $\eta_{AC} = \max[500, \ln(10^{-3})/\ln(\lambda_1)]$ is often inadequate for the second degenerate eigenvalue. For example, in Problem 2 the k_1/k_0 and k_2/k_0 eigenvalues are theoretically equal due to symmetry. Fifty replicas of this problem were run using the recommended 500 active cycles and 30,000 particles per cycle. Fig. 11 shows k_1/k_0 estimates with a 1σ interval. The average of the replicas was $0.9570 \pm 0.0015(1\sigma)$, which contains the benchmark value of 0.9581 within 1σ and 36 out of 50 replicas contained the benchmark value within a 1σ interval. This analysis does not appear biased. Fig. 12 shows the k_2/k_0 results. The k_2/k_0 average was $0.93096 \pm 0.00222(1\sigma)$ and 18 out of 50 replicas contained the benchmark value within a 1σ interval. This average value does not contain the benchmark FMM result of 0.95710; it is extremely biased. Other problems that had degenerate eigenvalues were tested and showed similar results.

Based on Fig. 12, it is clear that the recommended 500 active cycles is not sufficient. This problem was rerun using 5000 active cycles. These results are shown in Fig. 13. The k_2/k_0 average was 0.95214 ± 0.00048 $(1\ \sigma)$ and 24 out of 50 replicas contained the benchmark value within a 1σ interval. Though it is closer to the benchmark value, it still does not contain the benchmark FMM result of 0.95710 even within a 3σ range. Therefore, if degeneracy is suspected, it is recommended to run at least 10,000 active cycles regardless of η_{AC} in order to compute the smaller of the suspected degenerate eigenvalues.

The rate of convergence for these degenerate eigenvalues varied based on the problem. An accurate estimate of the necessary number of active cycles for these particular eigenvalues was not determined, though two comments can be made. First, since the appearance of degenerate eigenvalues is often caused by material and geometry symmetry, users can often have a general sense of which eigenvalues may be degenerate. Second, degenerate eigenvalues often appear as complex

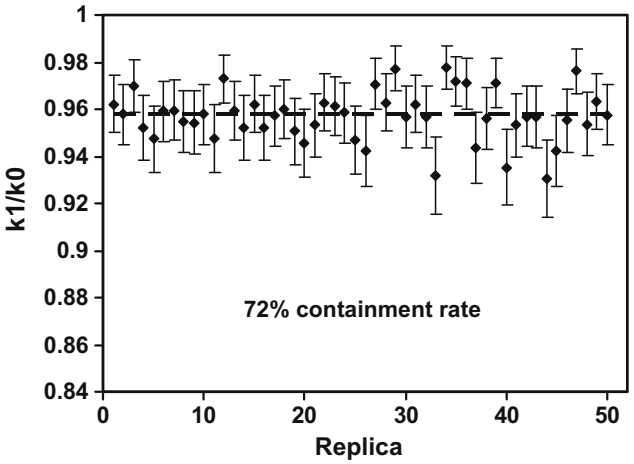

Fig. 11. Problem 2 – 50 replicas of k_1/k_0 $w/1\sigma$ standard deviation using 500 active cycles.

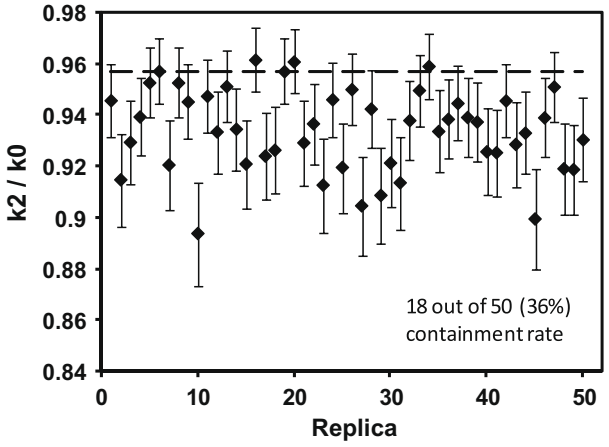

Fig. 12. Problem 2 – 50 replicas of k_2/k_0 $w/1\sigma$ standard deviation using 500 active cycles.

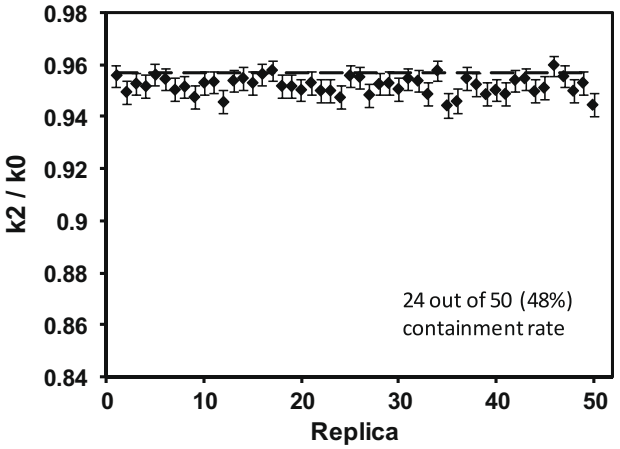

Fig. 13. Problem 2 – 50 replicas of k_2/k_0 $w/1\sigma$ standard deviation using 5000 active cycles.

conjugates of NP matrix with the imaginary part much smaller in magnitude than the real part. In this case, if the real or imaginary part of the corresponding eigenvector is used as a projection vector, the eigenvalue estimates do not suffer from the prolonged bias and can be used as accurate estimates of the degenerate eigenvalue. This is derived in the next subsection.

9.2. Appearance of complex components due to degeneracy

In this section, an analysis is made of the appearance of complex eigenvalue solutions. There are several immediate observations that can be made before beginning the analysis. First, it has been proven that the k-eigenvalue solutions of the mono-energetic transport equation are all real and discrete [4]. The same statement cannot be made for continuous energy problems, though the authors do not know of any real-world problems where complex eigensolutions have been encountered. Second, the NP matrix $\mathbf{A}_0$is not symmetric, so it cannot be guaranteed that the eigenvalues of $\mathbf{A}_0$ are all real. The issue is related to the apparent appearance of complex components due to statistical fluctuation for degenerate eigenvalues.

The questions that must be investigated are (1) under what conditions complex components are likely to arise, (2) how frequent complex components occur, (3) what the relative magnitude of the imaginary part is with respect to the real part, and (4) how to properly account for the appearance of complex solutions. This analysis provides mainly numerical results in answer to these questions, since analytical derivations cannot be made for the spurious complex output that is due to large or irregular statistical fluctuation.

The conditions under which complex components could arise were considered first. When only few active cycles were used $\left(\sim O\left(10^2\right)\right)$, consecutive eigenvalues sometimes appeared as complex conjugates even when their relative magnitude was only $k_{i+1}/k_i > 0.9$. When the number of active cycles became larger $\left(\sim O\left(10^3\right)\right)$ and the eigenvalues became more distinguishable, this did not occur. When the consecutive eigenvalues were much closer in magnitude $(k_{i+1}/k_i > 0.99)$, the frequency of complex components increased.

Table 9 shows the frequency of the k_2/k_0 and k_3/k_0 eigenvalue ratios appearing as complex conjugates of each other in 50 replicas of Problem 1 using a 25-bin mesh. Table 10 shows the frequency of the k_1/k_0 and k_2/k_0 eigenvalue ratios appearing as complex conjugates of each other in 50 replicas of Problem 2 using a 4-bin mesh. These tables were both generated using 30,000 particles per cycle. In Problem 2, the k_1 and k_2 eigenvalues (and likewise the k_1/k_0 and k_2/k_0 eigenvalue ratios) are theoretically exactly equal due to the symmetry in the problem, while in Problem 1, the k_2/k_0 and k_3/k_0 are numerically so close that they can not be statistically indistinguishable. As a quick aside, note that an odd number of source tally bins (25 in this case) were applied across the fissionable regions for the first case in Table 9, meaning that one bin covered part of the left and right regions. This is not a concern, however, because as long as there is only minimal cancellation of the associated eigenmode, any bin scheme can be applied, as was explained earlier. These results suggest that there is a relatively high rate of occurrence even with thousands of active cycles. Simply running more cycles does not seem to be an optimal way to prevent the incidence of complex components.

Table 9
Frequency of complex k_2/k_0 and k_3/k_0 eigenvalues in Problem 1.

Active cycles	Frequency of complex (%)
5000	24
8000	14
11,000	20

Table 10
Frequency of complex k_1/k_0 and k_2/k_0 eigenvalues in Problem 2.

Active cycles	Frequency of complex (%)
300	32
500	24
5000	18

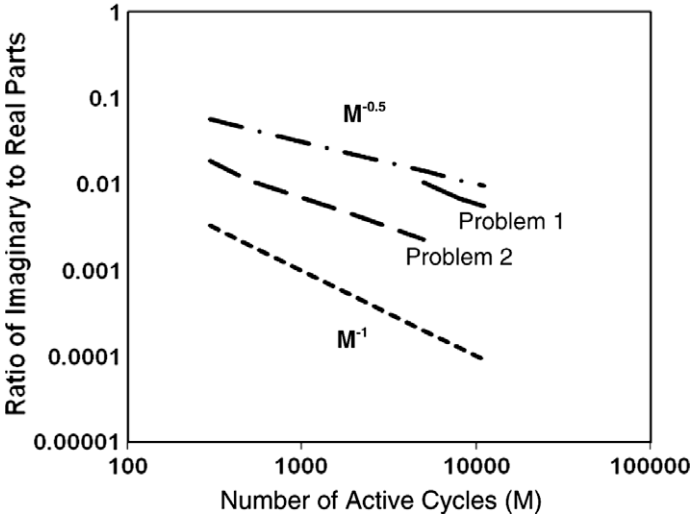

Fig. 14. Ratio of magnitudes of imaginary to real parts of k_2/k_0 and k_1/k_0 eigenvalue of A_0 in Problems 1 and 2, respectively.

The magnitude of the imaginary part relative to the real part is also important to consider. Specifically, we wish to quantify an upper bound of the imaginary part and study whether or not the magnitude changes based on the number of active cycles. It was expected that the largest imaginary parts would occur in problems with the fewest active cycles. Problems 1 and 2 were used again, and this assumption was confirmed. The average magnitudes of the imaginary components were plotted in a log–log plot in Fig. 14. These magnitudes were contrasted against M^{-1} and $M^{-1/2}$ to illustrate how the imaginary component decreases based on the number of active cycles. As can be seen, the magnitude is bounded and appears to decrease no slower than $M^{-1/2}$. This numerical result suggests that the appearance of complex component is strongly related to statistics while we have not so far obtained any numerical result indicating that the neglect of non-linear terms in error propagation would be part of the cause of the appearance of complex component. To clarify the latter aspect, we would need to develop non-linearity diagnosis which is outside the scope of present work. Instead we present a practical remedy to cope with the appearance of complex component.

Having quantified the magnitude of the imaginary parts will allow us to select appropriate methods to account for the appearance of complex eigenvalue ratios. Two such ideas are presented. The first method is to use the real part of the eigenvector of $\mathbf{A}_0$ as the projection vector and simply ignore the imaginary part. Since $\mathbf{A}_0$ is a real matrix, the eigenvalue problem described in Eq. (78) is rewritten as

$$\mathbf{A}_0^T\vec{p} = \lambda\vec{p}, \tag{116}$$
$$\mathbf{A}_0^T\vec{p}^* = \lambda^*\vec{p}^*, \tag{117}$$

where the conjugate quantities are defined as

$$\vec{p} = \vec{p}_R + i\vec{p}_I, \tag{118}$$
$$\vec{p}^* = \vec{p}_R - i\vec{p}_I, \tag{119}$$
$$\lambda = \lambda_R + i\lambda_I, \tag{120}$$
$$\lambda^* = \lambda_R - i\lambda_I. \tag{121}$$

R and I signify the real and imaginary parts, respectively. Taking the sum of Eqs. (116) and (117) yields

$$\mathbf{A}_0^T(\vec{p} + \vec{p}^*) = \lambda\vec{p} + \lambda^*\vec{p}^*. \tag{122}$$

Table 11
k_2/k_0 & k_3/k_0 of Problem 1 using real & imaginary part of eigenvector in CMPM with 24-bin mesh and 2σ standard deviation.

	GFM using 1800-bin mesh	CMPM 2σ CI using real part eigenvector	CMPM 2σ CI using imaginary part eigenvector
k_2/k_0, k_3/k_0	0.304653, 0.304636	(0.293625,0.329863)	(0.288569,0.324869)

The RHS and LHS of the equation can be manipulated individually as

$$\text{LHS}: \vec{p} + \vec{p}^* = (\vec{p}_R + i\vec{p}_I) + (\vec{p}_R - i\vec{p}_I) = 2\vec{p}_R, \tag{123}$$

$$\text{RHS}: \lambda\vec{p} + \lambda^*\vec{p}^* = (\lambda_R + i\lambda_I)(\vec{p}_R + i\vec{p}_I) + (\lambda_R - i\lambda_I)(\vec{p}_R - i\vec{p}_I) = 2\lambda_R\vec{p}_R - 2\lambda_I\vec{p}_I. \tag{124}$$

Substituting these into Eq. (122), we obtain

$$\mathbf{A}_0^T 2\vec{p}_R = 2\lambda_R\vec{p}_R - 2\lambda_I\vec{p}_I. \tag{125}$$

Assuming that the magnitude of the real eigenvalue component is much larger than the magnitude of the imaginary part, i.e., $|\lambda_I| \ll |\lambda_R|$, then the RHS of Eq. (125) can be approximated as

$$\mathbf{A}_0^T\vec{p}_R = \lambda_R\vec{p}_R - \lambda_I\vec{p}_I \approx \lambda_R\vec{p}_R. \tag{126}$$

Thus, the eigenvalue problem remains the same even if only the real part of the eigenvector is used. This proof is based on the assumption that the magnitude of the imaginary part of the eigenvalue is much less than the real part. Since the magnitude was found to be bounded by the number of active cycles in Fig. 14, this assumption appears to be valid. This was the method used in this research. All results presented use only the real part of the eigenvector of $\mathbf{A}_0^T$ as the projection vector. However, the imaginary part can also be used since [16]

$$\mathbf{A}_0^T 2i\vec{p}_I = \mathbf{A}_0^T(\vec{p} - \vec{p}^*) = \lambda\vec{p} - \lambda^*\vec{p}^* = 2i\lambda_R\vec{p}_I + 2i\lambda_I\vec{p}_R \approx 2i\lambda_R\vec{p}_I. \tag{127}$$

Table 11 shows k_2/k_0 and k_3/k_0 of Problem 1 using 15,000 active cycles and 30,000 particles per cycle, which correspond to these values of Table 5 obtained by different computational conditions. One can see that the use of the real and imaginary part of NP matrix eigenvector in CMPM yield statistically indistinguishable values for the corresponding GFM benchmark values. More details on these complex component issues were reported in a recent conference [16].

The second method to account for the imaginary components is to perform the autoregressive fitting to a complex time series. This would be a desirable method if the magnitude of the imaginary component was not negligible or if it were believed that the eigenvalue was, in fact, complex in nature. Allowing for complex valued processes is not difficult, though it does require some changes to the definitions of the autocovariance functions [17]. Since the authors do not know of any real-world problems that have complex eigensolutions and we did not encounter problems where $|\lambda_I|$ was not negligible compared to $|\lambda_R|$, the theory will not be presented here.

10. Summary and comments

A novel method of computing the k-eigenvalues of the neutron transport equation in Monte Carlo calculations was presented. The method is termed Coarse Mesh Projection Method and transforms the multi-dimensional MC source distribution to a form amenable to one dimensional time series techniques, enabling the eigenvalue ratios to be extracted statistically and thus eliminating the same discretization error effect as deterministic methods. The application of time series techniques allows for the largest non-fundamental eigenvalues to be calculated using very coarse mesh schemes (as few as two bins per coordinate direction) making the calculations extremely fast and efficient. It also enables one to compute the standard deviation of these eigenvalues from a single run (replica) of Monte Carlo calculation. To accurately calculate the eigenvalues, however, certain conditions must be met. First, there must be minimal-to-no cancellation of the sought-after eigenmode over the source distribution tally bins. The effect of other eigenmodes must also be removed, but this occurs automatically in CMPM by using the eigenvector of the transpose of the noise propagation (NP) matrix. Second, we impose that the minimum number of active cycles equal to $\eta_{AC} = \max\left[500, \ln(10^{-3})/\ln(\lambda_1)\right]$ must be used to ensure unbiased results. In cases where the eigenvalues are very close in magnitude ($k_{i+1}/k_i > 0.99$), more cycles, typically at least 10,000 cycles, are necessary to calculate the smaller of degenerate eigenvalues k_{i+1} accurately. Complex conjugate eigenvalues of the NP matrix can arise in these cases for about the likelihood of 20%. We have shown that if the imaginary part of the complex eigenvalue of noise propagation matrix is much smaller in magnitude than the real part, the real or imaginary part of the eigenvector can safely be used. It can be concluded that CMPM correctly computes the eigenvalues as far as the true eigenvalues of sought-after eigenmodes are real.

A three space-dimensional pressurized water reactor problem was analyzed. It was demonstrated that the computation of largest non-fundamental mode eigenvalue with CMPM needed only eight bins while the same computation with fission matrix method (FMM) needed at least several thousand bins. The author's opinion on this huge performance difference is as follows. FMM approximates infinite dimensional discrete eigenvalue spectrum by finite dimension. CMPM seeks to make not-sought-after eigenmodes cancel out. This difference in approach will be the cause of great performance gain of CMPM

over FMM for large three space-dimensional problems and CMPM is to be regarded the so-called feature extraction approach in signal processing. The feature extraction aspect was addressed in a recent conference by demonstrating that higher order autoregressive fitting improved eigenvalue estimation without increasing the number of bins [16]. Therefore, the present work is new and original in demonstrating that a methodology of feature extraction nature can be a valuable and efficient tool in nuclear criticality eigenvalue problems.

There are many further areas of study that could be explored with regard to this method. Among them, a stronger requirement for the necessary number of active cycles should be investigated since the correlation distance between cycles does not always provide an accurate gauge, especially when there is multiplicity or degenerate eigenvalues.

Acknowledgments

We express our sincere thanks to Dr Forrest B. Brown at Los Alamos National Laboratory for his suggestion of the study of practical implementation issues. This work was supported by US Department of Energy, mostly under Nuclear Engineering Educational Research (NEER) with the Contract No. DE-FG07-05ID14705 (2005–2008) and partly under the LANL student summer job of the first author (2005 and 2007) while he was a graduate student at University of New Mexico.

References

[1] J. Duderstadt, L. Hamilton, Nuclear Reactor Analysis, John Wiley& Sons, NY, 1976.
[2] S.A. Dupree, S.K. Fraley, A Monte Carlo Primer: A Practical Approach to Radiation Transport, Kluwer Academic/Plenum Publishers, NY, 2002.
[3] T. Ueki, F.B. Brown, D.K. Parsons, J.S. Warsa, Time series analysis of Monte Carlo fission sources I: dominance ratio computation, Nucl. Sci. Eng. 148 (2004) 374–390.
[4] D.C. Sahni, Some new results pertaining to criticality and time eigenvalues of one-speed neutron transport equation, Prog. Nucl. Energy 30 (3) (1996) 305–320.
[5] E.M. Gelbard, R.E. Prael, Monte Carlo work at Argonne National Laboratory, ANL-75-2, Argonne National Laboratory, 1974.
[6] R.J. Brissenden, A.R. Garlick, Biases in the estimation of k_{eff} and its error by Monte Carlo methods, Ann. Nucl. Energy 13 (1986) 63–83.
[7] T. Ueki, F.B. Brown, D.K. Parsons, D.E. Kornreich, Autocorrelation and dominance ratio in Monte Carlo criticality calculations, Nucl. Sci. Eng. 145 (2003) 279–290.
[8] J. Lieberoth, A Monte Carlo technique to solve the static eigenvalue problem of the Boltzmann transport equation, Nukleonik 11 (1968) 213.
[9] R. Durrett, Probability: Theory and Examples, second ed., Wadsworth Publishing Company, CA, 1996.
[10] G.E. Box, G.M. Jenkins, G.C. Reinsel, Time Series Analysis: Forecasting and Control, Prentice-Hall, Inc., Upper Saddle River, NJ, 1997.
[11] D.E. Kornreich, D.K. Parsons, The Green's function method for effective multiplication benchmark calculations multi-region slab geometry, Ann. Nucl. Energy 31 (13) (2003) 1477.
[12] T.A. Wareing, J.M. McGhee, J.E. Morel, S.D. Pautz, Discontinuous finite element Sn methods on three-dimensional unstructured grids, Nucl. Sci. Eng. 138 (2001) 256.
[13] K.W. Morton, Criticality Calculations by Monte Carlo Methods, AERE-TR-1903, Harwell, 1956.
[14] M. Nakagawa, T. Mori, Whole core calculations of power reactors by use of Monte Carlo method, J. Nucl. Sci. Technol. 30 (7) (1993) 692–701.
[15] D. MacMillan, Monte Carlo confidence limits for iterated-source calculations, Nucl. Sci. Eng. 50 (1973) 73–87.
[16] B.R. Nease, T. Ueki, The appearance of complex values in Monte Carlo calculation of degenerate eigenvalues, in: International Conference on Mathematics, Computational Methods& Reactor Physics (M&C 2009) Saratoga Springs, CD-ROM, New York, May 3–7, 2009.
[17] B.R. Nease, Time Series Analysis of Monte Carlo Neutron Transport Calculations, Ph.D. Dissertation, University of New Mexico, 2008.