Solving Eigenvalue Response Matrix Equations with Nonlinear Techniques

Jeremy A. Roberts[a,*], Benoit Forget[b]

[a]Department of Mechanical and Nuclear Engineering, Kansas State University, 3002 Rathbone Hall, Manhattan, KS 66506, USA
[b]Department of Nuclear Science and Engineering, Massachusetts Institute of Technology, 77 Massachusetts Avenue, 24-107, Cambridge, MA 02139, USA

Abstract

This paper presents new algorithms for use in the eigenvalue response matrix method (ERMM) for reactor eigenvalue problems. ERMM spatially decomposes a domain into independent nodes linked via boundary conditions approximated as truncated orthogonal expansions, the coefficients of which are response functions. In its simplest form, ERMM consists of a two-level eigenproblem: an outer Picard iteration updates the k-eigenvalue via balance, while the inner λ-eigenproblem imposes neutron balance between nodes. Efficient methods are developed for solving the inner λ-eigenvalue problem within the outer Picard iteration. Based on results from several diffusion and transport benchmark models, it was found that the Krylov-Schur method applied to the λ-eigenvalue problem reduces Picard solver times (excluding response generation) by a factor of 2–5. Furthermore, alternative methods, including Picard acceleration schemes, Steffensen's method, and Newton's method, are developed in this paper. These approaches often yield faster k-convergence and a need for fewer k-dependent response function evaluations, which is important because response generation is often the primary cost for problems using responses computed online (i.e., not from a precomputed database). Accelerated Picard iteration was found to reduce total computational times by 2–3 compared to the unaccelerated case for problems dominated by response generation. In addition, Newton's method was found to provide nearly the same performance with improved robustness.

Keywords:
response matrix, reactor physics, neutron transport, eigenvalue

1. Introduction and Background

Fundamental to reactor modeling is analysis of the steady-state balance of neutrons, described concisely as

$$\mathbf{T}\phi(\vec{\rho}) = \frac{1}{k}\mathbf{F}\phi(\vec{\rho}), \tag{1}$$

where the operator $\mathbf{T}$ describes transport processes, $\mathbf{F}$ describes neutron generation, ϕ is the neutron flux, $\vec{\rho}$ represents the relevant phase space, and k is the eigenvalue, the ratio of the number of neutrons in successive generations.

Since the late 1970's, full core analyses for light water reactors (LWR) have been performed using relatively low fidelity nodal methods based on clever homogenization of phase-space with proven success. However, as current reactors become increasingly heterogeneous due more aggressive fuel loadings and longer cycle lengths in existing LWR's, nodal methods are becoming less applicable, and for new, highly heterogeneous reactor designs, even less so. Although advances in production nodal codes, including use of generalized multigroup SP$_3$ transport with subassembly resolution, address issues related to more complicated designs [1], there likely is limited room for further improvement of the underlying approach. Consequently, a move toward full core analysis techniques that can leverage the high fidelity methods typically used for smaller problems is desired.

[*] Corresponding author; email address: jaroberts@ksu.edu.

1.1. The Eigenvalue Response Matrix Method

One such approach is the response matrix method (RMM), which is based on a spatial decomposition of the global problem of Eq. 1 into local fixed source problems connected by approximate boundary conditions. The response matrix method has been used in various forms since the early 1960's [2]. Using the terminology of Lindahl and Weiss [3], the method can be formulated using explicit volume flux responses, called the "source" RMM, or by using current responses that include fission implicitly and hence are functions of k, known as the "direct" RMM. Although both forms are used in various nodal methods, the source RMM is more widespread. This work is on the the direct RMM, which shall be referred to as the *eigenvalue* response matrix method (ERMM).

Several formulations of ERMM have been proposed since its first use in the 1960's. Here, a rather general approach is described based on expansions of the boundary conditions that couple subvolumes of the global problem, a formalism introduced as early as the work of Lindahl [4] and studied more recently by several authors [5, 6, 7].

Suppose the global problem of Eq. 1 is defined over a volume V. Then a local homogeneous problem can be defined over a subvolume V_i subject to

$$\mathbf{T}\phi(\vec{\rho}_i) = \frac{1}{k}\mathbf{F}\phi(\vec{\rho}_i)\,, \tag{2}$$

and

$$J_{-}^{\text{local}}(\vec{\rho}_{is}) = J_{-}^{\text{global}}(\vec{\rho}_{is})\,, \tag{3}$$

where $J_{-}^{\text{local}}(\vec{\rho}_{is})$ is a function of the incident boundary flux, typically the partial current, which quantifies net flows through a surface.

To represent the local problem numerically, an orthogonal basis, P_n, over the relevant phase space is defined

$$P_n(\vec{\rho}_{is}), \;\; n = 0, 1, \dots N \tag{4}$$

subject to

$$\int P_m(\vec{\rho}_{is})P_n(\vec{\rho}_{is}) = \delta_{mn} d\rho_{is}\,. \tag{5}$$

A response equation is defined

$$\mathbf{T}\phi_i^{ms}(\vec{\rho}_i) = \frac{1}{k}\mathbf{F}\phi_i^{ms}(\vec{\rho}_i) \tag{6}$$

subject to

$$J_{-}^{\text{local}}(\vec{\rho}_{is}) = P_m(\vec{\rho}_{is})\,. \tag{7}$$

The resulting outgoing currents $J_{-}(\vec{\rho}_{is})$ are used to define response functions

$$r_{im's'}^{ms} = \int P_n(\vec{\rho}_{is})J_{i+}^{m}(\vec{\rho}_{is'})d\rho_{is}\,. \tag{8}$$

The quantity $r_{im's'}^{ms}$ has a simple physical interpretation: it is the m'th order response out of surface s' due to a unit incident mth order condition on surface s of subvolume i.

The incident and outgoing currents are expressed as truncated expansions using the same basis

$$J_{is\pm}(\vec{\rho}_{it}) \approx \sum_{n=0}^{N} j_{is_{\pm}}^{n} P_n(\vec{\rho}_{is}) \tag{9}$$

where

$$j_{is_{\pm}}^{n} = \int P_n(\vec{\rho}_{is})J_{\pm}(\rho_{is})d\rho_{is}\,. \tag{10}$$

These coefficients are then represented in vector form as

$$\mathbf{J}_{i\pm} = (j_{i1_{\pm}}^{0}\, j_{i1_{\pm}}^{1} \,\cdots\, j_{i2_{\pm}}^{0}\, j_{i2_{\pm}}^{1} \,\cdots\, j_{iS_{\pm}}^{N})^{\mathsf{T}}\,, \tag{11}$$

and using these together with Eq. 8 yields the nodal balance equation

$$\mathbf{J}_{i+} = \begin{bmatrix} r^{01}_{i01} & r^{11}_{i01} & \cdots \\ r^{01}_{i11} & r^{11}_{i11} & \cdots \\ & & \ddots \end{bmatrix} \begin{bmatrix} j^{0}_{i1_-} \\ j^{1}_{i1_-} \\ \vdots \end{bmatrix} = \mathbf{R}_i \mathbf{J}_{i-} \, . \tag{12}$$

Global balance is defined by the eigenvalue response matrix equation

$$\mathbf{M}\mathbf{R}(k)\mathbf{J}_- = \lambda \mathbf{J}_- \, , \tag{13}$$

where $\mathbf{R}$ is the block diagonal response matrix of $\mathbf{R}_i$, $\mathbf{J}_-$ are vectors containing all incident current coefficients, $\mathbf{M} = \mathbf{M}^{\mathsf{T}}$ is the connectivity matrix that redirects outgoing responses as incident responses of neighbors, superscript T represents the matrix transpose, and λ is the current eigenvalue. If the response matrix $\mathbf{R}$ is conservative (i.e. it strictly maintains neutron balance),

$$\lim_{k \to k^*} \lambda = 1 \, , \tag{14}$$

where k^* is the true eigenvalue. For nonconservative response expansions, the deviation of λ from unity measures discontinuities introduced across node boundaries and may be used to evaluate accuracy of the expansions used (with respect to an infinite expansion).

1.2. Survey of ERMM Implementations

The method defined by Eqs. 2-12 originates from the work of Shimizu et al. [8, 2], which appears to be the first work on response matrix methods (although the authors acknowledged a connection between their work and the earlier and more general theory of invariant imbedding as developed by Bellman et al. [9]). The method was originally based on 1-D diffusion in slab geometry. Aoki and Shimizu extended the approach to two dimensions, using a linear approximation in space to represent boundary currents [10]. A fundamental shortcoming of this early work is what seems to be an assumed value (unity) of the k-eigenvalue when evaluating responses. Typically $k \approx 1$ for nuclear reactors, so the errors observed were only tens of pcm, which may have been deceptively small and not representative of more general cases. In the later 2-D analysis [10], the results compared favorably to fine mesh diffusion calculations.

Weiss and Lindahl generalized ERMM by considering arbitrarily high order expansions of the boundary currents in Legendre polynomials [11], while also accounting for k explicitly. Lindahl also studied expansions of the current, comparing Legendre expansions to an approach that divides the boundary in several segments in which the current is assumed flat [4]. A more complete overview of these approaches can be found in the review by Lindahl and Weiss [3].

These diffusion-based methods rely on semi-analytic solutions to the diffusion equation and hence require homogeneous nodes. Previous scoping studies examined diffusion-based responses using discretized operators [6]. By numerically integrating the diffusion equation, heterogeneous nodes are treated naturally, though no diffusion models with heterogeneous nodes were studied.

In addition to methods based on diffusion theory, previous work applied transport theory for generating responses. Pryor et al. used a hybrid stochastic-deterministic approach based on Monte Carlo and the collision probability method to generate responses [12, 13, 14]. Their work is unique in its definition of the response matrix $\mathbf{R}(k)$ based on a precomputed series expansion. Considering again Eq. 2, the solution ϕ (omitting indices) can be expressed as

$$\phi = \phi_0 + \frac{1}{k}\phi_1 + \frac{1}{k^2}\phi_2 + \ldots \, , \tag{15}$$

where ϕ_i is the flux for the ith neutron generation due to fission. The associated responses can be similarly defined

$$\mathbf{R}(k) = \mathbf{R}_0 + \frac{1}{k}\mathbf{R}_1 + \frac{1}{k^2}\mathbf{R}_2 + \ldots \, . \tag{16}$$

The authors claim this series can be truncated in as few as three terms by estimating the analytical sum, though the accuracy is not specified [14]. When no fissile material is present, $\phi_i = 0$ for $i > 0$, and so $\mathbf{R}(k) = \mathbf{R}_0$.

A somewhat similar approach was developed by Moriwaki et al. [15] in which Monte Carlo is used to generate assembly-level responses for full core analyses. Their method decomposes the response matrix into four physically distinct components: transmission of incident neutrons from one surface to another surface (T), escape of neutrons born in the volume out of a surface (L), production of neutrons in the volume due to neutrons born in the volume (A), and production of neutrons in the volume due to neutrons entering a surface (S). If all indices but the surface are neglected, a current response can be expressed as

$$r_t^s = T_t^s + \frac{1}{k} S^s (L_t + \frac{1}{k} A(L_t + \frac{1}{k} A(L_t + \cdots)) + \cdots) . \tag{17}$$

Like Eq. 16, this infinite sum represents the contributions of each generation to the total response. The matrices **T**, **L**, **A**, and **S** are precomputed, and the full matrix **R** is computed on-the-fly by iteration. In the actual implementation, the volume-dependent responses are unique for each pin in an assembly. Additionally, spatial segmentation is used on boundaries, but angular dependence is neglected.

The more recent extension of Ishii *et al.* improved the angular accuracy by including angular segmentation. However, the resulting amount of data required is quite significant, since the responses are then dependent on spatial segment, angular segment, energy group, and for volume responses, unique pins. Therefore, the approach in Eq. 16 may be more economical because no volume-dependent responses are required. However, computing pin reaction rates would still require volume-dependent responses but would preclude their use in solving Eq. 13.

Other related work has been development of the incident flux expansion method [16, 5]. Initial work by Ilas and Rahnema focused on a variational approach using a basis of Green's functions for each variable with one-dimensional discrete ordinates [16]. Mosher and Rahnema extended the method to two dimensions using discrete Legendre polynomials for space and angle expansions. In addition, they introduced a nonvariational variant that is equivalent to ERMM, but without explicit construction of matrices. Forget and Rahnema extended this nonvariational approach to three dimensions using Monte Carlo with continuous Legendre polynomials in space and angle [17]. In all cases, the responses were precomputed as functions of the k-eigenvalue, and linear interpolation was used to compute responses during global analysis.

1.3. Major Challenge

The application of ERMM to realistic steady state analyses with feedback effects entails several challenges, predominantly the shear number of responses functions and hence transport solves required. These response functions are entirely independent for a given state and k, making ERMM ideal for parallelization.

In the past work on ERMM, responses were precomputed as a function of k and interpolated as needed. In many cases, clever use of symmetry can reduce the amount of required data. For benchmark problems, this reduction is helpful, but as the effects of thermal feedback are included, each node becomes unique and, usually, asymmetric. As such, precomputation of responses would require inclusion of dependence on several variables in addition to k. There seems at this time no completely satisfactory way to parameterize a response function database for accurate steady state analyses. Parameterization becomes even more difficult if burnup is included for cycle analyses. Recent work has attempted to parameterize the responses for steady state analysis of cold critical experiments [18]. While the results are promising (sub-percent errors on pin fission rates), the problem assessed is not entirely representative of the current problems of interest. Consequently, this paper focuses on ERMM implementations suitable for online generation of response functions because, at present, this appears to be the only meaningful manner in which to apply ERMM. However, any improvements developed for online response generation would be readily applicable to generation of response databases should an adequate parameterization scheme be developed in the future.

1.4. Goals

The primary goal of this work is to develop a response matrix method that can efficiently leverage high fidelity deterministic transport methods for solving large-scale reactor eigenvalue problems on a variety of computer architectures. Several solvers that support this goal are developed in this paper, the remainder of which is organized as follows. Section 2 discusses solution of the response matrix equations via fixed point iteration and methods for solving the resulting eigenvalue problem for λ efficiently. Section 3 presents schemes that provide faster convergence in k than standard fixed-point iteration, which is important because the number of k-evaluations defines the number of responses evaluated. A numerical study of the methods applied to several benchmark problems is provided in Section 4, followed by concluding remarks in Section 5.

2. Solving the λ-Eigenvalue Problem

The k-eigenvalue can be interpreted physically as the ratio of neutrons produced in one generation to the previous generation. Alternatively, k can be viewed as the ratio of gains to losses in a given generation, and when applying this interpretation to the response matrix formalism, k can be updated via the process

$$f(k) = k_{n+1} = \frac{\mathbf{F}(k_n)\mathbf{J}_-}{\mathbf{A}(k_n)\mathbf{J}_- + \mathbf{L}(k_n)\mathbf{J}_-} , \tag{18}$$

where $\mathbf{F}(k)\mathbf{J}_-$ is the global fission rate, $\mathbf{A}(k)\mathbf{J}_-$ is the global absorption rate, and $\mathbf{L}(k)\mathbf{J}_-$ is the total leakage rate from global boundaries.

Equations 13 and 18 represent a Picard (fixed point) iteration for the k-eigenvalue. For each new k value, however, the λ-eigenvalue problem defined by Eq. 13 must be solved.

Historically, the method to solve Eq. 13 is equivalent to simple power iteration. For a given $k^{(n)}$, the current vector $\mathbf{J}_-$ is initialized, $\mathbf{MR}(k^n)$ is applied to the $\mathbf{J}_-$, and the resulting vector points closer to the dominant eigenvector of interest. The process repeats until converged.

Unfortunately, the asymptotic convergence rate to the dominant mode is equal to $\ln(1/\rho)$, in which the dominance ratio ρ is defined

$$\rho = |\lambda_2|/|\lambda| , \tag{19}$$

and the eigenvalues of $\mathbf{MR}$ satisfy $\lambda > |\lambda_2| \geq |\lambda_i| , \ \forall \, i > 2$. For many problems, ρ typically falls above 0.99 and is frequently larger than the dominance ratio associated with k.

Because of the large dominance ratio, 1000's of iterations are required to reduce residual norms $||\mathbf{MRJ}_- - \lambda\mathbf{J}_-||$ to within the tolerances typically employed. Chebyshev acceleration has been considered for accelerating convergence, but its utility is severely limited due to the eigenspectrum of $\mathbf{MR}$, a significant portion of which sits away from the real axis. For a representative problem, previous work showed that the expected (theoretical) speedup is limited to approximately two, compared to the factor of 20 gained if the spectrum is completely real [7]. Despite these theoretical limitations, Chebyshev acceleration has been used successfully for solving the inner problem [19]. However, such success is likely highly problem-dependent and subject to significant tuning.

Alternatively, more efficient eigenvalue solvers such as Krylov subspace methods can be used for the λ-eigenvalue problem. Krylov methods are based on generation of a Krylov subspace of dimension n, defined for an $m \times m$ operator $\mathbf{A}$ as

$$\mathscr{K}(n, x_0) \equiv \text{span}\{x_0, \mathbf{A}x_0, \mathbf{A}^2 x_0, \ldots, \mathbf{A}^{m-1}x_0\} , \tag{20}$$

for some initial, possibly random, vector x_0. The fundamental goal of all Krylov subspace methods is to find $x \in \mathscr{K}(n, x_0)$ that "best" solves the system of interest, be it an eigenproblem or linear system, where it is assumed that $n \ll m$.

Using $\mathscr{K}(n, x_0)$ directly is difficult numerically because repeated application of $\mathbf{A}$ sends the initial vector x_0 into the same direction, namely that of the dominant eigenvector of $\mathbf{A}$. Hence, the basis must be orthogonalized. The canonical approach for nonsymmetric operators is Arnoldi's method, which by successive application of the modified Gram-Schmidt process yields the Arnoldi decomposition

$$\mathbf{AV} = \mathbf{VH} + f e_n^{\mathsf{T}} , \tag{21}$$

where $\mathbf{V} \in \mathbb{R}^{m \times n}$ consists of orthonormal columns, $\mathbf{H} \in \mathbb{R}^{n \times n}$ is an upper Hessenberg matrix, e_n is a zero vector with its last entry equal to one, and f is the residual, which is orthogonal to the columns of $\mathbf{V}$.

The eigenvalues of $\mathbf{H}$, called *Ritz values*, tend to be good estimates of eigenvalues of $\mathbf{A}$, and given an eigenpair $(\tilde{\lambda}, y)$ of $\mathbf{H}_n$, the Rayleigh-Ritz estimate of the corresponding eigenvector of $\mathbf{A}$ is defined $x = \mathbf{V}_n y$ and is called a *Ritz vector*. Using these approximate eigenvalues and eigenvectors is the basis of Arnoldi's method.

The eigenpairs of $\mathbf{H}$ are found via a dense eigensolver, such as the QR method. While these dense problems are small (since $n \ll m$), they are solved repeatedly for increasing n until converged. If n becomes too large, the dense methods become too expensive. A more efficient approach is to restart Arnoldi's method. In the *explicitly* restarted Arnoldi method (ERAM), some combination of the existing n Ritz vectors is used to choose a single starting guess, from which a new Arnoldi factorization is generated [20]. An alternative to explicit restart is *implicit* restart, where the desired portion of the spectrum is retained continuously by contracting from a subspace of dimension n to a smaller space of size p and mapping back to the larger space. Several implicit restart schemes exist, and the one used in this paper is the Krylov-Schur (KS) method [21]. KS transforms a general Krylov decomposition (of which the Arnoldi decomposition is a special case) into a Krylov-Schur decomposition, in which $\mathbf{H}$ is a strictly upper triangle matrix. With this decomposition, it is comparatively easier numerically to keep the desired spectral information, and the method tends to be more efficient than other implicitly-restarted algorithms [20].

3. Solving the k-Eigenvalue Problem

The convergence properties of the fixed-point iteration are generally favorable [22], but in many cases, faster convergence in k is desirable. Because responses must be recomputed for each new value of k, methods that minimize the number of unique k values are critical for efficient application of ERMM.

3.1. Accelerating Fixed Point Iteration

In this section, several techniques are studied for accelerating the fixed point iteration in k. All the methods are based on extrapolation with respect to k, and hence no machinery beyond that needed for the fixed point iteration is required.

3.1.1. Regula Falsi and Related Methods

The current eigenvalue, λ, approaches an asymptotic value $\lambda^* \approx 1$ in the limit $k \rightarrow k^*$. For conservative responses with negligible iteration error, λ^* is *exactly* unity. Various schemes have been used to capitalize on this relationship between k and λ. In each of these schemes, an initial guess k_0 is made for which the corresponding λ_0 is found. Subsequently, k_1 is selected, potentially via balance, and λ_1 is computed. All successive values k_n are selected so that $\lambda_n \approx 1$. Such a scheme is often called *regula falsi* or the *method of false points* [4].

Lindahl studied the relationship between λ and k and found that $\tilde{k} = 1/k$ varies quite linearly with $\tilde{\lambda} \propto 1/\lambda$. Lindahl extended the concept by storing three or more pairs for interpolation via higher order polynomials [4].

Anghel and Gheorghu [23] modified the approach of Lindahl by assuming the exponential relation

$$\lambda \propto a e^{b/k} . \tag{22}$$

Because response functions tend to have exponential dependence on k, the authors assumed a similar dependence would also hold true for the current eigenvalue.

A more recent study by Forget and Rahnema [24] rediscovered the relationship between k and λ, referring to λ as the "normalization constant." Moving from a k-update via balance, they assumed the relation $k \propto 1/\lambda$ and observed good convergence without needing to compute the gain and loss operators needed for balance. In theory, the relationship $k \propto 1/\lambda$ is expected. Previous work [22] used one group diffusion theory to show that

$$\frac{d\lambda}{dB} \propto B \tag{23}$$

for small buckling $B = \sqrt{\nu\Sigma_f/k - \Sigma_a}$, which suggests

$$\lambda \approx aB^2 + b \approx \frac{a'}{k} + b' . \tag{24}$$

Lindahl found that $k^{-1} \propto \lambda^{-1}$ produced better results compared to $k \propto \lambda^{-1}$ and $k \propto \lambda$ but did not provide numerical results [4].

These two-term schemes are limited by their dependence on the asymptotic value λ^* being unity, or at least close enough so that $\lambda^* - 1$ is within the convergence criteria. If the responses or inner iterations are poorly converged, or the response expansions are not conservative, the schemes can become unstable.

3.1.2. *Steffensen's Method*

Steffensen's method, which, like the extrapolation schemes, relies on a sequence of evaluations of the fixed point. Steffensen's method can be written as the one step fixed point iteration

$$k_{n+1} = g(k_n) = k_n - \frac{(f(k_n) - k_n)^2}{f(f(k_n)) - 2f(k_n) + k_n} \,. \tag{25}$$

This process has second order convergence, meaning the error diminishes with the square of n. This can be demonstrated by expanding $g(k)$ about the fixed point $g(k^*) = k^*$, yielding

$$g(k) = k^* + \Delta g'(k^*) + \frac{\Delta^2}{2} g''(k^*) + \mathcal{O}(\Delta^3) \,. \tag{26}$$

To be (at least) second order, $g'(k^*)$ must vanish. Here,

$$\begin{aligned} \lim_{k \to k^*} g'(k) &= 1 - \frac{2(k - f(k))(1 - f'(k))}{f(f(k)) - 2f(k) + k} \\ &\quad + \frac{(k - f(k))^2 (f'(f(k)) f'(k) - 2f'(k) + 1)}{(f(f(k)) - 2f(k) + k)^2} \\ &= 1 - (2) + (1) \\ &= 0 \,, \end{aligned} \tag{27}$$

where the second two terms are reduced via L'Hôpital's rule. Hence, $k_{n+1} - k^* = \mathcal{O}(\Delta^2)$, so the method is second order as stated.

In practice, Steffensen's method is highly sensitive to the accuracy of the sequence estimates. Within ERMM, it has been observed that Steffensen's method becomes unstable unless very small tolerances ($\approx 10^{-9}$) are used for solving the λ-eigenvalue problem [7]. Furthermore, once the responses are evaluated for the initial guess, each successive Steffensen iteration requires two response evaluations. The savings gained by second order convergence may or may not outweigh the cost of additional evaluations, depending on the problem. However, Steffensen's method is likely ideal when responses are inexpensive to compute or to interpolate from a precomputed database.

3.2. *Newton Methods*

The eigenvalue response matrix problem has been recognized as nonlinear since it was first solved, but it has not been cast in a form for solution directly by Newton-based methods until quite recently [6, 25].

The eigenvalue response matrix equation, k update equation, and L_2 normalization of J_- can be written as the nonlinear residual

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} (\mathbf{MR}(k) - \lambda \mathbf{I})\mathbf{J}_- \\ \mathbf{F}(k)\mathbf{J}_- - (k\mathbf{L}(k)\mathbf{J}_-) \\ \frac{1}{2} - \frac{1}{2}\mathbf{J}_-^{\mathbf{T}}\mathbf{J}_- \end{bmatrix} = \mathbf{0} \,, \tag{28}$$

and the associated Jacobian is defined

$$\mathbf{f}'(\mathbf{x}) = \begin{bmatrix} (\mathbf{MR} - \lambda \mathbf{I}) & \mathbf{MR_k J}_- & -\mathbf{J}_- \\ (\mathbf{F} - k\mathbf{L}) & (\mathbf{F_k} - k\mathbf{L_k} - \mathbf{L})\mathbf{J}_- & 0 \\ -\mathbf{J}_-^{\mathbf{T}} & 0 & 0 \end{bmatrix} \,, \tag{29}$$

where the k subscripts indicate partial differentiation. For $\mathbf{R}(k)$ of size $m \times m$, the Jacobian is of size $(m+2)\times(m+2)$. Moreover, after one evaluation of the response quantities, only the first $m+1$ rows of the $(m+1)$th column of $\mathbf{f}'$ are not known *a priori*, and that unknown column requires only one additional evaluation of the response quantities to allow for a finite difference approximation of the partial derivatives with respect to k. Hence, like Steffensen's method, Newton's method requires two evaluations of k per iteration, if the latter approximates the derivative via functions evaluated at k and $k + \delta k$. Typically, a value of $\delta k \approx \sqrt{\epsilon_{\text{machine}}} \approx 10^{-8}$ is nearly optimal for minimizing roundoff error. However, at potentially reduced performance, the finite difference can use previous values of k; in this case, convergence would likely improve every iteration as successive k values approach the solution.

3.2.1. Newton's Method

Newton's method [26] solves a nonlinear system via the sequence

$$\mathbf{s} = -\mathbf{f}'(\mathbf{x}^{(n)})^{-1}\mathbf{f}(\mathbf{x}^{(n)}) \tag{30}$$

where **s** is the Newton step, and the Newton update is

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} + l\mathbf{s}\,, \tag{31}$$

with a step length l defined to guarantee a decrease in $||\mathbf{f}(\mathbf{x})||_2$. If a solution $\mathbf{x}^*$ exists, and $\mathbf{f}'$ is Lipschitz continuous near and nonsingular at $\mathbf{x}^*$, then Newton's method is known to exhibit quadratic convergence [26].

For a standard, non-parameterized eigenvalue problem $Ax = \lambda x$, Peters and Wilkinson [27] have shown the associated Jacobian (similar to Eq. 29 without the $(m+1)$th column and row) is nonsingular at the solution if λ is simple (which is true for the dominant mode of interest [3]). However, it does not appear the full Jacobian in Eq. 29 is guaranteed to be nonsingular at the solution, though, in practice, the conditions for singularity have not been observed.

3.2.2. Inexact Newton and JFNK

An *inexact* Newton method uses an approximate linear solve for the Newton step satisfying

$$||\mathbf{f}'(x^{(n)})\mathbf{s} + \mathbf{f}(\mathbf{x}^{(n)})||_2 \leq \eta||\mathbf{x}^{(n)}||_2\,, \tag{32}$$

where η is the "forcing term" that may vary at each iteration [26]. The inexact solution of the Newton step necessarily impacts convergence of Newton's method, but convergence typically remains superlinear. While any iterative linear solver could be used, the focus in this paper is on Krylov solvers, leading to Newton-Krylov methods.

Because solving for **s** involves only the action of the Jacobian, the Jacobian need not be explicitly formed, and Newton-Krylov methods become Jacobian-Free Newton-Krylov (JFNK) methods, for which Knoll and Keyes provide an extensive survey [28]. If the action of **MR** were performed in a matrix-free manner, the same algorithm could be used to evaluate the action of $\mathbf{f}'$ in a fully matrix-free approach.

The Jacobian action is usually applied in JFNK methods using a finite difference approximation; however, because the Jacobian in Eq. 29 is defined almost entirely *a priori*, only a relatively small portion of the action must be approximated via finite differences. This is critical for online response generation, for which evaluation of k-dependent responses is the dominant cost, because each Krylov vector generally represents a perturbed k.

3.2.3. Preconditioning JFNK

The key to effective use of Krylov-based linear solvers is often adequate preconditioning. For JFNK, a preconditioner **M** is in some way "close" to the Jacobian $\mathbf{f}'$ but is easier to construct or apply. Moreover, **M** can be applied either to the left, yielding the system $\mathbf{M}^{-1}\mathbf{f}'\mathbf{s} = -\mathbf{M}^{-1}\mathbf{f}$, or to the right by solving $\mathbf{f}'\mathbf{M}^{-1}\tilde{\mathbf{s}} = -\mathbf{f}$ and setting $\mathbf{s} = \mathbf{M}^{-1}\tilde{\mathbf{s}}$.

4. Numerical Results

In this section, the solvers developed in Section 3 are applied to several benchmark problems to determine which of the various algorithms is best suited for solving response matrix equations. All response matrix calculations were performed using `Serment`, a parallel eigenvalue response matrix code, while all responses were computed using libraries of `Detran`, a deterministic transport code with implementations of the discrete ordinates, method of characteristics, and diffusion approximations, and several modern transport solvers developed specifically for the fixed-source problems characteristic of response generation [29]. `Serment` and `Detran` use several external packages, including PETSc [30] and SLEPc [20] for various linear, nonlinear, and eigenvalue solvers.

Several diffusion and transport benchmarks were studied, including the 2-D and 3-D IAEA [31], the 2-D Biblis [32], and 2-D Koeberg diffusion benchmarks, as well as the 2-D C5G7 [33] and 3-D Takeda [34] transport benchmarks. For all problems, the multigroup dependence was exactly treated while the spatial dependence was expanded in discrete Legendre polynomials [5, 35] DLPs.

For the diffusion problems, each 2-D assembly was discretized using a uniform 20 × 20 spatial mesh, corresponding roughly to 1 cm square cells. For the 3-D IAEA problem, 20 cm cubes were represented by a $10\times10\times10$ spatial mesh to reduce the size of the reference calculation. A mesh-centered finite volume discretization was used for all problems.

For the 2-D C5G7 problem, pin cells were represented with a non-uniform, volume-conserving 7 × 7 mesh. The DLP basis on non-uniform meshes is *not* conservative, meaning λ does not approach unity. For the 3-D Takeda benchmark, 5 cm cubes were represented using a uniform 0.25 cm spatial mesh in all directions. Both problems were modeled using the discrete ordinates method in angle with the diamond difference approximation in space.

For the transport benchmarks, the angular dependence was expanded in DLPs for the angular flux or angular current, or a conservative basis of Chebyshev, Legendre, and/or Jacobi polynomials [35, 19] for the angular flux. For all cases, a product quadrature was employed that exactly integrates the conservative basis moments [35].

When computing responses, no symmetry was considered. For homogeneous problems, exploiting symmetry would reduce the number of responses by a factor of 4 in 2-D or a factor of 6 in 3-D. Such tricks are of course handy for benchmarking, but in reality, reactor assemblies are only symmetric at beginning-of-life (and that is on the order of 1/3 of the core fuel), and even then, only at cold zero power. Once any realistic treatment of temperature feedback is considered, all symmetry is lost, and essentially no insight is to be gained from artificially reducing the problem size.

Unless stated otherwise, convergence is defined by $||\mathbf{f}||_2 \leq 10^{-7}$, where $\mathbf{f}$ is the nonlinear residual of Eq. 28. This criterion makes comparison of Picard and Newton methods straightforward. For the diffusion and Takeda transport benchmarks, `Detran` was used to compute the reference solution, while for the 2-D C5G7, reference results from the original documentation were used.

4.1. *Diffusion Benchmarks*

All the diffusion benchmarks studied have homogeneous assemblies using two (IAEA, Biblis) or four (Koeberg) group data. Although not challenging by current standards, these benchmarks are useful for probing the response matrix solvers as functions of tolerance and expansion order.

4.1.1. *Tolerances and Errors*

A convergence criterion based on one quantity may not be an accurate measure of the error in another quantity. For example, assembly powers for the 2-D IAEA problem were computed for spatial orders of 0 through 4 subject to a tolerance on the residual norm ranging from 10^{-1} down to 10^{-12}. The relative eigenvalue error and maximum relative assembly error for each order as functions of tolerance are shown in Figure 1. The results indicate that the *convergence* error in assembly powers for a given order is negligible compared to the *truncation* error due to the order for tolerances below approximately 10^{-6}. A similar trend appears for the eigenvalue error, but for looser tolerances. For all subsequent analyses, a tolerance of 10^{-7} was selected to ensure only truncation errors affect the solutions.

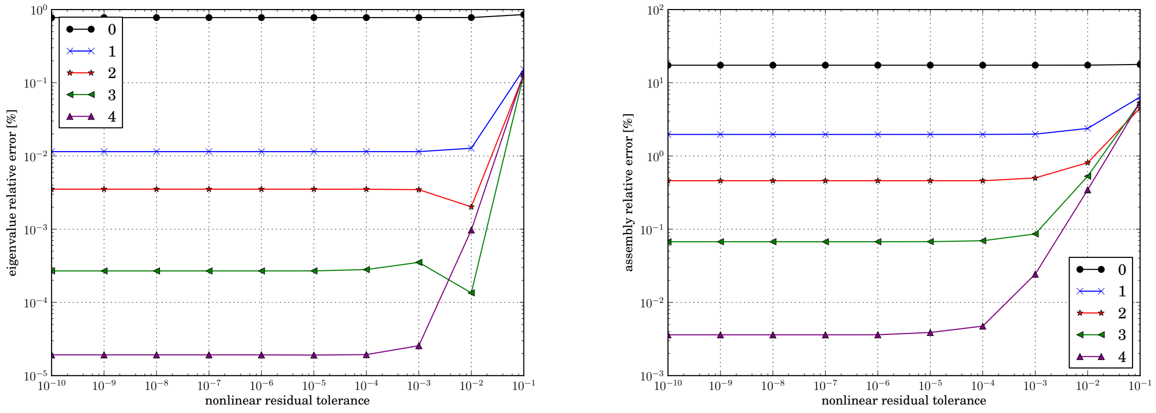
4.1.2. *Orders and Accuracy*

While a detailed discussion of basis accuracies is outside the present scope, it is illustrative to assess convergence of the DLP spatial basis applied to the diffusion benchmarks. Figure 2 shows the maximum relative error in the assembly powers and absolute relative eigenvalue error as functions of spatial order. For the 3-D IAEA problem, two approximations were used. The first is a full expansion of order m in both spatial variables, meaning that on a given side, the two-dimensional expansion is equivalent to the form

$$F(x,y) \approx a + bx + cy + dx^2 + exy + fy^2 + \ldots \,. \tag{33}$$

The second case uses an order reduction scheme that limits the sum of the x and y orders [17]. In this case,

$$F(x,y) \approx a + bx + cy + dx^2 + fy^2 + \ldots \,, \tag{34}$$



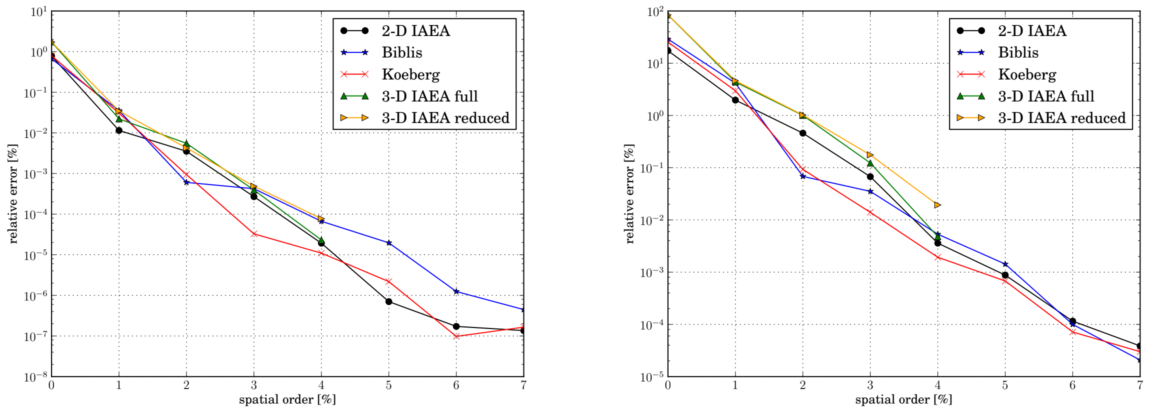
(a) Eigenvalue error.

(b) Assembly power error.

Figure 1: Absolute relative eigenvalue (1a) and absolute maximum relative assembly power (1b) errors for the 2-D IAEA problem as functions of the residual norm tolerance for several spatial orders.

where the cross term exy has been omitted. Previous experience has demonstrated that these cross terms, particularly at high order, have little value, clearly demonstrated in Figure 2.

For all the problems, a fourth order expansion yielded assembly (or nodal, for the IAEA-3D problem) errors below a tenth of a percent and eigenvalue errors on the range of a few pcm. Consequently, a fourth order expansion was selected for use in comparing the solvers in subsequent performance analyses.



(a) Eigenvalue error.

(b) Assembly power error.

Figure 2: Absolute relative eigenvalue (2a) and absolute maximum relative assembly power (2b) errors as functions of spatial order.

4.1.3. Inner Solver Comparison

For the Picard solver, several eigenvalue solvers were investigated to solve the inner λ-eigenvalue problem, including the power (PI), Krylov-Schur (KS), and explicitly-restarted Arnoldi methods (ERAM), which are each implemented in SLEPc [20]. Because convergence of the outer Picard iteration is sensitive to the inner convergence, the tolerance τ_λ of the inner problem was set more tightly at 10^{-10}.

Table 1 provides the number of inner and outer iterations, total computational time, and response generation time for each method and diffusion problem. For all problems, KS outperforms ERAM by a small margin and PI by a factor of two or three. Initial studies demonstrated that IRAM (not included by default with SLEPc) performs at about the same level as KS [7]. Were the tolerance smaller, the improvement of KS over PI would likely diminish.

For the three 2-D problems, the response time constituted a significant portion of the total computational time, ranging from about a third to half depending on the solver. For the 3-D IAEA problem, the global solver was the dominant cost because the diffusion problems underlying the response generation are inexpensive compared to the much larger global problem.

Table 1: Picard inner solver comparison for diffusion problems.

solver	time [a]	r. time [b]	inners	outers
		2-D IAEA		
PI	1.87	0.37	3416	6
KS	0.69	0.41	33	6
ERAM	0.77	0.40	36	6
		Biblis		
PI	2.16	0.68	3437	5
KS	0.97	0.68	31	5
ERAM	1.05	0.69	34	5
		Koeberg		
PI	5.83	1.62	2012	3
KS	2.38	1.66	21	3
ERAM	2.63	1.67	21	3
		3-D IAEA		
PI	910.63	18.77	4427	6
KS	210.91	19.47	75	6
ERAM	294.48	19.75	70	6

[a] Total time [s]
[a] Response generation time [s]

4.1.4. Outer Solver Comparison

Because KS outperformed the other inner solvers investigated, it was selected to study the Picard-based outer iteration schemes. For this study, Picard iteration, along with the accelerated variant based on the *regula falsi* method, was compared to Steffensen's method and Newton's method.

4.1.5. Picard Acceleration

The four Picard acceleration schemes were applied to the 2-D IAEA and Koeberg problems using a fourth order expansion. Figure 3 shows the nonlinear residual as a function of outer iteration for the unaccelerated case along with the four accelerated cases.

Picard iteration alone is a rapidly converging process, but acceleration schemes can further reduce the number of iterations required. Exponential and inverse-inverse extrapolation provide the most robust improvement, although for the Koeberg problem, they did not reduce the number of iterations. The acceleration schemes each critically depend on the limit $\lambda \to 1$, which is satisfied only if the responses are both conservative and computed very accurately. The diffusion equation for each response is small and can be solved nearly exactly via LU factorization. Moreover, the responses are conservative because a uniform mesh and DLP expansion were used. Even so, the linear-inverse and linear-linear suffered from their sensitivity to round-off errors, and even though care was taken when implementing the coefficients, the convergence tolerances used were not tight enough to ensure stability. Because the exponential scheme yielded a slightly smaller final residual norm, it was included for study with the remaining algorithms.

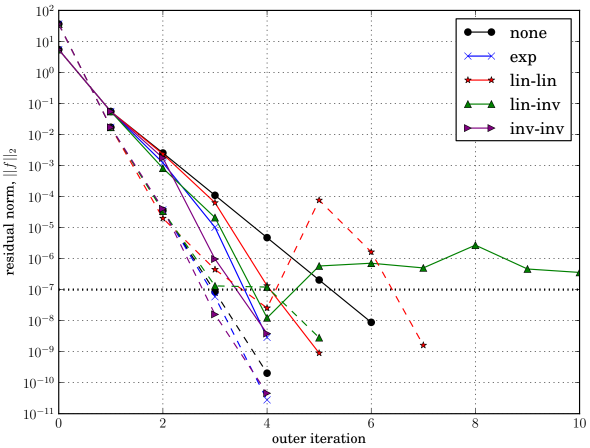


Figure 3: Comparison of Picard acceleration schemes for the 2-D IAEA problem (solid lines) and Koeberg problem (dashed lines).

4.1.6. Newton Variants

For Newton's method, unpreconditioned and preconditioned variants using ILU were studied. The ILU preconditioner is based on an explicit Jacobian constructed either once, using the initial responses, or every iteration, using the updated responses. In all cases, the underlying linear solves were performed with GMRES(30), and the ILU preconditioner was applied with 0 through 2 levels.

Table 2 provides results for the 2-D IAEA and Koeberg problems with a fourth order spatial expansion and 4 × 4 nodes per assembly, corresponding to 184962 and 369922 unknowns, respectively. Somewhat larger problems were used to highlight differences between the preconditioners because the preconditioners offered no benefit for the single node case.

For both problems, ILU(0) preconditioning offered the best performance with respect to time, although higher preconditioner levels led to lower numbers of inner iterations. Failure to update the preconditioner had no discernible effect on the iteration count and yielded lower computational times than when the preconditioner was updated at every iteration. This can be explained by noting that a majority of the Jacobian is relatively insensitive to small changes in k. Given the initial guess for k (unity in these cases) is expected to be pretty close to the final answer, the original Jacobian should be nearly equal to its final value.

4.1.7. Comparing Picard, Steffensen, and Newton

Several metrics can be used to compared the outer solvers. Ultimately, wall clock time is most important in practice. However, the number of iterations of each method, both outer and inner, is also indicative of the algorithm performance, independent of any particular implementation. These data are provided in Table 3 for each of the benchmarks. A fourth order spatial expansion was used for all problems, with order reduction applied to the 3-D IAEA problem. For the 2-D problems, a 4 × 4 node-per-assembly model was used, while for the 3-D IAEA problem, a single node was used, corresponding to 184962 unknowns for the 2-D IAEA and Biblis problems, 369922 for the Koeberg problem, and 988382 for the 3-D IAEA problem.

Table 2: Newton solver ILU preconditioner comparison for diffusion problems.

preconditioner	time[a]	r. time[b]	inners	outers
		2-D IAEA		
none	12.62	0.46	477	4
no update ILU(0)	10.43	0.46	144	4
no update ILU(1)	11.68	0.46	118	4
no update ILU(2)	12.37	0.46	86	4
ILU(0)	10.69	0.46	144	4
ILU(1)	14.07	0.45	118	4
ILU(2)	17.12	0.45	86	4
		Koeberg		
none	45.82	3.29	403	4
no update ILU(0)	43.18	3.34	157	4
no update ILU(1)	52.18	3.35	136	4
no update ILU(2)	58.26	3.35	91	4
ILU(0)	45.61	3.35	157	4
ILU(1)	67.44	3.38	136	4
ILU(2)	89.90	3.35	91	4

[a] Total time [s]
[b] Response generation time [s]

The tested solvers include Picard (P) with and without exponential extrapolation (exp), Steffensen's method (S), and Newton's method (N). For Newton's method, two schemes were examined. The first is the same as used for testing the preconditioning and is based on a Jacobian with k-derivatives computed using a finite difference with a $\Delta k = 10^{-8}$. The second scheme uses the two most recent k values for computing Δk, resulting in a larger Δk and, therefore, a less accurate finite difference. However, the accuracy of the finite difference did not affect the convergence, and for all four problems, the coarse difference yielded the same number of outer iterations as the fine difference while reducing the number of k evaluations by nearly half.

Steffensen's method provided the fastest convergence with respect to outer iterations, but it requires two k evaluations per outer iteration. Picard with exponential extrapolation yielded the lowest computational time and the fewest k evaluations. While Newton's method with a coarse Δk is competitive with respect to k evaluations, the overhead of solving the linear systems is higher than the cost of the λ-eigenvalue problem in the Picard iteration.

4.1.8. Comments

Based on these diffusion analyses, Picard iteration with KS for the inners and exponential extrapolation for accelerating the outers represents an ideal ERMM solver. The Newton methods yield perform nearly as well with respect to k evaluations, but the cost of applying the method is higher per iteration than the Picard variants, implying further work on preconditioning the inner solves is warranted.

4.2. 2-D C5G7

The 2-D C5G7 transport benchmark is a small quarter core model consisting of two UO_2 assemblies and two MOX assemblies, all surrounded by an assembly-width reflector. The model is based on 7 group data and homogenization of the fuel and cladding.

4.2.1. Orders and Accuracy

To assess the accuracy of the response schemes available for transport problems, the C5G7 benchmark was solved using a variety of angular bases. Because a uniform spatial mesh was not used, the DLP spatial expansion converged more slowly for the C5G7 problem than for the uniformly-meshed diffusion problems. For all cases, the response transport calculations were converged to a relative residual norm of 10^{-8}, and the outer calculation was converged to a nonlinear residual norm of 10^{-7}.

Table 4 shows the convergence of the eigenvalue and pin power errors as a function of space-angle order. By third order, the conservative basis yields very limited improvement, most likely due to the DLP spatial basis used. A nonuniform mesh yields a nonconservative DLP expansion that is inaccurate at low orders compared to a conservative expansion. Table 4 shows that both the DLP and Chebyshev bases yield maximum relative pin power errors of slightly more than 2%. These results are in contrast to results reported in Ref. [36], which used a full order spatial basis to eliminate spatial errors and showed that a conservative Chebyshev basis significantly outperforms a DLP basis. Consequently, the present results indicate that spatial expansion errors are dominant by second or third order, and, therefore, a full implementation and systematic study of new spatial bases is warranted.

4.2.2. Solver Comparison

The same global solvers used in Section 4.1.7 were also applied to the 2-D C5G7 problem. In this case, 64 processes were used for response generation, while a single process was used for the outer (λ-eigenvalue) problem.

Table 5 provides the wall time, as well as the total and response time summed over all processes. In addition, the number of inner iterations, outer iterations, and k evaluations are included. Newton's method with the coarse k derivative yielded the best performance, with a wall time of $1.09 \cdot 10^3$ seconds, total time of $6.97 \cdot 10^4$ seconds (summed over all processes), and a solver (excluding response generation) time of 34 seconds. Use of the fine k derivative required more outer iterations and, consequently, significantly more time. Steffensen's method required the fewest outer iterations but at the cost of more k evaluations. Standard Picard iteration readily converged, but extrapolation failed miserably. However, this failure is *completely expected*: the spatial basis is not conservative, so λ does not tend toward unity, and, hence, extrapolation does not apply. This highlights significant value in selecting a conservative basis, because the extrapolated Picard iteration was the best performing method for the diffusion problems.

4.3. 3-D Takeda

The 3-D Takeda benchmark is a simple benchmark, but it allows for in-depth examination of the convergence properties of the basis sets. The model consists of three homogeneous regions using a two group approximation in energy. For this study, the the "rods in" configuration was used.

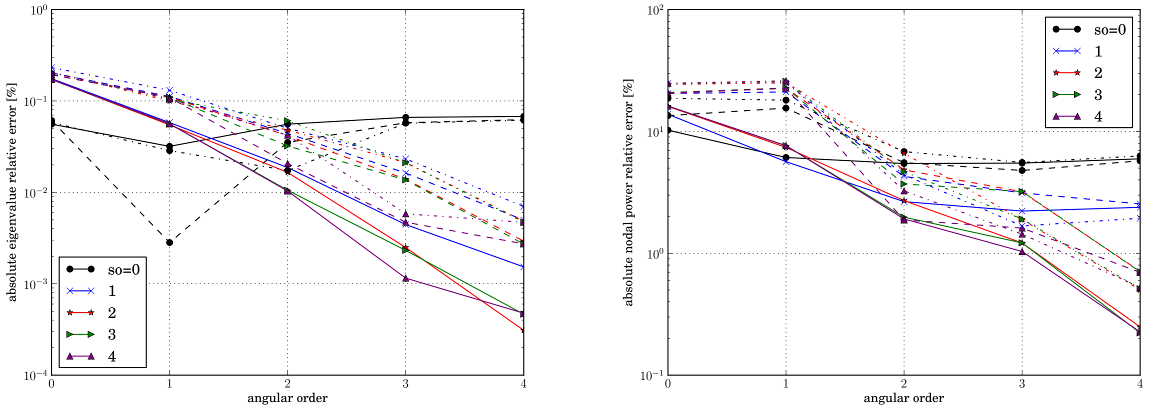
4.3.1. Order Convergence

Figures 4a and 4b provide the absolute relative error in the eigenvalue and the maximum absolute relative error in the nodal powers as a function of angular order for several spatial orders. Order reduction was used in both space and angle. Very little improvement was observed with increasing angular order for a spatial order of zero. For higher spatial orders, an increasing angular order yielded a monotonically decreasing error for both k and the nodal powers. The conservative basis outperformed the DLP variants, yielding nearly sub-1% nodal errors for a third order angular expansion and spatial orders greater than one. DLP-J yielded slightly better nodal powers than DLP-ψ at higher orders but higher k errors for all orders.

For all angular bases, a significant trend observed was the markedly diminishing returns for spatial orders above two. In other words, the most consistent improvement observed, irrespective of angular order and basis, was obtained by shifting from first to second order in space. This trend is reasonable because the nodes are homogeneous, and boundary quantities should, therefore, be relatively smooth functions of space.

4.3.2. Solver Comparison

The same solvers used for the diffusion and 2-D C5G7 problems were applied to the Takeda problem. A second order spatial expansion with a third order angular expansion in the azimuth and polar variables was used. Order reduction was applied to the spatial and angular terms. The problem was run with 64 processes, with one process for the global problem. Table 6 provides the wall time, total time summed over all processors, and the total response function time summed over all processors.



(a) Eigenvalue error.

(b) Nodal power error.

Figure 4: Takeda problem absolute relative eigenvalue (4a) and nodal power (4b) errors as a function of angular order for several spatial orders. The solid lines indicate the conservative basis, while the dashed and dashed-dot lines indicate the DLP basis used to expand the angular flux ψ and current J, respectively.

Similar to the diffusion results, the extrapolated Picard iteration proved to be the most efficient of the solvers studied. Newton's method with the coarse k finite difference yielded just as few k evaluations but with a slightly higher overall cost.

5. Conclusion

Based on the results for both the diffusion and transport problems, Picard iteration with exponential extrapolation appears to be the most efficient method studied, yielding minimum numbers of k evaluations while providing the lowest global solver overhead. However, extrapolation is based on λ converging to unity, and because this is not always guaranteed, Newton's method with the coarse finite difference provides a more consistently robust solver, with nearly as few k evaluations and only relatively small overhead due to the inner linear solves.

Further work on preconditioners for Newton's method will likely reduce the corresponding (global) computation time to levels similar to extrapolated Picard iteration. Even so, the efficacy and simplicity of Picard iteration suggests that care should be taken to select a conservative basis leading to $\lambda = 1$ upon convergence.

Ultimately, the present work supports broader efforts to use ERMM for production level analyses, but significant challenges remain, dominated by the vast number of responses required for realistic modeling. Solvers were developed that reduce the number of k evaluations required and minimize time spent generating responses, while related recent work successfully developed diffusion-based transport preconditioners that significantly reduce the time of individual response calculations [29]. Because responses are independent, parallel computation will be a natural component of any future ERMM implementation. Scoping studies have demonstrated nearly ideal scaling of `Serment` on a small research cluster [35], and ongoing work aims to test `Serment` on leadership-class machines.

References

[1] T. Bahadir, S.-Ö. Lindahl, Studsvik's Next Generation Nodal Code SIMULATE-5, in: Advances in Nuclear Fuel Management IV (ANFM 2009), 2009.

[2] A. Shimizu, K. Monta, T. Miyamoto, Application of the Response Matrix Method to Criticality Calculations of One-Dimensional Reactors, Journal of Nuclear Science and Technology 5 (1963) 11–17.

[3] S.-Ö. Lindahl, F. Weiss, The Response Matrix Method, Advances in Nuclear Science and Technology 13.

[4] S.-Ö. Lindahl, Multi-Dimensional Response Matrix Method, Ph.D. thesis, Kansas State University (1976).

[5] S. W. Mosher, F. Rahnema, The Incident Flux Response Expansion Method for Heterogeneous Coarse Mesh Transport Problems, Transport Theory and Statistical Physics 35 (1) (2006) 55–86.

[6] J. A. Roberts, B. Forget, Solving Eigenvalue Response Matrix Equations using Jacobian-Free Newton-Krylov Methods, in: Proc. International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2011), Rio de Janeiro, Brazil, May 8-12, American Nuclear Society, 2011.

[7] J. A. Roberts, B. Forget, Krylov Subspace Iteration for Eigenvalue Response Matrix Equations , in: Proceedings of the International Conference on the Physics of Reactors (PHYSOR 2012) Knoxville, TN, April 15-20, American Nuclear Society, 2012.

[8] A. Shimizu, Response Matrix Method, Journal of Nuclear Science and Technology 5 (1963) 1–10.

[9] R. Bellman, R. Kalaba, G. M. Wing, Invariant Imbedding and Mathematical Physics. I. Particle Processes, Journal of Mathematical Physics 1 (1960) 280.

[10] K. Aoki, A. Shimizu, Application of the Response Matrix Method to Criticality Calculations of Two-Dimensional Reactors, Journal of Nuclear Science and Technology 2 (4) (1965) 149–159.

[11] Z. Weiss, S.-Ö. Lindahl, High-Order Response Matrix Equations In Two-Dimensional Geometry, Nuclear Science and Engineering 58 (2) (1975) 166–181.

[12] R. J. Pryor, W. E. Graves, Response Matrix Method for Treating Reactor Calculations, in: National Topical Meeting on Mathematical Models and Computational Techniques for Analysis of Nuclear Systems, Vol. 8, Ann Arbor, MI, 1973.

[13] R. J. Pryor, Recent Developments in the Response Matrix Method, in: J. M. Kallfelz, R. A. Karam (Eds.), Advanced Reactors: Physics, Design, and Economics, Pergamon Press, New York, 1975.

[14] J. M. Sicilian, R. J. Pryor, Acceleration Techniques for Response Matrix Methods, in: Joint meeting of the American Nuclear Society and the Atomic Industrial Forum, Vol. 16, 1975.

[15] M. Moriwaki, K. Ishii, H. Maruyama, M. Aoyama, A New Direct Calculation Method Of Response Matrices Using A Monte Carlo Calculation, Journal of Nuclear Science and Technology 36 (10) (1999) 877–887.

[16] D. Ilas, F. Rahnema, A Heterogeneous Coarse Mesh Transport Method, Transport Theory and Statistical Physics 32 (2003) 445–471.

[17] B. Forget, A Three Dimensional Heterogeneous Coarse Mesh Transport Method for Reactor Calculations, Ph.D. thesis, Georgia Institute of Technology (2006).

[18] T. Hino, K. Ishii, T. Mitsuyasu, M. Aoyama, BWR Core Simulator Using Three-Dimensional Direct Response Matrix and Analysis of Cold Critical Experiments, Journal of Nuclear Science and Technology 47 (5) (2010) 482–491.

[19] D. Zhang, F. Rahnema, An Efficient Hybrid Stochastic/Deterministic Coarse Mesh Neutron Transport Method, Annals of Nuclear Energy 41 (2012) 1–11.

[20] V. Hernandez, J. E. Roman, V. Vidal, SLEPc: A Scalable and Flexible Toolkit for the Solution of Eigenvalue Problems, ACM Transactions on Mathematical Software 31 (3) (2005) 351–362.

[21] G. W. Stewart, A Krylov-Schur Algorithm for Large Eigenproblems, SIAM Journal on Matrix Analysis and Applications 23 (2002) 601.

[22] J. A. Roberts, M. Everson, B. Forget, On the Convergence of the Eigenvalue Response Matrix Method, Submitted to the Journal of Applied Mathematics.

[23] V. N. P. Anghel, H. Gheorghiu, An Iteration Strategy for Response Matrix Theory Codes, Annals of Nuclear Energy 14 (5) (1987) 219–226.

[24] B. Forget, F. Rahnema, New Eigenvalue Evaluation Technique in the Heterogeneous Coarse Mesh Transport Method, Transactions of the American Nuclear Society 93 (2005) 511–513.

[25] J. A. Roberts, B. Forget, Nonlinear Coarse Mesh Transport Using the Jacobian-Free Newton-Krylov Method, Transactions of the American Nuclear Society 102.

[26] C. T. Kelley, Iterative Methods for Linear and Nonlinear Equations, Society for Industrial Mathematics, 1995.

[27] G. Peters, J. H. Wilkinson, Inverse Iteration, Ill-Conditioned Equations and Newton's Method, SIAM Review 21 (3) (1979) 339–360.

[28] D. A. Knoll, D. E. Keyes, Jacobian-Free Newton-Krylov Methods: A Survey of Approaches and Applications, Journal of Computational Physics 193 (2) (2004) 357–397.

[29] J. A. Roberts, B. Forget, Multigroup Diffusion Preconditioners for Multiplying Fixed Source Transport Problems, Submitted to Journal of Computational Physics.

[30] S. Balay, W. D. Gropp, L. C. McInnes, B. F. Smith, Efficient Management of Parallelism in Object Oriented Numerical Software Libraries, in: E. Arge, A. M. Bruaset, H. P. Langtangen (Eds.), Modern Software Tools in Scientific Computing, Birkhäuser Press, 1997, pp. 163–202.

[31] Argonne Code Center: Benchmark Problem Book, Tech. Rep. ANL-7416 Supplement 2 (1977).

[32] E. Z. Müller, Z. J. Weiss, Benchmarking with the Multigroup Diffusion High-Order Response Matrix Method, Annals of Nuclear Energy 18 (9) (1991) 535–544.

[33] E. E. Lewis, M. A. Smith, N. Tsoulfanidis, G. Palmiotti, T. A. Taiwo, R. N. Blomquist, Benchmark Specification for Deterministic 2-D/3-D MOX Fuel Assembly Transport Calculations Without Spatial Homogenization, NEA/NSC.

[34] T. Takeda, H. Ikeda, 3-D Neutron Transport Benchmarks, Journal of Nuclear Science and Technology 28 (7) (1991) 656–669.

[35] J. A. Roberts, Advanced Response Matrix Methods for Full Core Analysis, Ph.D. thesis, Massachusetts Institute of Technology (2014).

[36] J. A. Roberts, B. Forget, Phase Space Basis Functions for Response Matrix Methods, in: Submitted for Inclusion in the Proceedings of the International Conference on the Physics of Reactors (PHYSOR 2014), 2014.

Table 3: Outer solver comparison for diffusion problems.

solver	time[f]	r. time[g]	inners	outers	k-evals.
		2-D IAEA			
P[a]	10.26	0.32	76	5	6
P+exp[b]	9.09	0.27	65	4	5
S[c]	11.20	0.39	80	3	7
N+δk[d]	10.35	0.44	144	4	8
N+Δk[e]	10.22	0.28	146	4	5
		Koeberg			
P	9.26	0.56	64	4	5
P+exp	9.06	0.56	62	4	5
S	9.36	0.56	64	2	5
N+δk	10.89	0.92	140	4	8
N+Δk	10.49	0.57	140	4	5
		Biblis			
P	25.65	1.61	54	3	4
P+exp	25.76	1.62	54	3	4
S	29.61	2.02	60	2	5
N+δk	43.10	3.26	157	4	8
N+Δk	42.07	2.05	157	4	5
		3-D IAEA			
P	213.51	17.93	75	6	7
P+exp	167.93	12.79	62	4	5
S	214.07	17.98	75	3	7
N+δk	269.36	20.93	129	4	8
N+Δk	261.27	13.02	129	4	5

[a] Picard

[b] Picard with exponential extrapolation

[c] Steffensen

[d] Newton with fine k difference

[e] Newton with coarse k difference

[f] Total time [s]

[g] Response generation time [s]

Table 4: 2-D C5G7 order convergence. All errors in %, with reference results from `Detran` on same mesh.

basis	order	e_k[a]	$\max\lvert e_i\rvert$[b]	$\frac{\max\lvert e_i\rvert}{p_i^{\text{ref}}}$	$\frac{\sum_i \lvert e_i\rvert}{N}$	$\frac{\sqrt{\sum_i e_i^2}}{N}$	$\frac{\sum_i \lvert e_i\rvert p_i^{\text{ref}}}{N}\bar{p}^{\text{ref}}$
DLP-ψ	0	1.00	33.35	108.87	9.49	0.36	11.11
DLP-ψ	1	0.72	37.78	18.52	2.63	0.12	3.07
DLP-ψ	2	0.13	4.67	7.74	0.92	0.04	1.25
DLP-ψ	3	0.01	2.35	6.24	0.35	0.02	0.37
Chebyshev-ψ	0	2.61	40.73	107.01	9.57	0.37	11.03
Chebyshev-ψ	1	0.07	19.54	11.09	1.01	0.06	1.26
Chebyshev-ψ	2	0.04	2.95	6.70	0.39	0.02	0.37
Chebyshev-ψ	3	0.04	2.42	6.43	0.35	0.02	0.35

[a] $e_k = |k - k^{\text{ref}}|/k^{\text{ref}}$

[b] $e_i = p_i - p_i^{\text{ref}}$, for ith pin

Table 5: Outer solver comparison for 2-D C5G7 problem with first order expansions. Picard with exponential extrapolation fails due to the nonconservative spatial basis, *i.e.* $\lambda \neq 1$.

solver	w. time[f]	time[g]	r. time[h]	inners	outers	k-evals.
P[a]	$1.67 \cdot 10^3$	$1.07 \cdot 10^5$	$1.07 \cdot 10^5$	16	6	7
P+exp[b]	$6.26 \cdot 10^3$	$4.00 \cdot 10^5$	$4.00 \cdot 10^5$	36	21	21
S[c]	$1.73 \cdot 10^3$	$1.11 \cdot 10^5$	$1.11 \cdot 10^5$	16	3	7
N+δk[d]	$4.60 \cdot 10^3$	$2.94 \cdot 10^5$	$2.94 \cdot 10^5$	76	8	16
N+Δk[e]	$1.09 \cdot 10^3$	$6.97 \cdot 10^4$	$6.96 \cdot 10^4$	33	4	5

[a] Picard

[b] Picard with exponential extrapolation

[c] Steffensen

[d] Newton with fine k difference

[e] Newton with coarse k difference

[f] Wall time [s]

[g] Total time summed over all processes [s]

[h] Total response generation time summed over all processes [s]

Table 6: Outer solver comparison for 3-D Takeda problem with second order spatial expansion and third order polar and azimuthal angle expansions.

solver	w. time[f]	time[g]	r. time[h]	inners	outers	k-evals.
P[a]	$9.76 \cdot 10^2$	$6.25 \cdot 10^4$	$5.91 \cdot 10^4$	56	13	14
P+exp[b]	$3.51 \cdot 10^2$	$2.24 \cdot 10^4$	$2.12 \cdot 10^4$	23	4	5
S[c]	$1.05 \cdot 10^3$	$6.72 \cdot 10^4$	$6.36 \cdot 10^4$	58	7	15
N+δk[d]	$7.30 \cdot 10^2$	$4.67 \cdot 10^4$	$4.26 \cdot 10^4$	57	5	10
N+Δk[e]	$3.87 \cdot 10^2$	$2.48 \cdot 10^4$	$2.14 \cdot 10^4$	44	4	5

[a] Picard

[b] Picard with exponential extrapolation

[c] Steffensen

[d] Newton with fine k difference

[e] Newton with coarse k difference

[f] Wall time [s]

[g] Total time summed over all processes [s]

[h] Total response generation time summed over all processes [s]