Joint International Topical Meeting on Mathematics & Computation and Supercomputing in Nuclear Applications (M&C + SNA 2007)
Monterey, California, April 15-19, 2007, on CD-ROM, American Nuclear Society, LaGrange Park, IL (2007)

A NOVEL SOURCE CONVERGENCE ACCELERATION SCHEME FOR MONTE CARLO CRITICALITY CALCULATIONS, PART I: THEORY

David P. Griesheimer
Bechtel Bettis, Inc.
P.O. Box 79
West Mifflin, PA 15122
grieshei@bettis.gov

Bryan E. Toth
Department of Nuclear Engineering and Radiological Sciences
University of Michigan
2355 Bonisteel Boulevard
Ann Arbor, MI 48109
betoth@umich.edu

ABSTRACT

A novel technique for accelerating the convergence rate of the iterative power method for solving eigenvalue problems is presented. Smoothed Residual Acceleration (SRA) is based on a modification to the well known fixed-parameter extrapolation method for power iterations. In SRA the residual vector is passed through a low-pass filter before the extrapolation step. Filtering limits the extrapolation to the lower order eigenmodes, improving the stability of the method and allowing the use of larger extrapolation parameters. In simple tests SRA demonstrates superior convergence acceleration when compared with an optimal fixed-parameter extrapolation scheme. The primary advantage of SRA is that it can be easily applied to Monte Carlo criticality calculations in order to reduce the number of discard cycles required before a stationary fission source distribution is reached. A simple algorithm for applying SRA to Monte Carlo criticality problems is described.

Key Words: power method, source convergence, Monte Carlo, acceleration, criticality

1. INTRODUCTION

Monte Carlo based criticality calculations are an important tool in nuclear design and safety analysis. The Monte Carlo method is especially well-suited for working with complex geometries commonly found in criticality safety problems. During criticality calculations Monte Carlo transport codes use an iterative method to converge to the correct multiplication factor (eigenvalue) and fission source distribution (eigenfunction) for the system of interest. For problems with large dominance ratios this iterative method can converge very slowly, requiring many batches of discarded neutrons before a stationary source distribution is achieved. The topics of fission source convergence [1-3] and stationarity [4] have been widely studied over the years. More recently, additional studies have focused on accelerating the convergence rate (i.e. reducing the number or size of discarded batches) [5-8]. In this paper we present a new scheme for improving the source convergence rate in Monte Carlo criticality calculations.

The paper is divided into two major sections. In the first section we derive a new acceleration scheme that is applicable to any type of power iteration method. The Smoothed Residual Acceleration (SRA) scheme is shown to provide superior acceleration relative to a fixed parameter extrapolation method for problems of interest. In the second section, we show how SRA can be used to accelerate the iterative source convergence method used in Monte Carlo eigenvalue calculations.

2. SMOOTHED RESIDUAL ACCELERATION (SRA)

2.1 Derivation

We begin the derivation of the SRA technique by considering a linear operator A with eigenvalues $\{\lambda_1, \ldots, \lambda_M\}$ and corresponding eigenfunctions $\{v_1, \ldots, v_M\}$ such that

$$A v_n = \lambda_n v_n \,. \tag{1}$$

Assume that the eigenvalues of A are ordered $|\lambda_1| > |\lambda_2| \geq \ldots \geq |\lambda_M| \geq 0$, with λ_1 referred to as the fundamental or dominant eigenvalue. One of the simplest techniques for determining the fundamental eigenfunction is the iterative power method. The power method begins with an initial eigenfunction guess $b^{(0)}$, and constructs successive eigenfunction approximations with the relationship

$$b^{(t)} = \frac{A b^{(t-1)}}{\lambda_1^{(t-1)}} \approx \frac{A b^{(t-1)}}{\lambda_1} \qquad \text{for } t >> 0 \,. \tag{2}$$

To show that $b^{(t)}$ converges to v_1 as $t \to \infty$, expand the initial guess $b^{(0)}$ in terms of the eigenbasis of A

$$b^{(0)} = \sum_{n=1}^{M} \gamma_n v_n \,. \tag{3}$$

Equation (2) can now be used to calculate $b^{(1)}$

$$b^{(1)} = \sum_{n=1}^{M} \frac{A \gamma_n v_n}{\lambda_1} = \sum_{n=1}^{M} \frac{\lambda_n}{\lambda_1} \gamma_n v_n \,. \tag{4}$$

Successive applications of Eq. (2) give

$$b^{(t)} = \sum_{n=1}^{M} \left(\frac{\lambda_n}{\lambda_1} \right)^t \gamma_n v_n \,. \tag{5}$$

Because the eigenvalues are ordered such that $|\lambda_1| > |\lambda_2| \geq \ldots \geq |\lambda_M|$ it follows that

$$\lim_{t\to\infty} b^{(t)} = \gamma_1 \nu_1 . \tag{6}$$

The convergence rate of the power iteration is dominated by the ratio of the second eigenvalue to the fundamental eigenvalue ($|\lambda_2/\lambda_1|$), referred to as the dominance ratio. By inspection of Eq. (5), we see that if the dominance ratio is close to 1 it will take many iterations for the power method to converge to a desirable level of accuracy. Unfortunately, large dominance ratios are common in many types of physical systems. For this reason, a lot of research has been conducted to develop numerical methods that converge faster than the power method. One such method accelerates the convergence of the standard power method by extrapolating between successive eigenfunction guesses. This extrapolation is achieved by modifying the power method, Eq. (2) as follows

$$b^{(t)} = \omega \frac{Ab^{(t-1)}}{\lambda_1} + (1-\omega) b^{(t-1)} , \tag{7}$$

where ω is referred to as the extrapolation factor. For illustrative purposes we can rewrite Eq. (7) as

$$b^{(t)} = b^{(t-1)} + \omega \left(\frac{Ab^{(t-1)}}{\lambda_1} - b^{(t-1)} \right). \tag{8}$$

In Eq. (8) the term in parentheses is the change in $b^{(t-1)}$ resulting from the linear transformation A/λ_1, hereafter referred to as the residual. From Eq. (8) we see that when $\omega > 1$ the residual is being artificially amplified in an attempt to force a larger change between the successive eigenvalue estimates; a technique known as overrelaxation. If $\omega < 1$, the residual change between iterations is effectively damped, causing the method to converge at a slower rate (underrelaxation). When $\omega = 1$, there is no extrapolation and Eq. (8) reduces to the standard power method, Eq. (2).

In fixed parameter extrapolation a single value of ω ($\omega > 1$) is chosen and used for all of the power iterations. This technique is known to accelerate the convergence rate of the power method [11]. It can be shown [11] that the fastest convergence rate for a fixed parameter extrapolation scheme is achieved when

$$\omega = \frac{1}{1 - \lambda_2 / (2\lambda_1)} . \tag{9}$$

Unfortunately, Eq. (9) requires prior knowledge of the dominance ratio, which is not often available. By inspection of Eq. (9) we note that $\omega \to 2$ as the dominance ratio approaches 1. Additional studies have proven that the extrapolated power method is unstable for a fixed $\omega > 2$ [11].

Faster convergence rates can be achieved by allowing different extrapolation factors for each iteration. This concept is the basis for Chebyshev acceleration, which uses Chebyshev polynomials to calculate the optimal extrapolation factors. For systems with a high dominance ratio, Chebyshev acceleration dramatically improves the convergence rate of the fixed parameter extrapolation power method. Presently, Chebyshev acceleration is widely used for accelerating fission source convergence in deterministic neutron transport codes.

The extrapolated power method, Eq. (8), is our starting point for the derivation of a new power method acceleration technique: smoothed residual acceleration (SRA). We begin by defining a shifted extrapolation factor

$$\alpha = \omega - 1\,. \tag{10}$$

Substituting Eq. (10) into Eq. (8) and rearranging gives a new expression for the extrapolated power method,

$$b^{(t)} = \frac{Ab^{(t-1)}}{\lambda_1} + \alpha\left(\frac{Ab^{(t-1)}}{\lambda_1} - b^{(t-1)}\right). \tag{11}$$

Notice that Eq. (11) looks similar to the original extrapolated power method, given in Eq. (8). The only difference is that Eq. (11) uses an updated eigenfunction estimate, $Ab^{(t-1)}/\lambda_1$, as the reference point for the extrapolation. Also, because α is shifted from the original extrapolation factor ω, the stable extrapolation range for Eq. (11) is also shifted. Equation (11) is overrelaxed when $\alpha > 0$, underrelaxed when $\alpha < 0$, and reduced to the regular power method when $\alpha = 0$.

In order to analyze the convergence behavior of the extrapolated power method given in Eq. (11) we begin with an arbitrary initial eigenfunction guess $b^{(0)}$. As before, expand the initial guess in terms of the eigenbasis of A

$$b^{(0)} = \sum_{n=1}^{M} \gamma_n v_n\,. \tag{12}$$

Substituting Eq. (12) into (11) gives an expression for $b^{(1)}$

$$b^{(1)} = \frac{1}{\lambda_1}\sum_{n=1}^{M} \gamma_n A v_n + \alpha\left(\frac{1}{\lambda_1}\sum_{n=1}^{M} \gamma_n A v_n - \sum_{n=1}^{M} \gamma_n v_n\right). \tag{13}$$

Using the eigenvalue / eigenfunction relationship given in Eq. (1) this expression can be simplified to yield

$$\begin{aligned} b^{(1)} &= \frac{1}{\lambda_1}\sum_{n=1}^{M} \gamma_n \lambda_n v_n + \alpha\left(\frac{1}{\lambda_1}\sum_{n=1}^{M} \gamma_n \lambda_n v_n - \sum_{n=1}^{M} \gamma_n v_n\right) \\ &= \sum_{n=1}^{M} \frac{\lambda_n}{\lambda_1}\gamma_n v_n + \alpha\left(\sum_{n=1}^{M}\left(\frac{\lambda_n}{\lambda_1} - 1\right)\gamma_n v_n\right) \end{aligned}$$

$$=\sum_{n=1}^{M}\left[\frac{\lambda_n}{\lambda_1}+\alpha\left(\frac{\lambda_n}{\lambda_1}-1\right)\right]\gamma_n\,\nu_n$$

$$b^{(1)}=\sum_{n=1}^{M}\xi_n\,\gamma_n\,\nu_n \tag{14}$$

where

$$\xi_n=\left[1+(1+\alpha)\left(\frac{\lambda_n}{\lambda_1}-1\right)\right]. \tag{15}$$

Repeating this process *t* times gives an expression for the eigenfunction approximation $b^{(t)}$

$$b^{(t)}=\sum_{n=1}^{M}\left(\xi_n\right)^t\gamma_n\,\nu_n\,. \tag{16}$$

Equation (16) shows that the decay rates of the individual eigenmodes are tied to the values of the coefficients ξ_n. In order for $b^{(t)}$ to converge to the fundamental eigenfunction, v_1, the following conditions must be met: 1) $\xi_1 = 1$, and 2) $|\ \xi_n\ | < 1$ for all $n > 1$. Condition 1 is immediately verified by setting $\lambda_n = \lambda_1$ in Eq. (15). Condition 2, however, is only satisfied when

$$-1<\alpha<\frac{1+\lambda_n/\lambda_1}{1-\lambda_n/\lambda_1} \tag{17}$$

for all modes *n*. Assuming that the smallest eigenvalue, λ_M, is close to zero, it follows immediately that convergence is guaranteed only for $|\alpha| < 1.0$. This condition is identical to the convergence criteria derived for the original formulation of the extrapolated power method. If a fixed value of α greater than 1 is chosen, then Eq. (17) is not satisfied for those eigenmodes with a corresponding eigenvalue

$$\lambda_n<\frac{\alpha-1}{\alpha+1}\lambda_1\,. \tag{18}$$

Any eigenmodes that do not satisfy Eq. (18) will be amplified between power iterations, causing the entire method to diverge. Equations (17) and (18) define an important convergence relationship between the shifted extrapolation parameter, α, and the highest-order (lowest magnitude) eigenvalue of *A*. Of particular importance is Eq. (17), which gives the largest stable extrapolation parameter, α_{critical}, as a function of the eigenvalue ratio. Figure 1 shows a plot of α_{critical} values versus eigenvalue ratio. This figure illustrates that α_{critical} increases rapidly as the eigenvalue ratio approaches 1, indicating that modes with eigenvalues close to 1 are stable for even large extrapolation factors.

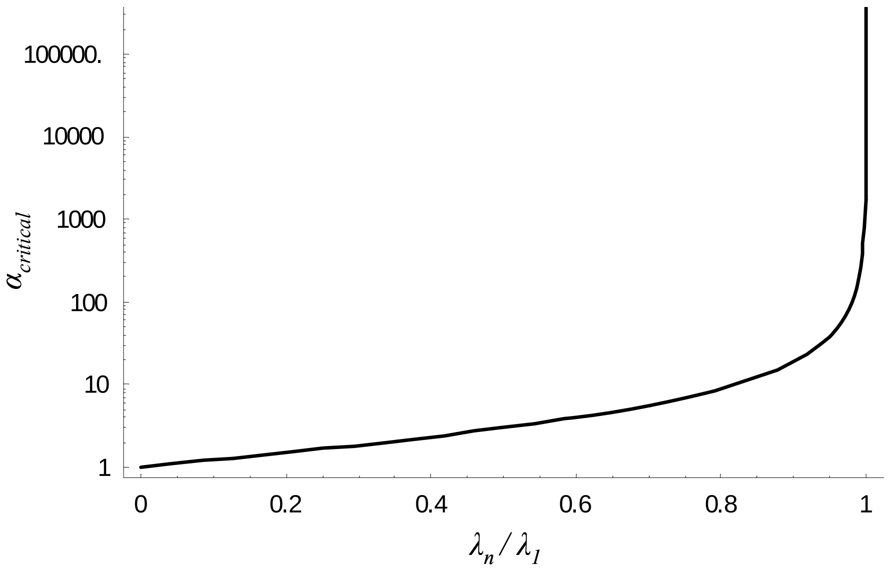

Figure 1. Critical (stable) extrapolation factor as a function of eigenvalue. The area underneath the curve is the region of stability for the extrapolation method.

This result can be understood by considering a single application of the operator A to an eigenfunction estimate b. During each application of A, the high order eigenmodes change (decay) more than the lower order modes. This means that the residual consists primarily of high-frequency terms due to the decay of the higher-order modes. Therefore, applying an extrapolation factor to the residual preferentially affects the higher order modes. However, the ultimate goal of the method is to improve the overall convergence rate by extrapolating the low order modes as far as possible between iterations. Choosing a large extrapolation factor will improve the convergence rate of the lower modes but it will also force the higher modes to change by an even larger amount. If the higher modes are extrapolated too far a high frequency noise contribution will build up as the method progresses. Consequently, the stability of the extrapolated power method is controlled by two main factors: the lower mode eigenvalues govern the convergence rate of the iterative process, while the higher eigenvalues determine how far the method can be extrapolated between iterations without becoming unstable.

Further consideration of these competing factors leads to an interesting question: is it possible to modify the original extrapolation technique to preferentially accelerate lower order modes, which are inherently more stable with respect to extrapolation? Conceptually this could be accomplished by applying some type of low-pass filter to the residual, in order to eliminate the high order modes that would diverge for $\alpha > 1$. Truncating the high order eigenmodes has the effect of smoothing the shape of the residual, giving rise to the name “smoothed residual acceleration” (SRA) for the method.

To demonstrate the effects of smoothing, consider a perfect (brick wall) low-pass filter, F, which allows all modes with eigenvalues $\geq \lambda_{n'}$ to pass and removes all modes $< \lambda_{n'}$. Applying this filter to the residual $Ab^{(t)}/\lambda_1$ - $b^{(t)}$ yields

$$F\left[\frac{Ab^{(t)}}{\lambda_1}-b^{(t)}\right]=\sum_{n=1}^{n'}\left(\frac{\lambda_n}{\lambda_1}-1\right)\gamma_n\,\nu_n\,. \tag{19}$$

Substituting the filtered residual for the true residual in Eq. (13) and repeating the preceding derivation, Eqs. (13) – (16), gives a new expression for the eigenfunction approximation after *t* iterations,

$$b^{(t)}=\sum_{n=1}^{n'-1}\left(\xi_n\right)^t\gamma_n\,\upsilon_n+\sum_{n=n'}^{M}\left(\frac{\lambda_n}{\lambda_1}\right)^t\gamma_n\,\upsilon_n\,. \tag{20}$$

Examination of Eq. (20) reveals that the application of the low pass filter has limited which modes are being accelerated. Modes up to the cutoff filter are being accelerated, while all modes after the filter cutoff are not accelerated and converge like the standard power method. Therefore, as long as the cutoff eigenvalue for *F* satisfies Eq. (18) for the chosen α, the method is guaranteed to converge. This result can be used immediately to show that SRA will converge for any extrapolation factor α, if the cutoff eigenvalue of *F* is set to

$$\lambda_{n'}=\frac{\alpha-1}{\alpha+1}\lambda_1\,. \tag{21}$$

This result gives a basic stability requirement for the method, but it does not guarantee that any acceleration will occur. In fact, it is easily shown that choosing $\lambda_{n'}$ (or α) according to Eq. (21) will, in general, make the overall convergence rate of the method worse. To see this, we begin by solving for $\xi_{n'}$, using the cutoff eigenvalue defined in Eq. (21)

$$\xi_{n'}=1+\left(1+\alpha\right)\left(\frac{\left(\alpha-1\right)\lambda_1}{\left(\alpha+1\right)\lambda_1}-1\right)=-1\,. \tag{22}$$

If the separation between $\lambda_{n'}$ and $\lambda_{n'-1}$ is small, it follows that $|\xi_{n'-1}| \approx 1$. For values of α where $|\xi_{n'-1}| > \lambda_2/\lambda_1$, SRA will actually converge at a slower rate than the standard power method. We overcome this problem by deriving an optimal value for both the extrapolation factor and filter cutoff threshold, which will provide maximum acceleration and guarantee convergence.

Inspection of the SRA algorithm, Eq. (20), reveals that the overall convergence rate is controlled by the greater of the value $(\lambda_{n'}/\lambda_1)$ and the maximum value of $|\xi_n|$ over $2 \leq n < n'$. Due to the monotonically non-increasing nature of the eigenvalues, it follows from Eq. (15) that the ξ_n sequence is also monotonically non-increasing. Therefore, it holds that the maximum value of $|\xi_n|$ is equal to the larger of the terms $|\xi_2|$ and $|\xi_{n'-1}|$. These results, in turn, can be used to show that maximizing the convergence rate of the SRA algorithm is equivalent to simultaneously minimizing $|\xi_2|$, $|\xi_{n'-1}|$, and $(\lambda_{n'}/\lambda_1)$.

To begin solving, consider a simpler simultaneous minimization problem involving only $|\xi_{n'-1}|$ and $(\lambda_{n'}/\lambda_1)$. It was previously demonstrated that both λ_n and ξ_n are monotonically non-increasing sequences. However, for values of $\xi_n < 0$, $|\xi_n|$ becomes a monotonically non-decreasing sequence. In this range $|\xi_{n'-1}|$ and $(\lambda_{n'}/\lambda_1)$ are simultaneously minimized when $-\xi_{n'-1} = (\lambda_{n'}/\lambda_1)$. By assuming that eigenvalues n' and n'-1 are close together ($\lambda_{n'} \approx \lambda_{n'-1}$), Eq. (15) can be set equal to $(\lambda_{n'}/\lambda_1)$ and solved for α

$$\alpha = \frac{2\,\lambda_{n'}/\lambda_1}{1-\lambda_{n'}/\lambda_1}\,. \tag{23}$$

Equation (23) gives an optimal relationship between the extrapolation factor α and the filter cutoff threshold $\lambda_{n'}$. The derivation of Eq. (23) assumes that the minimum between $|\xi_{n'-1}|$ and $(\lambda_{n'}/\lambda_1)$ occurs when $\xi_{n'-1}$ is less than zero. In order to demonstrate that this assumption is valid, we need to show that no positive value of $\xi_{n'-1}$ is less than $(\lambda_{n'}/\lambda_1)$. By simple algebra, it is easily shown that the inequality $\xi_{n'-1} \le (\lambda_{n'}/\lambda_1)$ only holds for $\alpha \le 0$. Since α must be greater than zero to provide acceleration, it follows that the initial assumption is valid.

Since the minimization between $|\xi_{n'-1}|$ and $(\lambda_{n'}/\lambda_1)$ occurs when $|\xi_{n'-1}| = (\lambda_{n'}/\lambda_1)$, the original three term minimization problem is reduced to a simpler two term minimization between $|\xi_2|$ and $(\lambda_{n'}/\lambda_1)$. As before, this minimization problem can be solved by finding the intersection point(s) between $|\xi_2|$ and $(\lambda_{n'}/\lambda_1)$. Setting $\pm\xi_2 = (\lambda_{n'}/\lambda_1)$ and solving for $\alpha > 0$ gives

$$\alpha = \frac{\lambda_{n'}-\lambda_2}{\lambda_2-\lambda_1}\,. \tag{24}$$

Equations. (23) and (24) give two separate optimization relationships between α and $\lambda_{n'}$. Setting Eq. (23) equal to (24) and rearranging gives the quadratic equation

$$\left(\frac{\lambda_{n'}}{\lambda_1}\right)^2 + \left(\frac{\lambda_2}{\lambda_1}-3\right)\frac{\lambda_{n'}}{\lambda_1} + \frac{\lambda_2}{\lambda_1} = 0\,, \tag{25}$$

which can be solved for the optimal filter cutoff $\lambda_{n'}$ as a function of the dominance ratio

$$\left(\frac{\lambda_{n'}}{\lambda_1}\right) = \frac{1}{2}\left[\left(\frac{\lambda_2}{\lambda_1}-3\right) - \sqrt{\left(\frac{\lambda_2}{\lambda_1}\right)^2 - 10\frac{\lambda_2}{\lambda_1}+9}\,\right]. \tag{26}$$

Once the optimal eigenvalue cutoff has been determined, the corresponding optimal value of α can be calculated via Eq. (23) or (24).

In order to compare the convergence properties of the power method with different acceleration schemes, it is useful to define a rate of convergence for the method,

$$\mu = \lim_{t\to\infty}\frac{\left|b^{(t+1)}-\gamma_1\nu_1\right|}{\left|b^{(t)}-\gamma_1\nu_1\right|}\,. \tag{27}$$

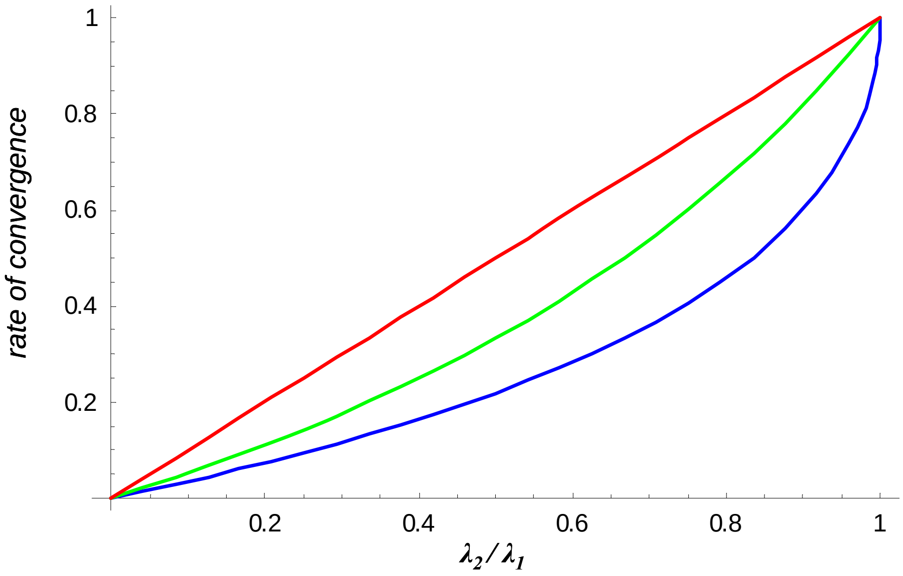

Figure 2. Convergence rate versus dominance ratio for the power method a) unaccelerated (red), b) with fixed extrapolation acceleration (green), and c) with smoothed residual acceleration (blue). Smaller convergence rates indicate better performance.

Notice that when comparing two flavors of the power method, the one with the smaller rate of convergence will converge faster.

In the case of SRA, choosing optimal values for $\lambda_{n'}$ and α guarantees that the overall rate of convergence is decreased to ($\lambda_{n'}/\lambda_1$), the minimum achievable for this technique. Figure 2 shows a plot of $\lambda_{n'}/\lambda_1$ (rate of convergence) versus dominance ratio for the power method with SRA. The concave-downward shape of the plot demonstrates that the SRA technique has improved the rate of convergence over the unaccelerated power method, especially in the range $0.9 \leq \lambda_2/\lambda_1 \leq 1$.

The rate of convergence provides a good metric for comparing the efficiency of different iterative methods. However, it is often more illustrative to consider the relative improvement of one acceleration method over another. For this we define an acceleration factor as the ratio of the rates of convergence between an unaccelerated and accelerated power method. This acceleration factor provides a way to measure the relative increase in efficiency per iteration for a given acceleration scheme. In the sense that power method acceleration techniques can be thought of as reducing the effective spectral radius of the linear operator, the acceleration factor gives a relative measure of how much the spectral radius has been improved.

Figure 3 shows a plot of the acceleration factor versus dominance ratio for the power method with SRA. Again, these results show that the SRA technique can produce significant improvements in the convergence efficiency of the power method, especially for problems with large dominance ratios. For example, in problems with a dominance ratio ≈ 0.9, the application of SRA makes the power method 1.5 times more efficient per iteration.

The results in Figures 2 and 3 demonstrate that SRA offers some acceleration over the standard power method, but it remains to be seen how SRA analytically compares against other

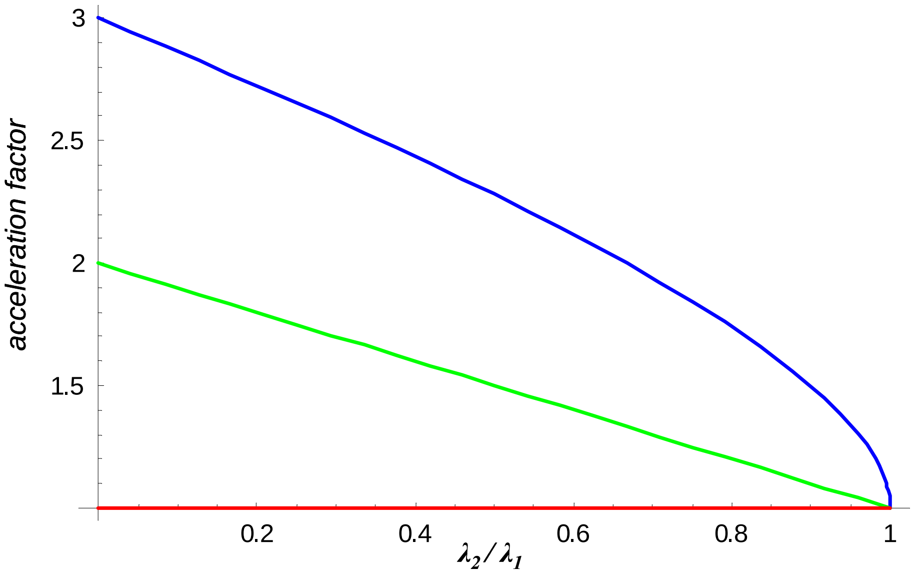

Figure 3. Acceleration factor (per iteration) versus dominance ratio for the power method a) unaccelerated (red), b) with fixed extrapolation acceleration (green), and c) with smoothed residual acceleration (blue). Larger acceleration factor values indicate more improvement over the unaccelerated power method.

acceleration techniques. We begin by comparing SRA against its predecessor: fixed parameter extrapolation. The optimal shifted extrapolation parameter for this technique can be calculated by substituting Eq. (9) into Eq. (10) to yield

$$\alpha_{opt} = \frac{\lambda_2/\lambda_1}{\lambda_2/\lambda_1 - 2}. \qquad (28)$$

Alternatively, Eq. (28) can be derived by setting $\xi_2 = -\xi_\infty$, and assuming that $\lambda_\infty \approx 0$. Substituting α_{opt} into Eq. (15) gives an equation for the series $\{\xi_n\}$. By inspection, the extrema for this series (i.e. $\max[|\xi_n|]$) occur at $n = 2$, and $n = \infty$, where

$$\max\left[|\xi_n|\right] = \frac{\lambda_2/\lambda_1}{2-\lambda_2/\lambda_1}. \qquad (29)$$

Equation (29) determines the limiting convergence behavior of the fixed parameter extrapolation method. This value can be used to calculate the rate of convergence and acceleration factor for the technique as a function of dominance ratio. These results are shown in Figures 2 and 3, compared against SRA and the unaccelerated power method. These results demonstrate that the SRA offers improved acceleration over the fixed parameter extrapolation method for all problems, regardless of dominance ratio. However, the improvement is especially pronounced for problems with high dominance ratios $\lambda_2/\lambda_1 \geq 0.9$.

Detailed comparisons between SRA and more advanced power method acceleration schemes such as Chebyshev acceleration and Krylov subspace methods are beyond the scope of this paper. However, it is expected that these advanced / adaptive acceleration techniques will significantly outperform the SRA. The real benefit of the SRA lies in its potential applicability to Monte Carlo eigenvalue calculations, which employ a version of the traditional power method. As noted in the introduction, few acceleration schemes have been successfully applied to iterative Monte Carlo source convergence. In this case, the SRA may provide the ability for Monte Carlo criticality codes to mimic the simplest extrapolation acceleration techniques commonly used in deterministic codes.

As one final note, the above convergence analysis is based on a purely theoretical analysis using a perfect low pass filter and assuming that the dominance ratio is known *a priori*. Unfortunately, the use of a perfect low-pass filter would essentially require the eigenbasis decomposition of the residual to be known. If this information is available, or could be calculated, then the solution itself could be calculated directly. Instead, we must consider realistic filtering techniques, with particular focus on methods that can be easily implemented in a Monte Carlo calculation.

2.2 Moving Average Filter

The primary obstacle to the practical application of SRA is the implementation of a suitable low pass filter for processing the residual. A wide variety of low-pass filters are available in signal processing literature. For convenience, we will limit our discussion in this paper to a very simple type of smoothing filter: the moving average. The moving average filter is a robust and versatile technique that is both easily analyzed and easily implemented in Monte Carlo transport simulations. In the following derivations we will demonstrate that the moving average actually behaves as a low-pass filter and can be used for to apply SRA to the power iterations during a Monte Carlo criticality calculation.

To begin, we define a moving average filter, F_{ma}, which replaces the value of a function *g(x)* at every point *x*, by the average value of *g(x)* within some interval *δ* about *x*. Using the mean value theorem, the filter *F* can be written as

$$F_{\mathrm{ma}}\left[g(x)\right]=\frac{1}{\delta}\int_{x-\delta/2}^{x+\delta/2}g(x')dx' . \tag{30}$$

This type of filter is especially convenient for use in Monte Carlo applications, which are best used for calculating integral quantities. To demonstrate the filtering properties of the moving average, expand *g(x)* in terms of a complete set of basis functions, $\{\psi_n(x)\}_{n=1,\infty}$,

$$g(x)=\sum_{n=1}^{\infty}c_n\,\psi_n(x), \tag{31}$$

with unknown expansion coefficients $\{c_n\}$. Substituting Eq. (31) into Eq. (30) yields a new expression for the filtered function

$$F_{\mathrm{ma}}\left[g(x)\right]=\frac{1}{\delta}\int_{x-\delta/2}^{x+\delta/2}\sum_{n=1}^{\infty}c_n\,\psi_n(x')dx'$$

$$F_{\text{ma}}\left[g(x)\right] = \sum_{n=1}^{\infty} \frac{c_n}{\delta} \int_{x-\delta/2}^{x+\delta/2} \psi_n(x')\,dx'$$

$$F_{\text{ma}}\left[g(x)\right] = \sum_{n=1}^{\infty} c_n\, F_{\text{ma}}\left[\psi_n(x)\right]. \tag{32}$$

Equation (32) gives the unsurprising result that "smoothing" the function *g(x)* by applying filter F_{ma} is equivalent to smoothing each of the basis functions $\{\psi_n(x)\}_{\text{n}=1,\infty}$ by applying F_{ma}.

In order to better understand the smoothing behavior of the filter F_{ma}, we must first rewrite the filter in form that is easier to analyze. To begin, we define a window function

$$W(x,\delta) = \begin{cases} 0 & \text{if } |x| > \delta/2 \\ 1/\delta & \text{if } |x| \le \delta/2 \end{cases}. \tag{33}$$

Note that the window function is related to the standard rectangle function by

$$W(x,\delta) = \Pi\left(\frac{x}{\delta}\right). \tag{34}$$

Using Eq. (33) and (30), the filter F_{ma} can be rewritten as a convolution of the window function and the argument of the filter

$$F_{\text{ma}}\left[g(x)\right] = \int_{-\infty}^{+\infty} g(x')W(x-x',\delta)\,dx'\,. \tag{35}$$

Using the convolution theorem, Eq. (35) can now be recast in frequency space as the product of the Fourier transforms of $g(x)$ and $W(x,\delta)$

$$\Im\left[F_{\text{ma}}\left[g(x)\right]\,\right] = \Im\left[g(x)\right]\,\Im\left[W(x-x',\delta)\right]. \tag{36}$$

The continuous Fourier transform of the window function can be solved exactly in terms of a spatial frequency f,

$$\Im\left[W(x,\delta)\right] = \int_{-\infty}^{+\infty} W(x,\delta)e^{-i2\pi f x}\,dx$$

$$\Im\left[W(x,\delta)\right] = \frac{1}{\delta}\int_{-\delta/2}^{\delta/2} e^{-i2\pi f x}\,dx$$

$$\Im\left[W(x,\delta)\right] = \frac{1}{\pi\delta f}\left[\frac{e^{i\pi\delta f} - e^{-i\pi\delta f}}{2i}\right]$$

$$\Im[W(x,\delta)] = \frac{\mathrm{Sin}[\pi\delta f]}{\pi\delta f}. \tag{37}$$

The right hand side of Eq. (37) is commonly defined as the normalized sine cardinal function (*sinc*$_\pi$),

$$\mathrm{sinc}_\pi[\delta f] = \frac{\sin[\pi\delta f]}{\pi\delta f}. \tag{38}$$

Substituting Eqns. (37) and (38) into Eq. (36) gives a final form for the continuous Fourier transform of F_{ma} applied to $g(x)$,

$$\Im[F_{\mathrm{ma}}[g(x)]\,] = \Im[g(x)]\,\mathrm{sinc}_\pi[\delta f]. \tag{39}$$

The *sinc*$_\pi$ function in Eq. (39) is a well known result in signal processing. As shown in Figure 6, *sinc*$_\pi$ acts as a low-pass filter, with δ controlling the width of the passband. Further inspection of Figure 6 reveals that the *sinc*$_\pi$ filter differs significantly from the ideal brick-wall low-pass filter assumed in the derivation of the SRA. While the *sinc*$_\pi$ filter is less than optimal, we show in the next section that it is especially well suited for use in Monte Carlo transport simulations.

The preceding derivation demonstrates that applying a moving average filter to a function *g(x)* in normal space is equivalent to applying a *sinc*$_\pi$ filter to *g(x)* in frequency space. However, the original derivation of the SRA (section 2.1) assumed that the residual function was filtered by eigenmode, rather than frequency. For certain simple, 1-D, homogeneous problems, the eigenfunctions of the system are very close to the basis functions used in the standard Fourier series. In this trivial case, each eigenfunction occurs at a single discrete frequency, and it is easy to determine the correspondence between filtering based on frequency and eigenmode. However, for more realistic problems, individual eigenfunctions will appear in frequency space as a continuous spectrum over a range of frequency values. In this case, it is not straightforward to relate the effects of filtering based on frequency as opposed to eigenmode. If eigenfunctions in the system have a wide frequency response, this can theoretically lead to a reduction in efficiency of the SRA algorithm. One particular concern is overfiltering of the low order eigenmodes. If the fundamental mode contains high-frequency components, these frequencies may be filtered out by SRA, causing the system to converge to the filtered eigenmode rather than the true eigenmode. In most cases the method will work well as long as the eigenvectors are naturally ordered according to frequency. While this result is intuitive (e.g. the lowest eigenvalues have the lowest frequency response) a proof of this conjecture is outside the scope of this paper.

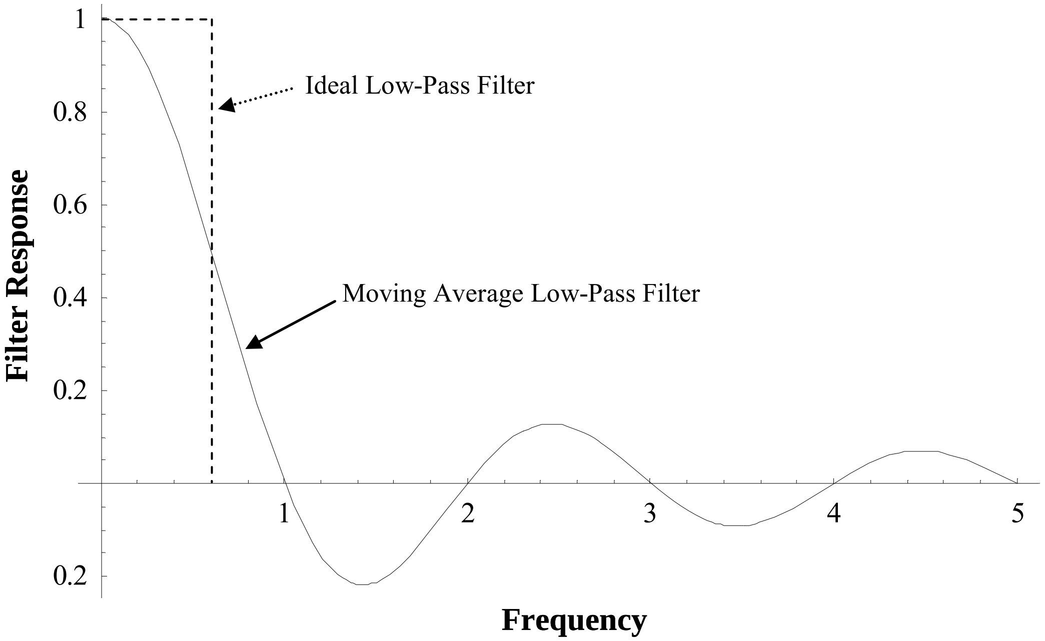

Figure 6. Comparison of the frequency response between an "ideal" low-pass filter (dashed line, *f* = 0.6), and a moving average low pass filter (solid line, $\delta = 1$). The frequency response of the moving average filter is given by the $\mathrm{sinc}_\pi(f)$ function.

3. MONTE CARLO APPLICATION OF SRA

3.1. Background

The previous section provided the theoretical underpinnings of the SRA technique and the moving average filter. In this section we seek to tie these unrelated concepts together with specific application to eigenfunction convergence acceleration in a Monte Carlo neutron transport calculation.

For completeness we begin this section with the derivation of the general neutron transport eigenvalue problem, which is to be solved by Monte Carlo simulation. We start with the fixed source neutron transport equation

$$\begin{aligned} &\hat{\Omega}\cdot\nabla\phi\left(\vec{r},E,\hat{\Omega}\right)+\Sigma_t\left(\vec{r},E\right)\phi\left(\vec{r},E,\hat{\Omega}\right) \\ &\quad -\int_0^\infty dE'\int_{4\pi}d\hat{\Omega}\,\Sigma_s\left(\vec{r},E'\to E,\hat{\Omega}\cdot\hat{\Omega}'\right)\phi\left(\vec{r},E,\hat{\Omega}\right)=Q\left(\vec{r},E,\hat{\Omega}\right). \end{aligned} \tag{40}$$

The notation adopted in Eq. (40) is standard within the nuclear engineering community, where $\phi(\vec{r},E,\hat{\Omega})$ and $Q(\vec{r},E,\hat{\Omega})$ represent the neutron flux and fixed neutron source, respectively, per unit volume, energy and solid angle.

The flux due to a single point source at $\vec{r}_0,E_0,\hat{\Omega}_0$ can be written in terms of a general Green's function, *G*, such that

$$\phi(\vec{r},E,\hat{\Omega}) = G\left(\vec{r}_0 \to \vec{r},\, E_0 \to E,\, \hat{\Omega}_0 \to \hat{\Omega}\right). \tag{41}$$

By superposition it holds that the flux for an arbitrary source distribution in volume *V* is given by

$$\phi(\vec{r},E,\hat{\Omega}) = \int_V d\vec{r}' \int_0^\infty dE' \int_{4\pi} d\hat{\Omega}'\, G\left(\vec{r}' \to \vec{r},\, E' \to E,\, \hat{\Omega}' \to \hat{\Omega}\right) Q(\vec{r}',E',\hat{\Omega}') \,. \tag{42}$$

In a multiplying system, the neutron source is proportional to the neutron flux, given by the relationship

$$Q\left(\vec{r},E,\hat{\Omega}\right) = \frac{1}{k}\frac{\chi(E)}{4\pi}\int_0^\infty dE' \int_{4\pi} d\hat{\Omega}'\, \upsilon\Sigma_f\left(\vec{r},E'\right)\phi\left(\vec{r},E',\hat{\Omega}'\right), \tag{43}$$

where $\chi(E)$ is the energy spectrum of neutrons produced in fission, υ is the number of neutrons produced per fission, Σ_f is the macroscopic fission cross section, and k is a multiplication factor to ensure that the neutron population is at steady state. Substituting the fission source from Eq. (43) into Eq. (42) gives a solution for the flux distribution in a multiplying medium

$$\phi(\vec{r},E,\hat{\Omega}) = \int_V d\vec{r}' \int_0^\infty dE' \int_{4\pi} d\hat{\Omega}\, G\left(\vec{r}' \to \vec{r}, E' \to E, \hat{\Omega} \to \hat{\Omega}\right) \frac{1}{k}\frac{\chi(E)}{4\pi}\int_0^\infty dE'' \int_{4\pi} d\hat{\Omega}''\, \upsilon\Sigma_f\left(\vec{r},E''\right)\phi\left(\vec{r},E'',\hat{\Omega}''\right). \tag{44}$$

Multiplying Eq. (44) by $\upsilon\Sigma_f$ and integrating over energy and angle gives

$$\begin{aligned}&\int_0^\infty dE \int_{4\pi} d\hat{\Omega}\, \upsilon\Sigma_f\left(\vec{r},E\right)\phi\left(\vec{r},E,\hat{\Omega}\right)\\ &\qquad = \frac{1}{k}\int_V d\vec{r}'\, M\left(\vec{r}' \to \vec{r}\right)\int_0^\infty dE \int_{4\pi} d\hat{\Omega}\, \upsilon\Sigma_f\left(\vec{r},E\right)\phi\left(\vec{r},E,\hat{\Omega}\right),\end{aligned} \tag{45}$$

where

$$M\left(\vec{r}' \to \vec{r}\right) = \int_0^\infty dE \int_{4\pi} d\hat{\Omega}\, \upsilon\Sigma_f\left(\vec{r},E\right)\int_0^\infty dE' \int_{4\pi} d\hat{\Omega}'\, G\left(\vec{r}' \to \vec{r}, E' \to E, \hat{\Omega}' \to \hat{\Omega}\right)\frac{\chi(E')}{4\pi}\,. \tag{46}$$

The kernel *M* is itself a Green's function relating the fission neutrons produced at *r* due to a unit source input at *r'*. If Eq. (45) is spatially discretized by partitioning *V* into *N* cells, the kernel *M* becomes an *N*×*N* matrix that relates the integrated scalar flux in each cell to the fission neutron production rate in all cells. This form of the kernel *M* is often referred to as the fission matrix.

For this derivation, however, we wish to avoid the traditional fission matrix formulation and focus on a more general result. Renaming the spatial fission source term

$$s\left(\vec{r}\right) = \int_0^\infty dE \int_{4\pi} d\hat{\Omega}\, \upsilon\Sigma_f\left(\vec{r},E\right)\phi\left(\vec{r},E,\hat{\Omega}\right), \tag{47}$$

allows Eq. (45) to be rewritten

$$s(\vec{r}) = \frac{1}{k}\int_V d\vec{r}'\, M(\vec{r}' \to \vec{r})\, s(\vec{r}). \tag{48}$$

Defining operator A as

$$A[\bullet] = \int_V d\vec{r}'\, M(\vec{r}' \to \vec{r})(\bullet), \tag{49}$$

yields the familiar eigenvalue equation,

$$A[s(\vec{r})] = k\ s(\vec{r}) \tag{50}$$

with the multiplication factor, k, as the eigenvalue.

3.2 Method of Successive Generations

Monte Carlo eigenvalue calculations use an iterative technique referred to as the method of successive generations [9, 10] (or simply source iteration) to estimate the fundamental eigenvalue / eigenfunction solution to the problem,

$$A[s(\vec{r})] = k\ s(\vec{r}), \tag{51}$$

where $s(\vec{r})$ is the spatial distribution of fission sites (eigenfunction), k is the multiplication factor (eigenvalue) for the system, and A is an operator which relates the spatial locations of fission sites between two successive generations.

The Monte Carlo source iteration is an analog of the traditional source iteration techniques used in deterministic neutron transport codes. At the beginning of the first neutron generation, a fixed number of source particles are randomly sampled from a spatial distribution corresponding to an initial source guess $s_0(\vec{r})$. The collection of source particles for a generation is referred to as the "source bank". Each particle in the source bank is independently tracked through the problem geometry until termination, either by absorption or leakage from the system. At each neutron-nucleus collision event the probability of a fission event occurring from the collision is calculated. This probability of fission, along with the expected number of neutrons produced per fission, υ, is used to randomly sample an integer number of fission neutrons produced at the spatial location of the collision. The position and energy of these fission neutrons are stored in the "fission bank" for the generation.

At the end of the generation the fission bank contains a set of fission neutrons produced by the particles stored in the source bank for the generation. In effect, this process is estimating the application of the fission kernel A for each delta point source in the source bank. When applied to all of the source particles we immediately see that the fission bank is a single stochastic realization of the operator A applied to the source bank. In order to express this mathematically we have selected the notation

$$\{f\} = A[\{s\}] \tag{52}$$

where $\{s\}$ is the set of discrete point sources contained in the source bank and $\{f\}$ is a set of discrete fission sites produced by one realization of the operator A applied to the points in $\{s\}$. In this notation, the variables $\{s\}$ and $\{f\}$ are used to denote a realization, or sample, taken according to the associated continuous probability density functions $s(\vec{r})$ and $A[s(\vec{r})]$, respectively .

Using these definitions for the source and fission banks, s and f, it is possible to define a stochastic analog for the power iteration method used to solve the eigenvalue problem in Eq. (51),

$$\{s\}^{(t)} = \frac{\{f\}^{(t-1)}}{\hat{k}_{\text{eff}}^{(t\text{-}1)}} \tag{53}$$

where $\hat{k}_{\text{eff}}^{(t-1)}$ is the Monte Carlo estimate of fundamental eigenvalue produced in generation (t-1). The iterative scheme shown in Eq. (53) is easy to implement in Monte Carlo particle transport codes. At the end of each neutron generation the fission bank is normalized by $\hat{k}_{\text{eff}}^{(t-1)}$ and used as the source bank for the next generation.

The eigenvalue normalization shown in Eq. (53) involves dividing the statistical weight of each particle in $\{f\}$ by the most recent estimate for k_{eff}. However, this approach can lead to problems when $k_{eff} \neq 1$ and the number of fission sites stored in the fission bank is expected to be different from the number of source particles. For example, if $k_{eff} < 1$ the fission bank will, on average, contain fewer sites than the corresponding source bank. Likewise, if $k_{eff} > 1$ the fission bank will usually contain more sites than the source bank. In order to maintain a constant number of particles (i.e. workload) per generation, stratified sampling is used in this case to select a fixed number of fission sites from the fission bank, rather than modifying the weight of sites in the bank. The sampled fission sites then serve as the source bank for the following generation. Since the expected difference between the number of particles in the source and fission banks is proportional to the multiplication factor k_{eff}, the use of stratified sampling for generating a fixed size source bank from the previous fission bank is a form of eigenvalue normalization. The stratified sampling approach works well for critical systems with $k_{eff} > 1$. For subcritical systems, however, the fission bank may contain significantly fewer particles than the source bank. In these cases the stratified sampling method will be forced to reuse fission sites in order to obtain the requested number of sites for the next generation.

The method of successive generations outlined above, or some variant, has been used in all Monte Carlo criticality codes. While this method is well known and widely used, the effects of stochastic fluctuations on the convergence behavior of the basic power method is not well characterized. Furthermore, the nature of the source bank "eigenfunction" (as a collection of source points rather than a traditional vector) prevents the use of most traditional power method acceleration techniques, such as Chebyshev acceleration and Krylov projection. However, we will now demonstrate that some convergence acceleration can be achieved in Monte Carlo criticality calculations by manipulating the fission sites stored in the source and fission banks for a given neutron generation.

3.3. Application of SRA to Monte Carlo Source Convergence

The proposed Monte Carlo source acceleration scheme is a variation on the well known fixed-parameter extrapolation method for accelerating convergence of the power method. Applying the extrapolated power method [11] to solving the eigenvalue problem given in Eq. (51) gives the iterative relationship

$$s^{(t)}(\vec{r}) = \frac{A\left[s^{(t-1)}(\vec{r})\right]}{k_{\text{eff}}^{(t\text{-}1)}} + \alpha\left[\frac{A\left[s^{(t-1)}(\vec{r})\right]}{k_{\text{eff}}^{(t\text{-}1)}} - s^{(t-1)}(\vec{r})\right], \tag{54}$$

where $s^{(t)}$ is the t^{th} estimate of the fission source distribution and α is the extrapolation parameter. As before, a stochastic analog for Eq. (54) can be defined as

$$\{s\}^{(t)} = \frac{\{f\}^{(t-1)}}{\hat{k}_{\text{eff}}^{(t\text{-}1)}} + \alpha\left[\frac{\{f\}^{(t-1)}}{\hat{k}_{\text{eff}}^{(t\text{-}1)}} - \{s\}^{(t-1)}\right]. \tag{55}$$

Notice that the first term in Eq. (55) is the same as the traditional Monte Carlo source iteration, Eq. (53). The second, or extrapolation, term in Eq. (55) uses the residual distribution between the sets $\{s\}^{(t-1)}$ and $\{f\}^{(t\text{-}1)}$ to produce a better estimate for the subsequent source distribution $\{s\}^{(t)}$. The difficulty in applying this result directly lies in estimating the residual distribution from the source and fission banks.

It was previously noted that the initial source bank $\{s\}^{(0)}$ is a fixed number of random samples taken from the continuous distribution $s_0(\vec{r})$. For convenience, let us further assume that the fission bank produced from this initial source, $\{f\}^{(0)}$, is itself a random sample taken from a continuous distribution, $A[s_0(\vec{r})]$. Since $\{s\}^{(0)}$ and $\{f\}^{(0)}$ are both collections of sampled fission sites, there is no convenient way to calculate the residual distribution, $\{f\}^{(0)} - \{s\}^{(0)}$, directly. Instead, we consider a statistical approach for estimating the residual.

Consider a random experiment where N samples x_i are drawn from the probability distribution $p(x)$. From basic probability theory it follows that the ratio of the number of samples N_s that satisfy $a < x_i < b$, to the total number of samples, N, is an unbiased estimator for the integral

$$\text{E}\left[\frac{N_s}{N}\right] = \int_a^b p(x)dx\,. \tag{56}$$

To apply this trivial result to the neutron transport process, let $N^{f}_{\vec{r}',dV}$ and $N^{s}_{\vec{r}',dV}$ represent the number of sites in the source and (normalized) fission bank, respectively, that are located within some volume element $dV = \delta_x \times \delta_y \times \delta_z$ about the location $\vec{r}\,'$. Applying Eq. (56) to the ratio of these quantities to the total number of neutron histories per generation yields

$$\mathrm{E}\left[\frac{N^{f}_{\vec{r}\,',dV}}{N}\right]=\int_{\vec{r}-\delta_x/2}^{\vec{r}+\delta_x/2} dx \int_{\vec{r}-\delta_y/2}^{\vec{r}+\delta_y/2} dy \int_{\vec{r}-\delta_z/2}^{\vec{r}+\delta_z/2} dz\, \frac{A[s(\vec{r}\,')]}{k_{\mathrm{eff}}} \tag{57}$$

and

$$\mathrm{E}\left[\frac{N^{s}_{\vec{r}\,',dV}}{N}\right]=\int_{\vec{r}-\delta_x/2}^{\vec{r}+\delta_x/2} dx \int_{\vec{r}-\delta_y/2}^{\vec{r}+\delta_y/2} dy \int_{\vec{r}-\delta_z/2}^{\vec{r}+\delta_z/2} dz\, s(\vec{r}\,')\ . \tag{58}$$

Subtracting Eq. (58) from Eq. (57) and dividing both sides of the result by the volume element *dV* gives

$$\mathrm{E}\left[\frac{N^{f}_{\vec{r}\,',dV}-N^{s}_{\vec{r}\,',dV}}{\delta_x\,\delta_y\,\delta_z\,N}\right]=\frac{1}{\delta_x\,\delta_y\,\delta_z}\int_{\vec{r}\,'-\delta_x/2}^{\vec{r}\,'+\delta_x/2} dx \int_{\vec{r}\,'-\delta_y/2}^{\vec{r}\,'+\delta_y/2} dy \int_{\vec{r}\,'-\delta_z/2}^{\vec{r}\,'+\delta_z/2} dz \left[\frac{A[s(\vec{r}\,')]}{k_{\mathrm{eff}}}-s(\vec{r}\,')\right] . \tag{59}$$

By inspection, Eq. (59) is equivalent to

$$\mathrm{E}\left[\frac{N^{f}_{\vec{r}\,',dV}-N^{s}_{\vec{r}\,',dV}}{\delta_x\,\delta_y\,\delta_z\,N}\right]=F_{\mathrm{ma}}\left[\frac{A[s(\vec{r}\,')]}{k_{\mathrm{eff}}}-s(\vec{r}\,')\right], \tag{60}$$

where F is a spatial moving average filter with window size $\delta_x \times \delta_y \times \delta_z$. Equation (60) proves that the statistic

$$\hat{R}(\vec{r})=\frac{N^{f}_{\vec{r}\,',dV}-N^{s}_{\vec{r}\,',dV}}{\delta_x\,\delta_y\,\delta_z\,N} \tag{61}$$

is an unbiased estimator for the smoothed residual at location $\vec{r}\,'$. Thus, estimating the smoothed residual at any point $\vec{r}\,'$ is as simple as counting the number of fission sites that occur within $\pm\delta_x/2$, $\pm\delta_y/2$, and $\pm\delta_z/2$ of $\vec{r}\,'$ in both the source and fission banks, and taking the difference between the two. Because the statistic given in Eq. (61) holds for any spatial position $\vec{r}\,'$, it follows that a separate estimate of the smoothed residual can be made for each fission site location $\vec{r}_i$ stored in the fission bank $\{f\}$.

With an estimate for the residual distribution in hand, the final challenge is to apply an analog of Eq. (55) to “extrapolate” the entries in the fission bank based on the pointwise estimate of the smoothed residual. The most straightforward approach is to adjust the relative importance (statistical weight) of the individual sites in the fission bank. The sites in the fission bank are distributed spatially according to the density function $f(\vec{r})$. By applying a weight factor

$$w_i=\frac{g(\vec{r}_i)}{f(\vec{r}_i)}, \tag{62}$$

to each point in the fission bank, and then randomly sampling points from the bank in proportion to their assigned statistical weights it is possible to produce a set of samples that are distributed according to the distribution $g(\vec{r})$. Setting $g(\vec{r})$ equal to the desired source shape $s(\vec{r})$ in Eq. (55) gives a weight modification formula for each point *i* in the fission bank

$$w_i = 1 + \alpha \frac{R(\vec{r}_i)}{f(\vec{r}_i)}, \tag{63}$$

where α is the extrapolation parameter. Replacing the residual, $R(\vec{r})$, and fission bank, $f(\vec{r})$, terms by their statistical estimators yields

$$w_i = 1 + \alpha \frac{\hat{R}(\vec{r}_i)}{\hat{f}(\vec{r}_i)}$$

$$w_i = 1 + \alpha \left(\frac{N^f_{\vec{r}',dV} - N^s_{\vec{r}',dV}}{\delta_x\,\delta_y\,\delta_z\,N} \right) \left(\frac{\delta_x\,\delta_y\,\delta_z\,N}{N^f_{\vec{r}',dV}} \right)$$

$$w_i = 1 + \alpha \left(\frac{N^f_{\vec{r}',dV} - N^s_{\vec{r}',dV}}{N^f_{\vec{r}',dV}} \right). \tag{64}$$

Equation (64) gives the appropriate weight modification formula for SRA source extrapolation in a Monte Carlo eigenvalue calculation, assuming the total number of points stored in the fission and source banks are equal. If the number of sites in the two banks is not equal, then additional normalization terms are required to achieve the desired fission source shape. Once the statistical weighting factors have been applied, stratified sampling is used to draw fission sites from the fission bank for use in the next generation. If done properly, the sampled fission sites should be distributed according to the extrapolated source distribution shown in Eq. (55), thus effectively mimicking source extrapolation in a Monte Carlo simulation.

4. CONCLUSIONS

A new acceleration scheme for the power method iterative eigenfunction solver has been developed. The smoothed residual acceleration (SRA) scheme is based on the well-known source extrapolation method for acceleration. In SRA, the difference, or residual, between successive eigenfunction estimates is calculated at the end of each iteration. This residual term is then passed through a filter to remove high frequency components. A linear combination of the last eigenfunction estimate and the “smoothed” residual is then used to extrapolate a better starting eigenfunction for the next iteration. The filtering step makes the SRA method more stable than traditional fixed-parameter extrapolation methods, allowing more extrapolation between iterations, and theoretically faster convergence.

The primary advantage of the SRA technique is that it can be easily applied to Monte Carlo criticality calculations in order to improve the source convergence rate. A simple algorithm has been developed to apply SRA to Monte Carlo eigenvalue calculations using the method of successive generations. The new algorithm is easy to implement in existing Monte Carlo codes, and may offer significant source convergence acceleration for systems with large dominance ratios. Additional implementation details and supporting numerical results for these conclusions are presented in a companion paper [12], included in these proceedings.

ACKNOWLEDGMENTS

The second author wishes to acknowledge that this research was performed under appointment to the Naval Nuclear Propulsion Fellowship Program sponsored by Naval Reactors Division of the U.S. Department of Energy.

REFERENCES

1. G. E. Whitesides, “A Difficulty in Computing the k-effective of the World,” *Trans. Am. Nucl. Soc.*, **14**, pp. 680 (1968).
2. Y. Naito, J. Yang, “The Sandwich Method for Determining Source Convergence in Monte Carlo Calculation,” *Journal of Nuclear Science and Technology*, **41**, pp. 559-568 (2004).
3. H. J. Shim, C. H. Kim, “Convergence Criterion of Fundamental Mode Fission Source Distribution in Monte Carlo Calculations,” *The Monte Carlo Method: Versatility Unbounded in a Dynamic Computing World*, Chattanooga, Tennessee, April 17−21, on CD-ROM (2005).
4. T. Ueki, F. B. Brown, “Stationarity Modeling and Information-Based Diagnostics in Monte Carlo Criticality Calculations,” *Nuclear Science and Engineering*, **149**, pp. 38-50 (2005).
5. T. Kuroishi, Y. Nomura, “Development of Fission Source Acceleration Method for Slow Convergence in Criticality Analyses by Using Matrix Eigenvector Applicable to Spent Fuel Transport Cask with Axial Burnup Profile,” *Journal of Nuclear Science and Technology*, **40**, pp. 433-440 (2003).
6. T. Yamamoto, Y. Miyoshi, “Reliable Method for Fission Source Convergence of Monte Carlo Criticality Calculation with Wielandt’s Method,” *Journal of Nuclear Science and Technology*, **41**, pp. 99-107 (2004).
7. J. P. Finch, J. S. Hendricks, C. K. Choi, “Vacation Matrix Method for Correct Source Distribution in Monte Carlo Criticality Calculations,” *The Monte Carlo Method: Versatility Unbounded in a Dynamic Computing World*, Chattanooga, Tennessee, April 17−21, on CD-ROM (2005).
8. D. P. Griesheimer, W. R. Martin, J. P. Holloway, “A Functional Expansion Method for Monte Carlo Eigenvalue Calculations,” *The Monte Carlo Method: Versatility Unbounded in a Dynamic Computing World*, Chattanooga, Tennessee, April 17−21, on CD-ROM (2005).
9. I. Lux, L. Koblinger, *Monte Carlo Particle Transport Methods: Neutron and Photon Calculations*, CRC Press, Ann Arbor, MI (1991).
10. J. Lieberoth, “A Monte Carlo Technique to Solve the Static Eigenvalue Problem of the Boltzmann Transport Equation", *Nukleonik*, 11, pp. 213 (1968).
11. S. Nakamura, *Computational Methods in Engineering and Science with Applications to Fluid Dynamics and Nuclear Systems*, Krieger Publishing, Malabar, FL (1986).
12. B.E. Toth, D.P. Griesheimer, “A Novel Source Convergence Acceleration Scheme for Monte Carlo Criticality Calculations, Part II: Implementation & Results,” *Joint International Topical Meeting on Mathematics & Computation and Supercomputing in Nuclear Applications*, Monterey, California, April 15-19, on CD-ROM (2007).