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 revisits the eigenvalue response matrix method as an approach for solving the k-eigenvalue problem fundamental to nuclear reactor analysis. The method is based on a coarse mesh decomposition of the domain in space. Coarse meshes are linked at their boundaries via incident flux response functions, which are solutions to fixed boundary source problems where the source is a member of an orthogonal basis in all phase space variables. In its most straightforward form, the method consists of a two-level eigenproblem. An outer Picard iteration updates the k-eigenvalue, while the inner eigenproblem imposes neutron balance between coarse meshes. Several eigensolvers, including power iteration and Krylov methods, are evaluated for this inner problem. To accelerate the process, Steffensen's method is investigated. The paper also demonstrates how the method can be recast in a form suitable for solution by Newton's method, and various Newton-Krylov methods are studied using both ILU and multigrid preconditioning. All of the methods are applied to a set of two-dimensional diffusion and transport benchmarks. The results indicate the Steffensen's method solves all the problems in the fewest number of response evaluations.

Keywords:
preconditioning, neutron transport, Krylov, discrete ordinates

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.

For the past several decades, 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, for more aggressive fuel loadings (including MOX) and longer cycle lengths in existing LWR's, these methods are becoming less applicable, and for new, highly heterogeneous reactor designs, even less so. While advances in production nodal codes including use of generalized multigroup SP_3 transport with subassembly resolution address several important issues [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. One such approach is the response matrix method, which is based on a spatial decomposition of the global problem of Eq. 1 into local fixed source problems connected by approximate boundary conditions.

1.1. The Eigenvalue Response Matrix Method

The response matrix method (RMM) 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. While both methods are used in various nodal methods, the former is more widespread; the focus of this work is on the latter, which shall be referred to as the *eigenvalue* response matrix method (ERMM) for clarity.

Various formulations of ERMM have been proposed since its first use. Here, we describe a rather general approach 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

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

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

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

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

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

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

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

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

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 the discontinuities introduced across node boundaries and may be used to evaluate the accuracy of the expansions used (with respect to an infinite expansion).

1.2. Survey of ERMM Methods

The method defined by Eqs. 2-12 has its roots in the work of Shimizu et al. [8, 2], which represents what appears to be the first work on response matrix methods. While independent from it, the authors acknowledge a connection between their work and the earlier and more general theory of invariant imbedding as developed by Bellman *et al.* [9]. This initial work was 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. Since k is typically around unity for nuclear reactors, the errors observed are only tens of pcm, which in general is pretty good, but in their case may be deceptively small. In the later 2-D analysis, the results observed compared favorably to fine mesh diffusion calculations.

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

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

In addition to methods based on diffusion theory, work was also done to use transport theory in generating responses. Pryor *et al.* used what might be now called a hybrid approach, employing Monte Carlo coupled with the collision probability method to generate responses [12, 13, 14]. That work is unique in its definition of the response matrix $\mathbf{R}(k)$. 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. Correspondingly, we can define the associated currents and responses, resulting in

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

The authors claim this series can be truncated in as few as three terms by estimating the analytical sum, though it is not clear with what accuracy [14]. Note that 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 taken by Moriwaki et al. [15] in which Monte Carlo was used to generate responses, nominally for application to full assemblies 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 we neglect all indices but the surface, a current response can be expressed as

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

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 actually unique for each pin in an assembly. Additionally, spatial segmentation was used on boundaries, but angular dependence was neglected.

The more recent extension of Ishii *et al.* addressed the limitation by including angular segmentation, increasing the achievable fidelity. 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. For this reason, it seems as though the approach contained in Eq. 16 might be more economical, as no volume-dependent responses are required. Of course, obtaining pin reaction rates would require such responses to be available but would preclude their use in solving Eq. 13.

Other related work has been development of the incident flux expansion method [16, 5]. The initial work by Ilas and Rahnema focused on a variational approach using a basis consisting of Green's functions for each variable using one-dimensional discrete ordinates [16]. Mosher and Rahnema extended the method to two dimensions, again using discrete ordinates, and used discrete Legendre polynomials for space and angle expansions. Additionally, they introduced a nonvariational variant that is equivalent to ERMM, though without explicit construction of matrices. Forget and Rahnema further 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

To apply ERMM even to realistic steady state analyses including feedback effects entails several challenges, the chief of which is the shear number of responses functions and hence transport solves required. Of course, these response functions are entirely independent for a given state and k, and so parallelization is a natural part of the solution.

In the most recent work, responses were pre-computed as a function of k and interpolated as needed. In many cases, clever use of symmetry can further minimize the required data. For benchmark problems, this is sensible, but as the effects of thermal feedback are included, each node becomes unique. As such, precomputation of responses would require dependence on several variables, in addition to k, be included in some manner. There seems at this time no completely satisfactory way to parameterize a response function database for accurate steady state analyses. The problem is further exacerbated 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—though which error and what norm used are not clearly noted), the problem assessed is not entirely representative of the problems of interest here.

Consequently, the major thrust of this thesis focuses on ERMM implementations suitable for on-the-fly generation of response functions, as this seems at present to be the only meaningful manner in which to apply ERMM. Of course, any successes achieve in this context would be readily applicable for 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. A secondary goal is to produce transport algorithms tailored to the relatively small problems associated with response function generation. To support these goals, a number of specific subtopics are addressed, each of which is described as follows.

Once the responses defining Eq. 13 are defined, an iterative scheme is required to solve for the boundary coefficients and eigenvalue. The most common approach historically has been to use the power method to solve the global balance equation for J, with a corresponding k-update based on neutron balance, leading to a fixed point iteration. Within this fixed point iteration, more efficient schemes can be used to solve the global balance equation. However, in practice the most expensive part of the computation is likely to be the response functions. If these are computed on-the-fly (that is, for each new k), a method that minimizes the number of k evaluations is highly desirable.

Solution of the response matrix equations via fixed point iteration, including an analysis of the convergence of the iteration, is discussed in Chapter **??**. In Chapter 2, efficient solvers for the inner λ eigenvalue problem are discussed. Additionally, methods to minimize k evaluations are developed, including acceleration of the fixed point iteration via Steffensen's method and solution of the response matrix equations via inexact Newton methods. Chapter 3 provides a numerical study of the the methods of Chapter 2 as applied to several benchmark problems.

2. Accelerating λ- and k-Convergence

Recall the global eigenvalue problem, expressed mathematically as

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

The k-eigenvalue can be interpreted physically as the ratio of neutrons produced in one generation to the previous generation. This can alternatively be viewed as the ratio of gains to losses in a given generation. Applying this interpretation to the response matrix formalism, we can update k via the process

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

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.

Equation 18 represents a Picard (fixed point) iteration for the k-eigenvalue. The convergence properties of this iteration are generally favorable [**?**], but here, we seek more efficient schemes. In particular, we develop more efficient methods for solving the inner λ-eigenvalue problem within a fixed point iteration. Additionally, we seek alternative iteration schemes that lead to fast convergence of k and thus minimize the number of response function evaluations needed. This is critical because for most problems, the computation of responses is by far the most time-intensive component of the process.

2.1. Solving the Inner Eigenvalue Problem

In the fixed point iteration analyzed in Chapter **??**, the current eigenvalue problem

$$\mathbf{MR}(k)\mathbf{J}_- = \lambda\mathbf{J}_- \tag{13}$$

must be solved for global balance, after which k can be updated. Historically, the method to solve Equation 13 amounts to simple power iteration. For a given $k^{(n)}$, the current vector is guessed and normalized, $\mathbf{MR}(k^n)$ is applied to the guess, and the resulting vector points closer to the dominant eigenvector of interest. The result is normalized, and the process repeats until converged.

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

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

and the eigenvalues of **MR** satisfy $\lambda > |\lambda_2| \geq |\lambda_i|, \ \forall\, i > 2$. For many problem, ρ typically falls above 0.99, significantly larger than the dominance ratio associated with k.

Due to the large ρ, 1000's of iterations are required to reduce residual norms $||\mathbf{MRJ}_- - \lambda\mathbf{J}_-||$ to within the tolerances used in our later analysis. Chebyshev acceleration was considered for accelerating convergence, but its utility is severely limited due to the nature of the eigenspectrum of **MR**, a significant portion of which sits away from the real axis . For a typical problem, our previous work showed that the expected (theoretical) speedup is limited to about 2, as opposed to the factor of 20 gained were the spectrum completely real [7]. Despite these theoretical limitations, Chebyshev acceleration was recently used successfully for solving the inner problem [19]. However, it seems likely its success would be highly problem-dependent and subject to significant tuning, two features we would ultimately choose to avoid.

2.1.1. Krylov Subspace Methods

As an alternative to the power method, we investigate Krylov subspace methods for the eigenvalue problem, the most well-known of which is the Arnoldi method. The Arnoldi method entails generation of the Krylov subspace defined by Eq. **??**. The Arnoldi process is used as in GMRES to obtain the upper Hessenberg matrix $\mathbf{H} \in \mathbb{R}^{n\times n}$. It turns out that the eigenvalues of **H**, called "Ritz values," tend to be good estimates of eigenvalues of **A**, and given an eigenpair $(\tilde{\lambda}, y)$ of $\mathbf{H}_m$, the Rayleigh-Ritz estimate of the corresponding eigenvector of **A** is defined $x = \mathbf{V}_m y$, and is called a "Ritz vector."

The eigenpairs of **H** are found via a dense eigensolver, such as the QR method. While this is done on a small system (since $m \ll n$), it is done repeatedly for increasing m until some criterion is met. If m becomes too large, the dense methods become too expensive. To circumvent this issue, the Arnoldi method must be restarted. In the *explicitly* restarted Arnoldi method (ERAM), some combination of the existing m Ritz vectors is used to choose a single starting guess, from which a new Arnoldi factorization is generated. There are several ways to do this; in this work, the default SLEPc implementation is used, described in Ref. [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 m to a smaller space of size p and mapping back to the larger space. This is done via implicitly-shifted QR, and leads to the implicitly-restarted Arnoldi method (IRAM) [21]. An alternative to IRAM used in this work is the Krylov-Schur (KS) method [22], which transforms a general Krylov decomposition (of which the Arnoldi decomposition is a special case) into a Krylov-Schur decomposition, where the upper Hessenberg matrix **H** above becomes a strictly upper triangle matrix **T**. Using this decomposition, it is comparatively easier numerically to keep the desired spectral information, and the method is apparently often more efficient than IRAM. As for ERAM, the default SLEPc implementation of KS is used; the method and its implementation in SLEPc are described in Ref. [23].

2.2. Accelerating the Fixed Point Iteration

In the sections to follow, we investigate several techniques for accelerating the outer fixed point iteration for k. All the methods are based on some form of extrapolation with respect to k, and hence no machinery beyond that needed for the fixed point iteration is required for their use.

2.2.1. Regula Falsi and Related Methods

Recall that the current eigenvalue λ approaches unity in the limit $k \to k^*$. For conservative responses with negligible iteration error, λ tends to *exactly* unity. Various schemes have been used to capitalize on this relationship. In each case, 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 [24] modified the approach of Lindahl by assuming the exponential relation

$$\lambda \propto a e^{b/k} . \tag{20}$$

Because response functions tend to have exponential dependence on k, they assumed that would also apply to the current eigenvalue. As the results below will indicate, this form compares favorably with Lindahl's.

A more recent study by Forget and Rahnema [25] rediscovered the relationship between k and λ, the latter of which they denoted the "normalization constant." Moving from a k-update via balance, they assumed the relation $k \propto 1/\lambda$ and found good results. In theory, we might expect this to be the case. Expanding the result of Eq. **??** for small B, we find that

$$\frac{d\lambda}{dB} \propto B , \tag{21}$$

suggesting that

$$\lambda \approx aB^2 + b \approx \frac{a'}{k} + b' . \tag{22}$$

Of course, this is based on a one group approximation in the limit of very small nodes. Interestingly, Lindahl found that $k^{-1} \propto \lambda^{-1}$ produced better results, having compared to both $k \propto \lambda^{-1}$ and $k \propto \lambda$ (though without giving numerical results) [4].

All four two-term schemes have been implemented and are assessed in the results to follow. A key limitation of these schemes is that they depend on λ getting to unity, or at least close enough so that its departure from unity is within the convergence criteria used. If either the responses or the inner iterations are poorly converged, or the response expansions are not conservative, the schemes can become unstable.

2.2.2. Steffensen's Method

A similar approach to those described above is Steffensen's method, which, like the others, relies on a sequence of evaluations of the fixed point. Steffensen's method is most easily motivated via use of Aitken's δ^2 process. Consider a monotonic sequence

$$\{k\} \equiv \{k_0, k_1, k_2, \ldots, k_n, \ldots\} \tag{23}$$

such that

$$\lim_{n\to\infty} k_n = k^* . \tag{24}$$

If the sequence is linearly convergent, then

$$\frac{k_{n-1} - k^*}{k_{n-2} - k^*} \approx \frac{k_n - k^*}{k_{n-1} - k^*} . \tag{25}$$

Consequently,

$$(k_{n-1} - k^*)^2 \approx (k_n - k^*)(k_{n-2} - k^*) \tag{26}$$

yielding Aitken's δ^2 approximation

$$k^* \approx k_n^* = k_{n-2} - \frac{(k_{n-1} - k_{n-2})^2}{k_n - 2k_{n-1} + k_{n-2}} . \tag{27}$$

Suppose the sequence is generated by a series of Picard iterations as defined in Eq. 18. If the improved estimate k_n^* is used in place of k_n, the resulting iteration is referred to as Steffensen's method. Since this is akin to extrapolation by a second order polynomial fit, we might expect the improved estimate yields an iterative scheme of second order, which is indeed true for Steffensen's method.

Steffensen's method can be written as the one step fixed point iteration

$$k' = g(k) = k - \frac{(f(k) - k)^2}{f(f(k)) - 2f(k) + k} . \tag{28}$$

Following Section **??**, we expand $g(k)$ about the fixed point k^*, yielding

$$g(k) = g(k^*) + \Delta g'(k^*) + \frac{\Delta^2}{2} g''(k^*) + \mathcal{O}(\Delta^3) . \tag{29}$$

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

$$\begin{aligned} \lim_{k\to k^*} g'(k) &= 1 - \frac{2(k - f(k))(1 - f'(k))}{f(f(k)) - 2f(k) + k} \\ &\quad + \frac{(k - f(k))^2 (f'(f(k))f'(k) - 2f'(k) + 1)}{(f(f(k)) - 2f(k) + k)^2} \\ &= 1 - (2) + (1) \\ &= 0 , \end{aligned} \tag{30}$$

where the second two terms are reduced via L'Hôpital's rule.

In practice, Steffensen's method is highly sensitive to the accuracy of the sequence estimates. Within the 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.

Furthermore, note that once the responses are evaluated for the initial guess, each successive Steffensen iteration requires two response evaluations. In general, the savings gained by second order convergence may or may not outweigh the cost of additional evaluations depending on the problem.

2.3. Solving ERME's via Newton's Method

The eigenvalue response matrix problem has been recognized as a nonlinear problem since it was first solved, but it does not appear it has been cast in a form for solution directly by Newton-based methods until quite recently [26].

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

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} (\mathbf{MR}(k) - \lambda\mathbf{I})\mathbf{J}_- \\ \mathbf{F}(k)\mathbf{J}_- - (k\mathbf{L}(k)\mathbf{J}_-) \\ \frac{1}{2}\mathbf{J}_-^{\mathbf{T}}\mathbf{J}_- - \frac{1}{2} \end{bmatrix} = \mathbf{0}\,, \tag{31}$$

and the associated Jacobian is defined

$$\mathbf{f}'(\mathbf{x}) = \begin{bmatrix} (\mathbf{MR} - \lambda\mathbf{I}) & \mathbf{MR_k}\mathbf{J}_- & \mathbf{J}_- \\ (\mathbf{F} - k\mathbf{L}) & (\mathbf{F_k} - k\mathbf{L_k} - \mathbf{L})\mathbf{J}_- & 0 \\ \mathbf{J}_-^{\mathbf{T}} & 0 & 0 \end{bmatrix}. \tag{32}$$

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 close to optimal. However, at likely reduced performance, the finite difference can make use of previous values of k; in this case, convergence would likely improve every iteration as each k is closer to the solution.

2.3.1. Newton's Method

Newton's method [27] solves a nonlinear system via the sequence

$$\mathbf{s} = -\mathbf{f}'(\mathbf{x}^{(n)})^{-1}\mathbf{f}(\mathbf{x}^{(n)}) \tag{33}$$

where $\mathbf{s}$ is the Newton step, and the Newton update is

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} + l\,\mathbf{s}\,, \tag{34}$$

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 [27].

For the nonlinear residual of Eq. 31 and Jacobian of Eq. 32, these constraints appear to be true in practice. For a standard, non-parameterized eigenvalue problem $Ax = \lambda x$, Peters and Wilkinson [28] have shown the associated Jacobian (akin to Eq. 32 without the 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 this is always true for the full Jacobian in Eq. 32, though in practice the conditions for singularity have not been encountered.

2.3.2. Inexact Newton and JFNK

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

$$||\mathbf{f}'(x^{(n)})\mathbf{s} + \mathbf{f}(\mathbf{x}^{(n)})||_2 \leq \eta||\mathbf{x}^{(n)}||_2\,, \tag{35}$$

where η is the "forcing term" that may vary at each iteration [27]. The inexact solution of the Newton step necessarily impacts convergence of Newton's method, but typically convergence remains superlinear. In general, any iterative method for a linear system could be used, but we focus on the specific application of Krylov solvers, yielding Newton-Krylov methods.

Since solving for $\mathbf{s}$ involves only the action of the Jacobian, the Jacobian need not be explicitly formed. Then Newton-Krylov methods become Jacobian-Free Newton-Krylov (JFNK) methods, for which Knoll and Keyes provide an extensive survey [29]. If the action of $\mathbf{MR}$ were performed in a matrix-free manner, the same algorithm could be used in evaluating the action of $\mathbf{f}'$ for a fully matrix-free approach.

Often in the context of JFNK, the action of the Jacobian is approximated as a finite difference approximation; however, since the Jacobian in Eq. 32 is defined almost entirely *a priori*, only a relatively small portion of the action need be approximated via finite differences. This is critical for on-the-fly response generation, for which evaluation of k-dependent responses is the dominant cost, since each Krylov vector generally represents a perturbed k.

2.3.3. Preconditioning JFNK

The key to effective use of Krylov-based linear solvers is often adequate preconditioning. For JFNK, a preconditioner $\mathbf{M}$ is in some way "close" to the Jacobian $\mathbf{f}'$ but is easier to construct or apply. Moreover, $\mathbf{M}$ 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 $\mathbf{s} = \mathbf{M}^{-1}\tilde{\mathbf{s}}$.

In the numerical studies of the next chapter, we employ incomplete factorization using an approximate Jacobian matrix as a basis. The easiest choice is to use a full Jacobian computed using the initial guess (which itself has been preconditioned by a single coarse Picard iteration), possibly omitting the partial derivatives to save on a function evaluation. While this approximate Jacobian requires the explicit construction of $\mathbf{MR}$, this is relatively easy to do given the extreme sparsity of $\mathbf{M}$.

3. Comparison of ERME Solvers

In this chapter, the solvers developed in Chapter 2 are applied to several benchmark problems. The purpose is largely

two-fold. While previous chapters have discussed in some depth the accuracy of various approximations, a further comparison based on somewhat larger, more representative problems is valuable. Secondly, and more importantly, we aim to assess which of the various algorithms is best suited for solving response matrix equations.

3.1. Diffusion Benchmarks

Several diffusion benchmarks were first used to investigate the various response matrix algorithms previously described. While such benchmarks do not represent particularly "challenging" problems by modern standards, they are simple to model and provide easy application problems for testing the solvers (and expansions schemes) to a fully converged state.

The problems considered are the 2-D and 3-D IAEA benchmarks, and the 2-D Biblis problem, all two-group problems, and the four-group, 2-D Koeberg benchmark. Each problem is based on homogenized assemblies, and brief specifications are provided in Appendix **??**.

For all problems, the multigroup dependence is exactly treated while the spatial dependence is expanded in DLPs. Each 2-D assembly is discretized using a 20×20 spatial mesh, corresponding roughly to 1 cm square cells. For the 3-D IAEA problem, 20 cm cubes are represented by a $10 \times 10 \times 10$ spatial mesh to reduce the size of the reference calculation. While the discretizations used are not fully spatially-converged grids, they are adequate for the present study. The reference solution uses the same mesh spacing for consistency.

When computing responses, no symmetry is 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 hence there is essentially no insight to be gained from artificially reducing our problem size.

Unless stated otherwise, convergence is defined by $||\mathbf{f}||_2 \leq 10^{-7}$, where $\mathbf{f}$ is the nonlinear residual defined by Eq. 31. Using this criterion makes comparison of Picard and Newton methods much more straightforward. The reference fine mesh solution uses a tolerance of 10^{-12}, though this applies to the residual $\mathbf{A}\phi - \lambda\phi$ of the fine mesh eigenvalue problem.

3.1.1. Tolerances and Errors

As was discovered in early chapters, a convergence criterion based on one quantity may not be an accurate measure of the error in some other quantity. To illustrate, the 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 for each order as a function of tolerance is shown in Figure 1, while the maximum relative assembly error for each order is shown in Figure 2. The latter figure indicates that the *convergence* error in assembly powers for a given order is vanishingly small compared to the *truncation* error due to the order for tolerances below about 10^{-6}. A similar trend appears for the eigenvalue error, but for even looser tolerances. For the subsequent diffusion analyses, a tolerance of 10^{-7} was selected to ensure only truncation errors affect the solutions observed.

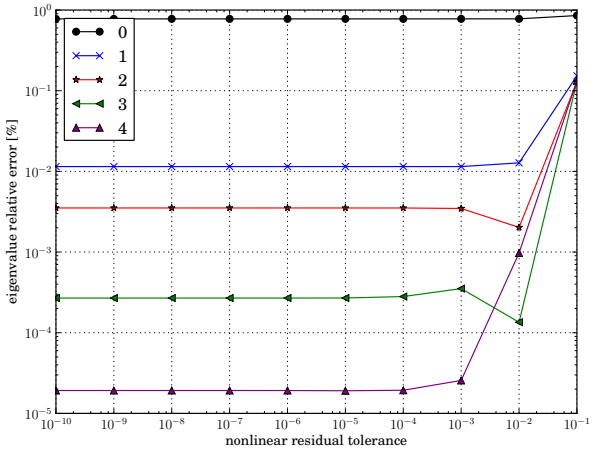

Figure 1: Absolute relative eigenvalue error for the 2-D IAEA problem as a function of residual norm tolerance for several spatial orders .

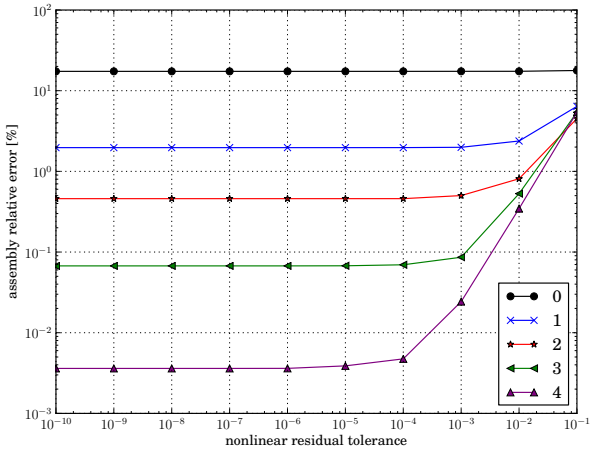

Figure 2: Maximum absolute relative assembly power error for the 2-D IAEA problem as a function of residual norm tolerance for several spatial orders.

3.1.2. Orders and Accuracy

While some attention was paid to the accuracy of spatial expansions in Chapter **??**, it is illustrative to assess convergence of the DLP basis applied to the diffusion benchmarks. Figure 4 shows the maximum relative error in the assembly

powers as a function of spatial order, while Figure 3 does the same for the absolute relative error in the eigenvalue. For the 3-D IAEA problem, two cases are performed. The first uses a full expansion of order m in both spatial variables. That means on a given side, the two-dimensional expansion is equivalent to the form

$$F(x,y) \approx a + bx + cy + dx^2 + exy + fy^2 + \dots . \qquad (36)$$

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

$$F(x,y) \approx a + bx + cy + dx^2 + fy^2 + \dots , \qquad (37)$$

where the cross term exy has been omitted. Previous experience has demonstrated these cross terms, particularly at high order, have little value, and this is demonstrated quite strongly in the results of Figures 3 and 4.

For all the problems, a fourth order expansion yields 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.

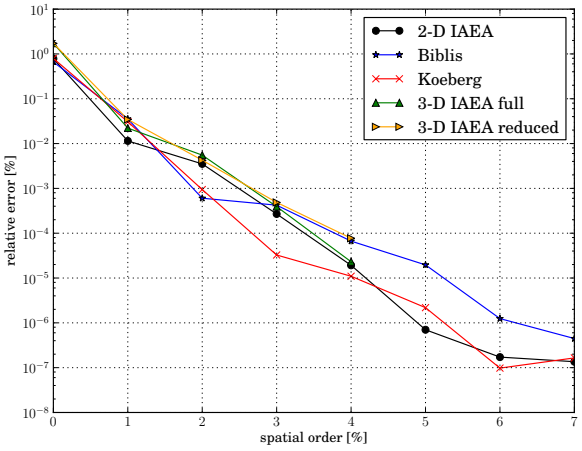

Figure 3: Absolute relative eigenvalue error as a function of spatial order.

3.1.3. Inner Solver Comparison

For use in 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), each of which is implemented in SLEPc [30]. Because convergence of the outer Picard iteration is somewhat 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 and total computational time for each method for each of the diffusion problems. The computational time is further divided into the global (response matrix) time and the

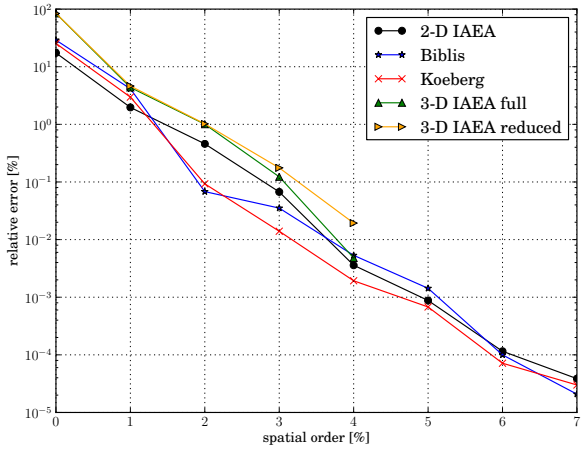

Figure 4: Maximum absolute relative assembly power error as a function of spatial order.

response (diffusion solver) time. 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, and were the tolerance greater, KS would likely perform increasingly better.

For the three 2-D problems, the response time constitutes a significant portion of the total computational time, range from about a third to half depending on the solver. For the 3-D IAEA problem, the global solver becomes the dominant cost. This makes some sense, as the diffusion problems underlying the response generation are quite cheap compared to the much larger global problem.

3.1.4. Outer Solver Comparison

Because KS was the best of the inner solvers investigated, it was selected for studies of 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.

3.1.5. Picard Acceleration

The four Picard acceleration schemes were applied to the 2-D IAEA and Koeberg problems using a fourth order expansion. Figure 5 shows the nonlinear residual as a function of outer iteration for the unaccelerated case along with the four accelerated cases.

As the numerical results of the last section and of Chapter **??** show, the Picard iteration by itself is a quickly converging process. However, the acceleration schemes can offer some improvement. Here, exponential and inverse-inverse extrapolation provide the most robust improvement, though they actually provide no reduction in iterations for the Koeberg problem. Recall that all the schemes depend critically on the limit $\lambda \to 1$, and this only occurs if the responses are

solver	time [s]	r. time [s][a]	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] Response generation time

Table 1: Picard inner solver comparison for diffusion problems.

computed very accurately and the responses are conservative. Here, the diffusion responses are computed essentially by LU factorization since they are such small problems and are conservative based on the uniform mesh and DLP expansion. Even so, the linear-inverse and linear-linear suffer from their sensitivity to round-off errors, and while care was taken when implementing the coefficients, the convergence tolerances used are apparently not tight enough to ensure stability. Because the exponential scheme yields a slightly smaller final residual norm, it was included for study with the remaining algorithms.

3.1.6. Newton Variants

For Newton's method, both an unpreconditioned and ILU-preconditioned variant 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 using a fourth order spatial expansion and 4×4 nodes per assembly. This yields a somewhat larger problem that better illustrates differences in the preconditioner; the preconditioners offer no benefit for the single node case. The problems have 184962 and 369922 unknowns, respectively.

For both problems, ILU(0) preconditioning offers the best performance with respect to time despite higher levels yielding lower numbers of inner iterations. Interestingly, not updating the preconditioner has no discernible effect on the iteration count and yields lower computational times than when the preconditioner is updated at every iteration. This can be explained by noting that the majority of the Jacobian is relatively insensitive to small changes in k. Given the initial guess (unity in these cases) is expected to be pretty close to the final answer, the original Jacobian should be pretty close to its final value.

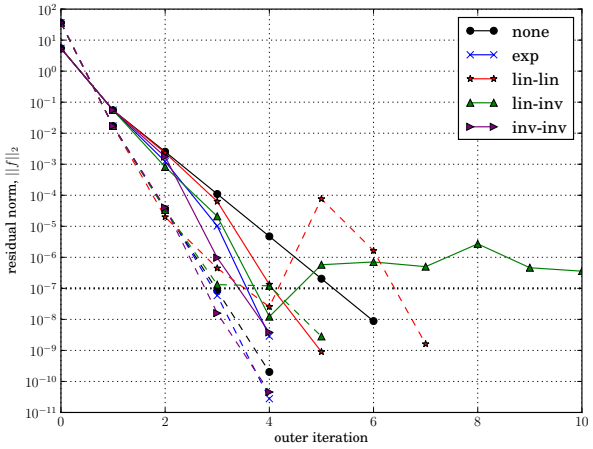

Figure 5: Comparison of Picard acceleration schemes for the 2-D IAEA problem (solid lines) and Koeberg problem (dashed lines).

3.1.7. Comparing Picard, Steffensen, and Newton

To compare the methods, several metrics are of interest. Ultimately, the wall clock time is of most importance. 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 throughout, with order reduction being applied to the 3-D IAEA problem. For the 2-D problems, a 4×4 node-per-assembly model is used, while for the 3-D IAEA problem, a single node is used. This corresponds to 184962 unknowns for the 2-D IAEA and Biblis problems, 369922 for the Koeberg problem, and 988382 for the 3-D IAEA problem.

The solvers tested included 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 lags the k evaluation, using the last iterate for computing Δk. This results in a larger Δk and hence less accurate finite difference. However, the accuracy of the finite difference has no observable effect on the convergence, and for all four problems, the lagged approach has the same number of outer iterations as the regular approach but re-

preconditioner	time [s]	r. time [s][a]	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] Response generation time

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

duces the number of k evaluations by nearly half.

Steffensen's method provides the fastest convergence with respect to outer iterations, but it relies on two k evaluations per outer iteration. Picard with exponential extrapolation yields 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.

3.1.8. Comments

From the diffusion analyses, it appears that Picard iteration using KS for the inners and the exponential extrapolation for accelerating the outers yields the best performance. The Newton methods perform about 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.

3.2. 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 problem is specified in Appendix **??**. Additionally, the transport approximations used are described, and the `Detran` reference results used in the following analysis are compared to the MCNP reference results from the original specification.

3.2.1. Orders and Accuracy

To assess the accuracy of the response schemes available for transport problems, the C5G7 benchmark was solved us-

solver	time [s]	r. time [s][f]	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] Response generation time

Table 3: Outer solver comparison for diffusion problems.

ing both a DLP and Chebyshev-Legendre basis in angle. A DLP spatial basis and a full energy expansion were used throughout. Because the spatial mesh is not exactly uniform, the DLP spatial expansion is not likely to converge as quickly for the C5G7 problem as for the uniformly-meshed diffusion problems (or the small test problems of Chapter **??**). 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}.

Figure 4 shows the convergence of the eigenvalue and pin power errors as a function of space-angle order. By third order, the conservative basis appears to approach a very limited improvement. This is most likely due to the DLP spatial basis used; since the mesh is not uniform, a DLP expansion is not conservative, and low order expansions cannot be expected to yield good results. Since both the DLP and Chebyshev bases yield maximum relative pin power errors of just over 2%, it appears the spatial expansion becomes the dominant source of error. These results are in contrast to those of Chapter **??**, for which a full order spatial basis helped isolate just the error due to expanding in angle. This indicates that a full implementation and systematic study of the modified spatial basis developed in Section **??** would potentially be of great value.

3.2.2. Solver Comparison

The same set of global solvers used in Section 3.1.7 were also applied to the 2-D C5G7 problem. In this case, parallel computation was necessary, and 64 processes were used throughout with a single process for the global balance problem; see Chapter **??** for more details on the parallel implementation of ERMM.

Table 5 provides the wall time, as well as the total and response time summed over all processes. Included also are the number of inner iterations, outer iterations, and k evaluations. Newton's method with the coarse k derivative yields the best performance. Surprisingly, using the fine k derivative requires more outer iterations and hence significantly greater time. Steffensen's method, as for the diffusion problems, requires the fewest outer iterations but at the cost of more k evaluations. Standard Picard iteration performs reasonably well, but the extrapolated Picard iteration fails miserably. However, this is *completely expected*: since the spatial basis is not conservative, λ does not tend toward unity, and hence extrapolation does not apply. This highlights a significant value in selecting a conservative basis, since the extrapolated Picard iteration was the best performing method for the diffusion problems.

3.3. 3-D Takeda

The 3-D Takeda benchmark is an old and rather simple benchmark, but it allows us to examine in more depth the convergence properties of the basis sets. The model consists of three homogeneous regions using a two group approximation. More details can be found in Appendix **??**.

3.3.1. Order Convergence

For the response matrix model, each node is a 5 cm cube discretized with uniform 0.25 cm cells in all dimensions. Using order reduction in both space and angle (see Section 3.1.2), the DLP was used for both ψ and J, and the conservative basis (Chebyshev-Legendre + Jacobi-Legendre) was used for ψ; all bases were used for orders 0 through 4.

Figures 6 and 7 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. It is readily apparent little is gained with increasing angular order when the spatial order is limited to zero. For higher spatial orders, an increasing angular order yields a monotonically decreasing error for both k and the nodal powers. The conservative basis outperforms the DLP variants, yielding nearly sub-1% nodal errors for a third order angular expansion and spatial orders greater than 1. DLP-J yields slightly better nodal powers than DLP-ψ at higher orders, but yields higher k errors for all orders.

For all the bases, a significant trend is that spatial orders above 2 yield diminishing returns; that is, the most consistent improvement over all angular orders and bases is a shift from first to second order in space. This is reasonable because the nodes are homogeneous and boundary quantities should therefore be relatively smooth functions of space.

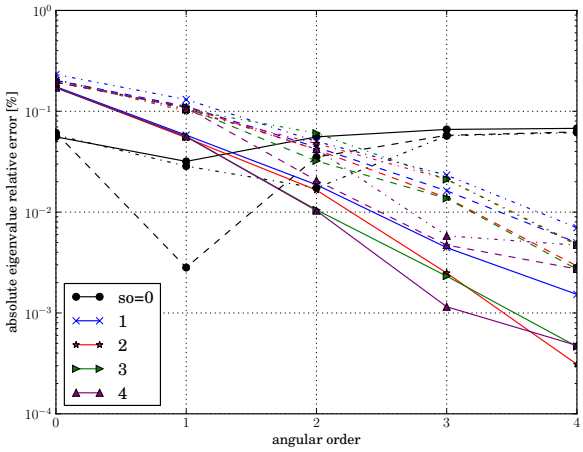

Figure 6: Takeda problem absolute relative eigenvalue error as a function of angular order for several spatial orders. The solid lines indicate the conservative basis, while the dashed and dashed-dot lines indicate the DLP basis used to expand the angular flux ψ and current J, respectively.

3.3.2. Solver Comparison

The same set of solvers as used for the diffusion problems and the 2-D C5G7 problem were applied to a 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 both the spatial and angular terms. The problem was run on 64 processors, with

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

solver	w. time[f][s]	time[g][s]	r. time[h][s]	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
[g] Total time summed over all processes
[h] Total response generation time summed over all processes

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

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.

Similar to the diffusion results, the extrapolated Picard iteration proves 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.

3.4. Summary

Based on the results for both the diffusion and transport problems, Picard iteration with the exponential extrapolation appears generally to be the most efficient of the methods, yielding minimum numbers of k evaluations while providing the lowest global solver overhead. However, the 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.

It is anticipated further work on preconditioners for Newton's method would put the corresponding (global) computation time more in line with that of the extrapolated Picard iteration. Even so, the efficacy and simplicity of the Picard iteration suggests that care should be taken to select a conservative basis leading to $\lambda = 1$ upon convergence.

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

solver	w. time[f][s]	time[g][s]	r. time[h][s]	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
[g] Total time summed over all processes
[h] Total response generation time summed over all processes

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

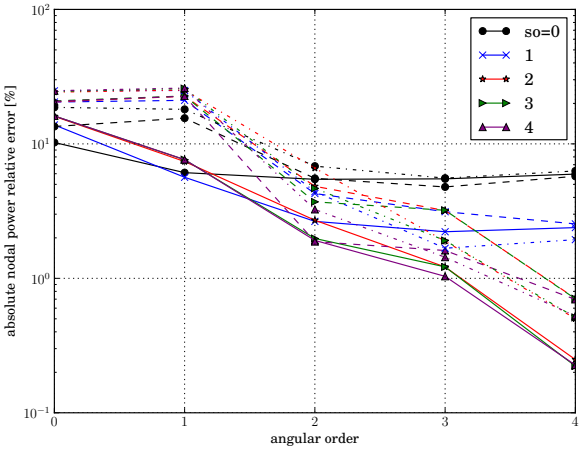

Figure 7: Takeda problem absolute relative nodal power error as a function of angular order for several spatial orders. The solid lines indicate the conservative basis, while the dashed and dashed-dot lines indicate the DLP basis used to expand the angular flux ψ and current J, respectively.

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, A. Tomas, V. Vidal, Arnoldi methods in slepc, Tech. Rep. STR-4, Universidad Politécnica de Valencia, available at http://www.grycap.upv.es/slepc (2006).

[21] D. C. Sorensen, Implicit Application of Polynomial Filters in a k-step Arnoldi Method, SIAM Journal of Matrix Analysis and Applications 13 (1) (1992) 357–385.

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

[23] V. Hernandez, J. E. Roman, A. Tomas, V. Vidal, Krylov-schur methods in slepc, Tech. Rep. STR-7, Universidad Politécnica de Valencia, available at http://www.grycap.upv.es/slepc (2007).

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

[25] 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.

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

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

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

[29] 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.
[30] 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.