Multigroup diffusion preconditioners for multiplying fixed-source transport problems

Jeremy A. Roberts, Benoit Forget

Published Version Information

Citation: Roberts, J. A., & Forget, B. (2014). Multigroup diffusion preconditioners for multiplying fixed-source transport problems. Journal of Computational Physics, 274, 455-472.

Digital Object Identifier (DOI): doi:10.1016/j.jcp.2014.06.034

Publisher's Link: http://www.sciencedirect.com/science/article/pii/S0021999114004410

Multigroup Diffusion Preconditioners for Multiplying Fixed-Source Transport Problems

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

Several preconditioners based on multigroup diffusion are developed for application to multiplying fixed-source transport problems using the discrete ordinates method. By starting from standard, one-group, diffusion synthetic acceleration (DSA), a multigroup diffusion preconditioner is constructed that shares the same fine mesh as the transport problem. As a cheaper but effective alternative, a two-grid, coarse-mesh, multigroup diffusion preconditioner is examined, for which a variety of homogenization schemes are studied to generate the coarse mesh operator. Finally, a transport-corrected diffusion preconditioner based on application of the Newton-Shulz algorithm is developed. The results of several numerical studies indicate the coarse-mesh, diffusion preconditioners work very well. In particular, a coarse-mesh, transport-corrected, diffusion preconditioner reduced the computational time of multigroup GMRES by up to a factor of 17 and outperformed best-case Gauss-Seidel results by over an order of magnitude for all problems studied.

Keywords:
preconditioning, neutron transport, Krylov, discrete ordinates

1. Introduction

As predictive modeling becomes an increasingly important goal to the computational nuclear engineering community, methods that once were too

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

expensive to apply to full reactor models are now being applied with some success on the largest machines available. This includes recent work on both stochastic [1] and deterministic [2] transport methods.

A major focus of research in deterministic transport methods has been the development of highly efficient solvers for the eigenvalue problems and linear systems characteristic of neutron transport in nuclear reactor models. Krylov solvers have been applied with success to a variety of reactor problems, and substantial efforts have improved the performance of these solvers by using application-specific preconditioners [3, 4, 5].

Our goal in this paper is to develop preconditioners for use with Krylov solvers applied to fixed-source (i.e., inhomogeneous) transport problems relevant to reactor analysis. All of our analysis is based on the discrete ordinates (S_N) method, though much of the analysis applies equally to the method of characteristics (MOC). Ultimately, this work supports our simultaneous effort to develop advanced response matrix methods, which aim to solve large reactor eigenvalue problems by spatially-decomposing a domain into independent nodes linked via approximate boundary conditions [6, 7, 8]. The boundary conditions are defined in terms of fixed source problems for each node, and because many such transport problems are required, methods to solve them efficiently are highly desirable.

2. Background

The steady-state, multigroup, neutron transport equation is defined as

$$\begin{aligned} \hat{\Omega}\cdot\nabla\psi_g(\vec{r},\Omega) + \Sigma_{tg}(\vec{r})\psi_g(\vec{r},\Omega) &= \frac{1}{4\pi}\sum_{g'=1}^{G}\Sigma_{sg\leftarrow g'}(\vec{r})\phi_{g'}(\vec{r}) \\ &+ \frac{\chi_g(\vec{r})}{4\pi k}\sum_{g'=1}^{G}\nu\Sigma_{fg'}(\vec{r})\phi_{g'}(\vec{r}) + q_g(\vec{r},\hat{\Omega})\,, \quad g = 1\ldots G\,, \end{aligned} \tag{1}$$

where

$$\phi_g(\vec{r}) = \int_{4\pi} d\Omega\psi_g(\vec{r},\Omega)\,, \tag{2}$$

and the notation is standard [9]. Isotropic scattering has been assumed for clarity of presentation, though this is not a general requirement. Fission is explicitly represented in Eq. (1), which leads to *multiplying* fixed-source problems. The parameter k is an eigenvalue for the homogeneous form of

Eq. (1) (i.e., for $q_g = 0$), but in this paper k is used as a variable parameter. For clarity in the development to follow, the one speed or "within-group" transport equation is defined as

$$\hat{\Omega} \cdot \nabla \psi(\vec{r}, \Omega) + \Sigma_t(\vec{r}) \psi(\vec{r}, \Omega) = \frac{1}{4\pi} \phi(\vec{r}) \Sigma_s(\vec{r}) + q(\vec{r}, \hat{\Omega}) \,, \tag{3}$$

where the group g is suppressed.

2.1. Operator Notation

In this paper, the S_N method is used throughout to discretize Eq. (1) in angle with the diamond-difference approximation in space [10]. By following the presentation of Larsen and Morel [11], the scalar flux is defined as

$$\phi(\vec{r}) = \int_{4\pi} d\Omega \, \psi(\vec{r}, \Omega) \approx \sum_{n=1}^{N} w_n \psi_n(\vec{r}) \,. \tag{4}$$

Let a *discrete-to-moment* operator $\mathbf{D}$ satisfy

$$\phi = \mathbf{D}\psi \,, \tag{5}$$

where space and angle indices are implicit. In addition, let a *moment-to-discrete* operator $\mathbf{M}$ satisfy

$$\psi = \mathbf{M}\phi \,, \tag{6}$$

where generally $\mathbf{D} \neq \mathbf{M}^{-1}$. Finally, by defining the operator

$$\mathbf{L}(\cdot) \equiv \left(\hat{\Omega} \cdot \nabla + \Sigma_t(\vec{r}) \right) (\cdot) \,, \tag{7}$$

the one-group, discretized transport equation becomes

$$\mathbf{L}\psi = \mathbf{M}\mathbf{S}\phi + q \,, \tag{8}$$

where $\mathbf{S}$ is the scattering operator. For the case of isotropic scattering in one group, $\mathbf{M} = \frac{1}{4\pi}$ while $\mathbf{S} = \Sigma_s(\vec{r})$.

For the multigroup problem, this notation generalizes to[4]

$$\mathbf{L}_g \psi_g = \mathbf{M} \sum_{g'=1}^{G} \left(\mathbf{S}_{gg'} + \frac{1}{k} \mathbf{X}_g \mathbf{F}_{g'} \right) \phi_{g'} + q_g \,, \tag{9}$$

where fission is explicitly represented, and $\mathbf{X}$ represents the fission spectrum χ in operator form.

Equations (8) and (9) are defined explicitly in terms of the unknown angular flux ψ. Frequently, and particularly for reactor physics, reaction rates are of most interest, for which only the scalar flux is needed. Consequently, the angular flux is rarely stored explicitly in practice, but rather is computed on-the-fly during a sweep through the space-angle grid. This sweeping operation can be represented explicitly by casting the transport equations in terms of only the scalar flux (and higher-order flux moments if scattering is anisotropic).

To illustrate, consider the multiplication of Eq. (8) by the space-angle *transport sweep* operator $\mathbf{T} = \mathbf{D}\mathbf{L}^{-1}$, which leads to

$$\mathbf{D}\psi = \mathbf{D}\mathbf{L}^{-1}\mathbf{M}\mathbf{S}\phi + \mathbf{D}\mathbf{L}^{-1}q \tag{10}$$

The substitution of $\phi = \mathbf{D}\psi$ into Eq. (10) yields

$$(\mathbf{I} - \mathbf{T}\mathbf{M}\mathbf{S})\phi = \mathbf{T}q\,, \tag{11}$$

or

$$\mathbf{A}_{\mathrm{WG}}\phi = b\,, \tag{12}$$

where

$$\mathbf{A}_{\mathrm{WG}} \equiv \mathbf{I} - \mathbf{T}\mathbf{M}\mathbf{S} \tag{13}$$

is the *within-group transport operator* and

$$b = \mathbf{T}q \tag{14}$$

represents the *uncollided* flux moments. Equation (11) is a linear equation in standard form for the scalar flux moments ϕ.

The operator $\mathbf{D}\mathbf{L}^{-1}$ has a very specific physical interpretation. The inverse $\mathbf{L}^{-1}$ represents the set of space-angle sweeps from one global boundary to another. The additional factor $\mathbf{D}$ implies that along the space-angle sweep, ϕ is updated, which eliminates the need to store ψ. This process is exactly the way in which $\mathrm{S_N}$ solvers have traditionally been implemented.

The extension of Eq. (11) to the multigroup transport equation leads to the similar form

$$\left(\mathbf{I} - \mathbf{T}_{\mathrm{MG}}\mathbf{M}_{\mathrm{MG}}\left(\mathbf{S}_{\mathrm{MG}} + \frac{1}{k}\mathbf{X}_{\mathrm{MG}}\mathbf{F}_{\mathrm{MG}}^{\mathsf{T}}\right)\right)\phi = \mathbf{T}_{\mathrm{MG}}q_{\mathrm{MG}}\,, \tag{15}$$

where $\mathbf{T}_{\mathrm{MG}}$ and $\mathbf{M}_{\mathrm{MG}}$ are block diagonal operators having blocks $\mathbf{T}_g$ and $\mathbf{M}$, respectively, $\mathbf{X}_{\mathrm{MG}}$, $\mathbf{F}_{\mathrm{MG}}$, and q_{MG} are the column vectors having elements $\mathbf{X}_g$, $\mathbf{F}_g$, and q_g respectively, and the multigroup scattering operator is defined

$$\mathbf{S}_{\mathrm{MG}} = \begin{pmatrix} \mathbf{S}_{11} & \cdots & \mathbf{S}_{1G} \\ \vdots & \ddots & \vdots \\ \mathbf{S}_{G1} & \cdots & \mathbf{S}_{GG} \end{pmatrix} , \tag{16}$$

where the blocks $\mathbf{S}_{gg'}$ are the group-to-group scattering operators of Eq. (9). Similar to the within-group problem, the *multigroup transport operator* is defined as

$$\mathbf{A}_{\mathrm{MG}} = \mathbf{I} - \mathbf{T}_{\mathrm{MG}}\mathbf{M}_{\mathrm{MG}} \left(\mathbf{S}_{\mathrm{MG}} + \frac{1}{k}\mathbf{X}_{\mathrm{MG}}\mathbf{F}_{\mathrm{MG}}^{\mathsf{T}} \right) . \tag{17}$$

So far, boundary conditions have been neglected. Some boundary conditions, including reflecting, periodic, and white conditions, lead to an additional set of unknowns, namely the incident (or exiting) boundary fluxes. In theory, the treatment of these conditions is straightforward, but we refrain from further discussion for two reasons. First, and most importantly, response function calculations, the motivating application, use vacuum conditions only, so a treatment of reflecting boundaries is beyond the present scope. Second, the addition of boundary unknowns complicates somewhat the presentation of the algorithms to be discussed below. However, a detailed treatment of boundary conditions with Krylov solvers can be found in the work of Warsa *et al.* [3].

2.2. Classical Transport Solvers

Consider the fixed source multigroup transport problem represented by

$$\mathbf{A}_{\mathrm{MG}}\phi = b\,. \tag{18}$$

The traditional method for solving the multigroup equations is a nested iteration in which a series of within-group equations is solved for each group, and the scattering (and possibly) fission sources are updated following each within-group calculation. More modern treatments view the multigroup equations as a complete set to be solved simultaneously.

The Gauss-Seidel method has long been used to solve the multigroup equations. For standard problems, the method first solves the fast within-group equation. The updated fast-group flux can then be used to define

the in-scatter source for the next group, and so on. Because this algorithm implies the inversion of each group *block* (as defined in Eq. (15)), the method is more accurately described as *block* Gauss-Seidel in energy.

For problems with no upscatter and no fission, the Gauss-Seidel method is an essentially exact scheme, assuming the within-group equations are solved exactly. However, for cases with upscatter or fission, extra "upscatter" iterations are required, and sometimes the convergence of Gauss-Seidel becomes prohibitively slow.

The standard method for solving the within-group transport equation has been *source iteration*. The basic idea is a simple one: given an external source (including in-scatter and fission), the flux is guessed, the within-group scattering source is computed, the new flux is computed, and the process is repeated until converged. Mathematically, source iteration is defined by the process

$$\phi^{(n)} = \mathbf{TMS}\phi^{(n-1)} + \mathbf{T}q\,. \tag{19}$$

However, recall that Richardson iteration for the system $\mathbf{A}x = b$ is defined by the process

$$x^{(n)} = (\mathbf{I} - \mathbf{A})x^{(n-1)} + b\,. \tag{20}$$

Because $\mathbf{I} - \mathbf{A}_{\mathrm{WG}} = \mathbf{TMS}$, source iteration is equivalent to Richardson iteration[12].

2.3. Krylov Solvers

Because of the limitations of the Gauss-Seidel and source iteration schemes, much work has been done to apply modern linear solvers to transport problems. One of the most successful class of solvers studied consists of Krylov subspace methods.

Linear (and eigenvalue) solvers can be classified as *stationary* or *nonstationary* methods. Stationary methods produce updated solutions using only a single previous solution, and the Gauss-Seidel and Richardson methods are both stationary. Other well-known examples of stationary methods are the Jacobi and successive over-relaxation (SOR) methods. On the other hand, nonstationary methods produce a solution based on two or more previous iterates (or the information used to create those iterates).

Krylov methods are nonstationary methods that rely on the construction of a *Krylov subspace* of dimension m, defined for an $n \times n$ operator $\mathbf{A}$ as

$$\mathcal{K}_m(\mathbf{A}, x_0) \equiv \mathrm{span}\{x_0,\, \mathbf{A}x_0,\, \mathbf{A}^2x_0,\, ,\ldots,\, ,\mathbf{A}^{m-1}x_0\}\,, \tag{21}$$

for some initial, possibly random, vector x_0 [13]. For brevity, $\mathcal{K}_m(\mathbf{A}, x_0)$ is abbreviated as $\mathcal{K}_m$. The main idea of Krylov subspace methods is to find $x \in \mathcal{K}_m$ that "best" solves the system of interest, be it an eigenproblem or linear system, usually for $m \ll n$.

Although the vectors $[x_0, \mathbf{A}x_0, \ldots \mathbf{A}^{m-1}x_0]$ form a basis of $\mathcal{K}_m$, this basis is ill-conditioned because the repeated application of $\mathbf{A}$ sends x_0 into the direction of the dominant eigenvector of $\mathbf{A}$. For numerical computation, a more suitable approach is to use an orthogonal basis of $\mathcal{K}_m$. For nonsymmetric operators, an orthogonal basis can be generated by Arnoldi's method, which uses successive application of the modified Gram-Schmidt (or an equivalent) process to construct the Arnoldi decomposition

$$\mathbf{AV} = \mathbf{VH} + fe_m^{\mathsf{T}}\,, \tag{22}$$

where $\mathbf{V} \in \mathbb{R}^{n\times m}$ consists of orthonormal columns, $\mathbf{H} \in \mathbb{R}^{m\times m}$ is an upper Hessenberg matrix, e_m is the m-vector of all zeros except a one in the mth location, and f is the residual, which is orthogonal to the columns of $\mathbf{V}$.

A popular Krylov method for nonsymmetric linear systems (such as the transport equations above) is GMRES [14]. The basic idea of GMRES is straightforward: the nth step of GMRES produces an $m \times m$ Hessenberg matrix and the corresponding basis $\mathbf{V}$, and the approximate solution x_m is found by minimizing the residual norm $||r||_2 = ||\mathbf{A}x_m - b||_2$ for $x_m \in \mathcal{K}_m$. In other words, $x_m = \mathbf{V}y$, where y satisfies

$$\begin{aligned} ||\mathbf{AV}y - b||_2 &= ||\mathbf{V}^{\mathsf{T}}\mathbf{AV}x_n - \mathbf{V}^{\mathsf{T}}b|| \\ &= ||\mathbf{H}y - \mathbf{V}^{\mathsf{T}}b||\,. \end{aligned} \tag{23}$$

Equation (23) shows that GMRES finds the best solution $x_m \in \mathcal{K}_m$ in a least-squares sense.

Krylov solvers have been used to solve both the within-group [3] and multigroup [2] transport equations. For the multigroup equations in particular, the independent nature of the group-wise blocks makes parallelization in energy much more straightforward than for Gauss-Seidel. For problems in which there is no fission and upscatter is limited to a subset of thermal groups, it is possible to solve the downscatter groups via Gauss-Seidel and to use a Krylov method on the thermal block of Eq. (15), which leads to improved efficiency for some problems [2]. However, when fission is included, there is always thermal-to-fast coupling, and solving the full system via a Krylov method is to be preferred.

3. Diffusion-Based Preconditioners

Krylov methods are often more robust than classical stationary methods, but their performance can be improved significantly by *preconditioning*. A preconditioner $\mathbf{P}$ is an operator whose inverse satisfies $\mathbf{P}^{-1} \approx \mathbf{A}^{-1}$ in some sense and is relatively inexpensive to apply. A *left-preconditioned* linear system is

$$\mathbf{P}^{-1}\mathbf{A}x = \mathbf{P}^{-1}b\,, \tag{24}$$

while a *right-preconditioned* system is

$$\mathbf{A}\mathbf{P}^{-1}y = b \tag{25}$$

with $x = \mathbf{P}^{-1}y$. The left-preconditioned residual differs from the original residual but may be a better measure of the error $x^* - x$, where x^* is the solution, while the right-preconditioned system preserves the original residual. In practice, the difference between the convergence behaviors of left- and right-preconditioned GMRES is typically negligible unless the preconditioner is ill-conditioned[13].

A preconditioner typically leads to a clustering of eigenvalues. As an extreme example, suppose that $\mathbf{P} = \mathbf{A}$. The preconditioned operator is then $\mathbf{A}\mathbf{P}^{-1} = \mathbf{I}$, for which all the eigenvalues are unity. Of course, to apply $\mathbf{P}^{-1}$ in this case represents solving the original problem. Although preconditioners cannot in practice be expected to yield a set of eigenvalues equal to unity, any clustering typically improves convergence, and often just pushing eigenvalues away from the origin tends to improve convergence [11].

3.1. Diffusion Synthetic Acceleration

Frequently, the most successful preconditioners are based on *a priori* knowledge of the physics or structure of the problem. This has long been the case for accelerating transport problems, for which a low-order approximation, often based on diffusion, provides an efficient update or correction to an unconverged transport solution. Here, a brief summary is given of *diffusion synthetic acceleration* (DSA), a diffusion-based preconditioner that has long been used to improve source iteration [10] and was more recently applied to Krylov solvers for the within-group equations [3]. The development closely follows the excellent treatment of Larsen and Morel [11].

Consider the nth flux iterate ϕ^n from source iteration. Then, one additional iteration leads to

$$\phi^{(n+\frac{1}{2})} = \mathbf{TMS}\phi^{(n)} + \mathbf{T}q\,, \tag{26}$$

where $\mathbf{S}$ is assumed to contain both the scatter and fission within-group terms, and the new iterate is assigned a half index. The subtraction of Eq. (26) from Eq. (11) leads to

$$(\mathbf{I} - \mathbf{TMS})\phi - \phi^{(n+\frac{1}{2})} = -\mathbf{TMS}\phi^{(n)}\,, \tag{27}$$

where ϕ is the solution. By adding $\mathbf{TMS}\phi^{(n+\frac{1}{2})}$ to both sides of Eq. (27) and rearranging the result, we have

$$\epsilon = \overbrace{(\mathbf{I} - \mathbf{TMS})^{-1}\mathbf{TM}}^{\text{what is approximated}}\overbrace{\mathbf{S}(\phi^{(n+\frac{1}{2})} - \phi^{(n)})}^{z}\,, \tag{28}$$

where the error $\epsilon = \phi - \phi^{(n+\frac{1}{2})}$ satisfies the transport equation

$$(\hat{\Omega}\cdot\nabla + \Sigma_t)\varepsilon - \frac{\Sigma_s}{4\pi}\epsilon = \frac{z}{4\pi}\,, \tag{29}$$

and

$$\epsilon = \int_{4\pi} d\Omega\, \varepsilon\,. \tag{30}$$

The error equation is just as expensive to solve as the original transport equation. As an alternative, the diffusion approximation is used to define the approximate error equation

$$(-\nabla\cdot D\nabla + \Sigma_t - \Sigma_s)\epsilon = z\,, \tag{31}$$

or, in operator form,

$$\epsilon = \mathbf{C}_{\text{WG}}^{-1}z = \mathbf{C}_{\text{WG}}^{-1}\mathbf{S}(\phi^{(n+\frac{1}{2})} - \phi^{(n)})\,. \tag{32}$$

This leads to preconditioned source iteration,

$$\begin{aligned}
\phi \approx \phi^{(n+1)} &= \phi^{(n+\frac{1}{2})} + \mathbf{C}_{\text{WG}}^{-1}\mathbf{S}(\phi^{(n+\frac{1}{2})} - \phi^{(n)}) \\
&= (\mathbf{I} + \mathbf{C}_{\text{WG}}^{-1}\mathbf{S})\phi^{(n+\frac{1}{2})} - \mathbf{C}_{\text{WG}}^{-1}\mathbf{S}\phi^{(n)} \\
&= \Big(\mathbf{I} - (\mathbf{I} + \mathbf{C}_{\text{WG}}^{-1}\mathbf{S})(\mathbf{I} - \mathbf{TMS})\Big)\phi^{(n)} + (\mathbf{I} + \mathbf{C}_{\text{WG}}^{-1}\mathbf{S})\mathbf{T}q \\
&= (\mathbf{I} - \mathbf{P}_{\text{WG-DSA}}^{-1}\mathbf{A}_{\text{WG}})\phi^{(n)} + \mathbf{P}_{\text{WG-DSA}}^{-1}\mathbf{T}q\,,
\end{aligned} \tag{33}$$

where

$$\mathbf{P}_{\text{WG-DSA}}^{-1} = (\mathbf{I} + \mathbf{C}_{\text{WG}}^{-1}\mathbf{S}) \tag{34}$$

is the within-group diffusion preconditioning process. Although source iteration provides an intuitive way to define $\mathbf{P}_{\text{WG-DSA}}$, the effectiveness of DSA highly depends on discretization of the transport and diffusion operators. When applied to Krylov solvers, DSA is far less sensitive to discretization and more effective for realistic problems [3].

It is natural to extend the diffusion preconditioner to multigroup problems, which leads to the process

$$\mathbf{P}_{\text{MG-DSA}}^{-1} \equiv \mathbf{I} + \mathbf{C}_{\text{MG}}^{-1} \left(\mathbf{S}_{\text{MG}} + \mathbf{X}_{\text{MG}} \mathbf{F}_{\text{MG}}^{\mathsf{T}} \right) , \tag{35}$$

where $\mathbf{C}_{\text{MG}}$ is the multigroup diffusion operator defined block-wise as

$$\mathbf{C}_{\text{MG},gg'} \equiv \delta_{gg'} \left(-\nabla \cdot D_g(\vec{r}) \nabla + \Sigma_{tg}(\vec{r}) \right) - \Sigma_{sgg'}(\vec{r}) - \chi_g \nu \Sigma_{fg'}(\vec{r}) \, . \tag{36}$$

An initial review of the literature yielded no application of multigroup diffusion as a preconditioner for multigroup transport problems. Related work addressed the acceleration of outer Gauss-Seidel upscatter iterations [12] and fission iterations based on the rank one fission operator ($\mathbf{X}\mathbf{F}^{\mathsf{T}}$) [15], but each method used an equivalent one-group formulation. However, the full multigroup diffusion problem is itself expensive for large problems, which may explain why it has not been used extensively for acceleration.

3.2. Coarse-Mesh Diffusion Preconditioning

The preconditioning of the multigroup equations with diffusion can lead to very large diffusion operators, and for many problems, the computational cost of constructing and applying the preconditioner is prohibitive. As an alternative, the use of coarse-mesh diffusion preconditioners is proposed. Coarse-mesh diffusion operators have long been central to acceleration techniques in reactor analysis, a chief example being the nonlinear diffusion acceleration (NDA) scheme developed by Smith [16] for nodal diffusion methods and later extended to transport methods [17].

Although no results were found in the literature describing the use of coarse-mesh diffusion as a preconditioner, more general coarse-mesh schemes in space, angle, and energy have been of substantial recent interest, particularly for multigrid preconditioning. Multigrid methods, like DSA (itself a two-grid method in angle), use a coarse-grid solution to improve a fine-grid solution by way of restriction (essentially averaging) and prolongation (essentially interpolation) operations. The idea is that slowly varying error modes, which tend to converge slowly for a stationary solver like Jacobi iteration,

become highly oscillatory, quickly converging modes on the coarse mesh. A complete description of multigrid methods is outside the present scope, but the following sections describe implementation of a two-grid diffusion preconditioner. For a more complete overview of multigrid methods, the reader would be best served by one of the standard reviews available, e.g., Ref. [18].

3.2.1. A Spatial Two-Grid Multigroup Diffusion Preconditioner

In this work, a two-grid spatial scheme is applied to the diffusion preconditioner. Recent work suggests that multigrid methods in the energy variable can work very well [5, 4]. Nonlinear diffusion acceleration methods also typically uses a coarse energy mesh to great effect [17]. However, our initial studies using a coarsened energy variable within a coarse-mesh diffusion preconditioner suggest that the simultaneous restriction of space, angle, and energy may have inherent difficulties.

The two-grid, coarse-mesh diffusion preconditioner is a natural extension to Eq. (35) and is defined as

$$\mathbf{P}^{-1}_{\text{MG-CMDSA}} \equiv \mathbf{I} + \boldsymbol{P}\mathbf{C}_H^{-1}\boldsymbol{R}\left(\mathbf{S}_{\text{MG}} + \mathbf{X}_{\text{MG}}\mathbf{F}^{\mathsf{T}}_{\text{MG}}\right) , \tag{37}$$

where $\boldsymbol{P}$ (not to be confused with the preconditioner) and $\boldsymbol{R}$ are the *prolongation* and *restriction* operators, respectively, and $\mathbf{C}_H$ is the multigroup diffusion operator defined on the coarse spatial mesh.

3.2.2. Coarse-Mesh Operator Homogenization

To define $\mathbf{C}_H$, cross sections must be *homogenized* over the fine mesh. The most physically-sound approach for producing averaged cross sections is to use flux-weighting in a way that preserves reaction rates. Although a variety of such homogenization techniques exist, a rather conventional scheme is used that is simple to apply in preconditioning. For the particular case of group constant generation via an assembly-level lattice physics solver, the results of the scheme are called *assembly homogenized cross sections* [19].

Suppose the total cross section Σ_t is to be averaged over coarse-mesh cell j. Let the fine-mesh flux in cell $i \in j$ be denoted ϕ_i. Then the average total cross section in the jth coarse-mesh cell is defined as

$$\Sigma_{t,j} = \frac{\sum_{i\in j} V_i \phi_i \Sigma_{t,i}}{\sum_{i\in j} V_i \phi_i} , \tag{38}$$

where V_i is the fine-mesh cell volume. If the coarse-mesh flux is defined as the volume average, i.e.,

$$\phi_j = \frac{\sum_{i \in j} V_i \phi_i}{V_j} \,, \tag{39}$$

where

$$V_j = \sum_{i \in j} V_i \,, \tag{40}$$

then the total interaction rate in coarse-mesh cell j is defined as

$$\sum_{i \in j} V_i \Sigma_{t,i} \phi_i = \Sigma_j \sum_{i \in j} V_i \phi_i = V_j \phi_j \Sigma_{t,j} \,. \tag{41}$$

Hence, the averaged quantities preserve the integrated reaction rate associated with the given fine-mesh flux. All group constants, including the diffusion coefficient D, can be generated in this way.

The obvious problem with this approach is that the fine-mesh flux ϕ_i is not known. The simplest approximation is to assume a constant flux, which leads to volume-weighted cross sections. For preconditioning, this is a suitable approximation because the conservation of reaction rates is not a prerequisite. Unlike certain nonlinear acceleration techniques (e.g., NDA) that rely on conservation to provide an integral form of the *solution* at each step, preconditioning—a linear process—only seeks to provide an additive improvement.

However, that flexibility certainly does *not* preclude the use of more accurate flux shapes to achieve better results. In typical lattice physics applications, group constants are found by solving the transport equation in a pincell or assembly subject to reflecting conditions, and using the resulting spectrum for weighting. For preconditioning, a simple scheme could be used in which pincells approximating parts of the full problem are solved, and the resulting fluxes are used to produce homogenized materials in the appropriate cells. The effect of more realistic shape functions on the preconditioner efficiency was examined, and the numerical results to follow in Section 4.2 show that little improvement is gained by better shapes.

One might recognize that the current flux iterate is freely available for use in homogenization; however, because the flux and homogenization process would change from step to step, the entire process would become nonlinear, and its use with standard Krylov linear solvers would be suspect. Consequently, a study of this approach in solvers allowing variable preconditioners (e.g., FGMRES [20]) would be valuable future research.

3.2.3. Restriction

To restrict the fine-mesh input vector for application of the inverse coarse-mesh diffusion operator, a simple spatial average is used. As an example, consider a 1-D problem discretized into 6 fine meshes. The coarse meshing process is based on a level parameter l that defines the number of fine meshes per coarse mesh. Suppose $l = 2$, leading to the fine-to-coarse mapping $[0, 0, 1, 1, 2, 2]$. The restriction operator is defined

$$\boldsymbol{R} = \begin{pmatrix} v_0 & v_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & v_2 & v_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & v_4 & v_5 \end{pmatrix} , \tag{42}$$

where

$$v_i = \frac{V_i}{V_j} , \qquad \text{for fine mesh } i \text{ in coarse mesh } j , \tag{43}$$

and, hence, $\sum_{i \in j} v_i = 1$.

3.2.4. Prolongation

To prolong the coarse-mesh result back to the fine mesh, the coarse-mesh value is distributed on the fine grid based on the (approximate) flux $\tilde{\phi}$ used to produce the coarse-mesh diffusion operator. Given a coarse-mesh value ϕ_j, the prolonged flux is

$$\phi_{i \in j} = \phi_j \frac{l\tilde{\phi}_i}{\sum_{i \in j} \tilde{\phi}_i} . \tag{44}$$

For the example above, suppose each coarse region is assumed to have a fine-mesh flux shape of $\tilde{\phi} = [a, b]$, where $a + b = 1$. The prolongation operator is then defined

$$\boldsymbol{P} = \begin{pmatrix} 2a & 0 & 0 \\ 2b & 0 & 0 \\ 0 & 2a & 0 \\ 0 & 2b & 0 \\ 0 & 0 & 2a \\ 0 & 0 & 2b \end{pmatrix} . \tag{45}$$

In the constant-flux approximation (i.e., $a = b = 0.5$), multiplication of $\boldsymbol{R} \in \mathbb{R}^{3\times6}$ by $\boldsymbol{P} \in \mathbb{R}^{6\times3}$ yields the identity matrix $\mathbf{I} \in \mathbb{R}^{3\times3}$.

3.2.5. Smoothing

Although the coarse-mesh scheme described so far represents a complete preconditioner, it can be significantly improved by a *smoothing* operation. The motivation for smoothing is that the coarse-mesh solve damps low-frequency error modes of the fine-mesh problem but not high-frequency modes. In more physical terms, a coarse-mesh solve can be expected to get the gross shape right but not the finer details. A smoothing operator uses a few iterations of a classical scheme like Richardson, Jacobi, or Gauss-Seidel with the fine-mesh operator. In many multigrid algorithms, a smoother is also often used before the coarse-mesh solve to ensure the error is "smooth" before restriction; however, for the problems studied, pre-smoothing did not lead to improved performance.

For the coarse-mesh, diffusion preconditioner, smoothing requires the action of the fine-mesh diffusion matrix. Although the cost of producing this matrix may be somewhat large, that cost is much smaller than inverting the operator in a fine-mesh preconditioner (and much, much smaller than the application of the transport operator). The smoothing process used is the weighted Jacobi method, which for the generic problem $\mathbf{A}x = b$ is defined by the process

$$x^{(n+1)} = -\omega \mathbf{A}_D^{-1}(\mathbf{A}_L + \mathbf{A}_U)x^{(n)} + (1-\omega)\mathbf{A}_D^{-1}b\,, \tag{46}$$

where $\mathbf{A}_D$ represents the diagonal of $\mathbf{A}$, $\mathbf{A}_L$ and $\mathbf{A}_U$ represent the strictly lower and strictly upper triangular parts of $\mathbf{A}$, and ω is the weighting parameter. In practice, the selection of an appropriate value for ω is dependent on both the problem and smoothing process[13], but for the problems studied in this paper, suitable values for ω were always less than one.

A similar two-grid preconditioner with smoothing was developed for application to the one-group P_1 equations [21]. In that method, a consistently-discretized, fine-mesh diffusion operator was inverted, and the fine-mesh transport operator was used with a weighted-Richardson smoother. For the P_1 equations, the application of either the transport operator or the associated diffusion operator should result in similar computational costs, but for many-angle S_N calculations, the application of the fine-mesh diffusion operator is significantly less costly than the application of the transport operator.

3.3. Transport-Corrected Diffusion Preconditioners

A final technique developed is to form a preconditioner that contains information from the transport operator rather than relying solely on the

diffusion approximation. In theory, the explicit formation of the transport operator defined by Eq. (17) is possible. With an explicit operator, various approximate factorizations become natural preconditioner options. However, the construction of such operators requires the use of matrices with sizes proportional to the number of angles, at least implicitly. This can quickly lead to huge memory requirements, a complicated construction process, or both.

The construction of explicit transport operators has been studied for the discrete ordinates method [22] and for the method of characteristics [23]. The former work treated the angular flux directly, and the analysis was limited to relatively small, multigroup problems in one dimension. The latter work noted that the cost of constructing the transport operator was exceedingly large and developed a marginally successful parallel scheme limited to the within-group equations.

In the present work, the transport operator is always a "matrix free" operator, which means that the action $y \leftarrow \mathbf{A}x$ is performed by functions and not explicit matrix operations. Hence, the action of $\mathbf{A}$ is available, but $\mathbf{A}$ itself is not, so preconditioners based on approximate factorizations are not directly applicable. Ultimately, any of the preconditioners studied that use a transport operator are limited to actions of the operator only.

To develop a matrix-free, transport-based preconditioner, suppose that a matrix inverse $\mathbf{P}_0^{-1}$ is available that represents an initial approximation of $\mathbf{A}^{-1} \in \mathbb{R}^{n \times n}$. For application to transport problems, $\mathbf{P}_0$ shall be the diffusion preconditioner $\mathbf{P}_{\text{MG-DSA}}$ or a coarse-mesh variant. The present goal is to improve this preconditioning process by using the (possibly approximate) action of $\mathbf{A}^{-1}$.

One way to compute an improved (approximate) matrix inverse is to apply a method attributed by various sources to Shulz or Hotelling and Bodewig [24, 25], which, in fact, represents the use of Newton's method to find a matrix inverse. For brevity, the method is called the Newton-Shulz method. A rigorous analysis of the method can be found elsewhere [25], but we motivate it by considering the simple problem $ax = 1$ proposed in Ref. [26]. By viewing this simple equality as the nonlinear problem $f(x) = 1/x - a = 0$, for which the Jacobian is defined as $f'(x) = -1/x^2$, Newton's method leads

to the process

$$\begin{aligned} x^{(n+1)} &= x^{(n)} + \frac{1}{[x^{(n)}]^{-2}} \left(\frac{1}{x^{(n)}} - a \right) \\ &= 2x^{(n)} - x^{(n)} a x^{(n)} \,. \end{aligned} \tag{47}$$

For the more general case of matrices, this suggests the process

$$\mathbf{X}^{(n+1)} = \mathbf{X}^{(n)}(2\mathbf{I} - \mathbf{A}\mathbf{X}^{(n)}) \,, \tag{48}$$

where $\mathbf{X}^{(n)} \approx \mathbf{A}^{-1}$ and $\mathbf{X}^{(n+1)}$ is an improved approximation.

As is typically the case for Newton's method, the convergence to the desired solution requires an initial guess that is sufficiently close to that solution. By using $\mathbf{X} = \mathbf{P}_{\text{MG-DSA}}$ or a coarse-mesh variant as the initial guess, convergence has always been achieved in our studies. Hence, a simple, one-step, transport-corrected diffusion preconditioning process is defined by

$$\mathbf{P}^{-1}_{\text{TC-MG-DSA-1}} = \mathbf{P}^{-1}_{\text{MG-DSA}}(2\mathbf{I} - \mathbf{A}_{\text{MG}}\mathbf{P}^{-1}_{\text{MG-DSA}}) \,. \tag{49}$$

Although Eq. (49) represents an improved preconditioner, the application of $\mathbf{A}_{\text{MG}}$ requires an additional space-angle-energy sweep, the number of which we are ultimately trying to minimize. As an alternative, we can substitute

$$\mathbf{A}_{\text{MG}} \approx \tilde{\mathbf{A}}_{\text{MG}} \tag{50}$$

into Eq. (49) in place of $\mathbf{A}_{\text{MG}}$, where the tilde indicates some approximation, e.g., a coarser angular quadrature. While this appears at first to be a multigrid method, the (approximate) transport operator is not inverted, but rather it is used to improve the diffusion preconditioning process. In the terminology of Newton methods, the use of the approximate operator leads to an approximate Jacobian.

Because only a few iterations of Newton's method are needed to converge if an appropriate initial guess is available, the use of the approximate operator $\tilde{\mathbf{A}}_{\text{MG}}$ will lead to *its* approximate inverse and *not* that of $\mathbf{A}_{\text{MG}}$. However, $\tilde{\mathbf{A}}^{-1}_{\text{MG}}$ should be even closer to $\mathbf{A}^{-1}_{\text{MG}}$ than the original diffusion preconditioning process, so the application of the full transport operator can be expected to yield a much more valuable improvement if used in the final Newton-Shulz iteration.

Algorithm 1 provides the complete preconditioning process for application to an input vector based on the recursive application of the Newton-Schulz

method defined in Algorithm 2. The true operator is applied once in the last step, but the number of actions of $\tilde{\mathbf{A}}_{\text{MG}}$ grows exponentially with the number of iterations: one for $k = 2$, three for $k = 3$, seven for $k = 4$, and so on. Hence, the cost of the preconditioner is likely to become excessive for more than two or three iterations unless the approximate operator is significantly less expensive than the true operator. We reiterate that any of the multigroup diffusion preconditioners can be used with Algorithm 1, and due to the expense incurred for fine-mesh diffusion operations, the coarse-mesh DSA variants are most likely to outperform fine-mesh DSA for this purpose.

Data: transport operator $\mathbf{A}_{\text{MG}}$, approximate transport operator $\tilde{\mathbf{A}}_{\text{MG}}$, diffusion preconditioner $\mathbf{P}_{\text{MG-DSA}}$, number of corrections k, input vector x
Result: output vector y
$y \leftarrow \text{Newton-Shulz}(\tilde{\mathbf{A}}_{\text{MG}}, \mathbf{P}_{\text{MG-DSA}}, k-1, x)$
$y \leftarrow 2x - \mathbf{A}_{\text{MG}} y$
$y \leftarrow \text{Newton-Shulz}(\tilde{\mathbf{A}}_{\text{MG}}, \mathbf{P}_{\text{MG-DSA}}, k-1, y)$

Algorithm 1: Transport-Corrected Diffusion Preconditioner

Data: operator $\mathbf{A}$, initial preconditioner $\mathbf{P}_0$, number of corrections k, input vector x
Result: output vector y
if $k > 1$ **then**
  // Apply earlier steps recursively
  $y \leftarrow \text{Newton-Shulz}(\mathbf{A}, \mathbf{P}_0, k-1, x)$
  $y \leftarrow 2x - \mathbf{A} y$
  $y \leftarrow \text{Newton-Shulz}(\mathbf{A}, \mathbf{P}_0, k-1, y)$
else
  // Apply the initial preconditioner
  $y \leftarrow \mathbf{P}_0^{-1} x$
end

Algorithm 2: Newton-Shulz for an arbitrary operator $\mathbf{A}$ and initial preconditioner $\mathbf{P}_0$

4. Numerical Results

In this section, the theoretical development of Sections 2 and 3 is applied to representative test problems. The goal of these tests is to provide a thorough comparison of several linear solver options and associated preconditioners, ultimately yielding a recommended approach for the application of the discrete-ordinates method to advanced response matrix methods. The test problems range from a relatively small BWR assembly model, ideally suited for a full analysis of the spectra of preconditioned-operators, to a larger PWR assembly that provides a suitable benchmark for computational performance.

Each of the preconditioners studied is based in some way on diffusion, for which a standard mesh-centered discretization is used. Although the consistency of the diffusion discretization used for DSA-preconditioning is not a focus of this paper, we note that the mesh-centered discretization is *not* consistent with the diamond-difference spatial discretization used for the transport equation for which an edge-centered discretization is known to be consistent [27].

All of the methods assessed have been implemented in `Detran`, an open-source, transport code written in C++ that was originally started at MIT and is now undergoing development at Kansas State University. `Detran` uses several external packages, including PETSc [28] for various core linear algebra functions and Python for a versatile front end.

4.1. Spectral Analysis of Multigroup Preconditioners

As recently noted by Hamilton, comparatively little work has been done to examine the spectrum (i.e., the set of eigenvalues λ) of discretized transport operators [5]. Such analysis is largely unuseful for studying the behavior of classical algorithms, such as the power method for eigenvalue problems or Gauss-Seidel or source iteration for fixed-source problems because the convergence of these algorithms is governed primarily by the dominant modes that are reasonably easy to analyze. Moreover, the detailed spectral analysis of an operator usually requires the use of dense linear algebra, and for all but the most trivial problems, the operators in question have been far too large to manipulate directly.

On modern computers, memory and processors have improved to the point where one can create an explicit representation of the transport operator and the various preconditioning operators. A full spectral analysis

can help provide an understanding of why Krylov methods work very well in many cases but fail in others, and how various preconditioners transform the spectrum in a way that renders problems easier to solve.

4.1.1. Test Problem Description

As an illustrative example, the 2-D, three-group, 9×9 pin BWR bundle described in Appendix A was studied subject to a constant source in all energy groups and uniformly-distributed throughout the entire domain, and vacuum boundaries on all sides. A diamond-difference spatial discretization and 6×6 Chebyshev-DP$_N$ product quadrature were used. To minimize the number of unknowns, the pins were discretized coarsely using a 3×3 mesh. Although the problem is relatively small, its size represents a lower-bound for transport problems typical of the response matrix methods that our research ultimately supports.

Figure 1 shows the sparsity pattern of the full transport operator—and it is certainly not a sparse operator. The structure of the full operator highlights the structure of the group-to-group scattering and fission matrices. Distinct columns arise in the upper right because only the fueled cells contribute to those blocks, while in the lower left, the columns are due to the fuel having no $3 \leftarrow 1$ scattering. The total size of the operator is 3267, and the number of nonzeros is 8382033, about 78% dense. In practice, such high density would make the use of explicit operators impractical. However, recall that all the operators used to create $\mathbf{A}$ are inherently sparse, even if never constructed. By going from a ψ-based representation to one for ϕ, we have essentially gone from a typically sparse differential representation to a characteristically dense integral representation.

4.1.2. Results

To obtain a preliminary understanding of the preconditioners developed in Section 3, the complete spectrum of $\mathbf{A}_{\text{MG}}$ and $\mathbf{A}_{\text{MG}}\mathbf{P}^{-1}$ were computed, where the preconditioner $\mathbf{P}$ is the multigroup DSA (DSA), coarse-mesh DSA (CMDSA), or transport-corrected CMDSA (TC+CMDSA). For CMDSA, two cases were investigated: one with no smoothing, and one with three smoothing iterations and $\omega = 0.7$ (denoted by CMDSA+S). A full pincell was selected as the coarse mesh, and a spatially flat spectrum was used for homogenization. For TC+CMDSA, three steps of an approximate transport operator were used based on a 2×2 product quadrature and CMDSA+S as the initial preconditioner. Initially, a 1×1 quadrature (one angle per octant)

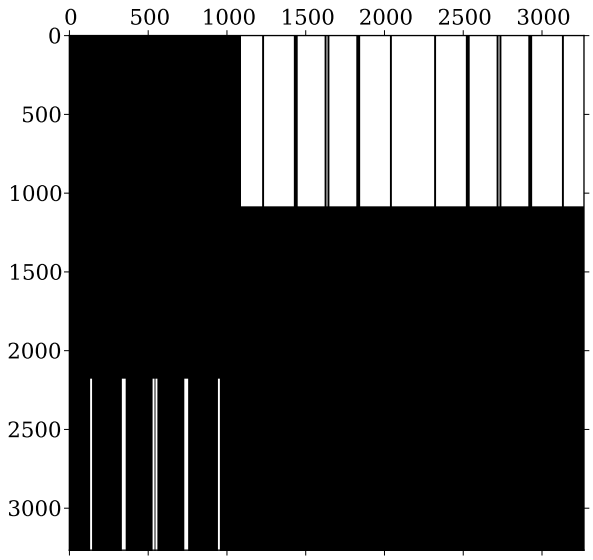


Figure 1: Three group transport operator sparsity pattern.

was used, but this led to instability. In all cases, any inversion of a diffusion operator was performed by LU-factorization.

The spectrum was computed as a function of k, the scaling parameter in the fission source of Eq. (1). When the homogeneous form of Eq. (1) is solved, k is an eigenvalue, often denoted by k_{eff}, that represents the ratio of production (from fission) to losses (from absorption and leakage at global boundaries). If all boundaries are subject to vacuum conditions, then $k_{\text{eff}} \equiv k_v \approx 0.2966$ for the BWR bundle, and for reflecting conditions, $k_{\text{eff}} \equiv k_\infty \approx 1.3$. In the limit $k \to k_v$, the operator $\mathbf{A}_{\text{MG}}$ becomes singular, which corresponds to an exact balance between production and losses, i.e., the system becomes "critical." Therefore, we limit our parametric study to values of k that satisfy $k_v < k \leq k_\infty$ and expect small values of k (and, hence, nearly-singular operators) to yield more challenging problems.

Figure 2 shows the spectrum of $\mathbf{A}_{\text{MG}}$ with each of the preconditioned spectra for $k = 1$. All of the preconditioners condense the eigenvalues to a smaller region, as desired. Specifically, DSA and CMDSA yield spectral shifts that are quite similar, but CMDSA leaves more of the spectrum below about 0.8 on the real axis. By adding smoothing, CMDSA+S shifts the spectrum toward unity and provides fairly distinct clustering. At the same time, it

leads to a comparatively significant complex component, but keep in mind the $\mathrm{Re}(\lambda)$ and $\mathrm{Im}(\lambda)$ scales are not the same. The addition of transport correction to CMDSA+S leads to a spectrum that visually appears to be nearly ideal: all values are tightly clustered away from the origin, in this case centering fairly close to unity.

To analyze the effect of the preconditioner more quantitatively, each preconditioner was tested for a range of k values. Figure 3 shows the Frobenius norm of the residual matrix $\mathbf{R} = \mathbf{I} - \mathbf{A}_{\mathrm{MG}}\mathbf{P}^{-1}$ and the number of right-preconditioned GMRES iterations for each preconditioner as a function of k. The Frobenius norm of the residual matrix measures how close a preconditioning process is to the inverse of the operator of interest. Equivalently, it measures how close the resulting spectrum is to unity because the eigenvalues of the preconditioned operator tend to 1 as $||\mathbf{R}||_F \to 0$. The number of GMRES iterations required is a more direct measure of a preconditioner's efficacy as better preconditioners usually lead to fewer iterations. For this study, GMRES was applied without restarts and converged to a tolerance of 10^{-8} on the relative norm.

The results indicate that the convergence of GMRES has a small dependence on k that is approximately the same for each preconditioner. For small values of $k - k_v$, a greater number of iterations is required, but the relative difference between each preconditioner is nearly constant. The Frobenius norm of the residual matrix depends more strongly on k, especially for the coarse-mesh preconditioners, although this dependence does not appear to indicate poor convergence.

Table 1 summarizes the results for $k = 1$. In addition, the table includes the standard deviation of the eigenvalues of each operator, which measures how well a preconditioner leads to clustered eigenvalues. Usually a preconditioner is successful (at least spectrally) if it reduces all of the parameters listed in Table 1. In practice, the reduction in iterations given a reduction in the Frobenius norm of the residual matrix or variance of the spectrum may be far from a linear relationship. For the cases studied, a preconditioner can be effective even if it yields a larger Frobenius norm of the residual matrix.

4.2. A Comparison of Spatial Shape Functions

The multigroup coarse-mesh DSA preconditioner (CMDSA) is based on the homogenization of the fine-mesh problem using an assumed shape function. As noted previously, a simple approximation is to assume a spatially-

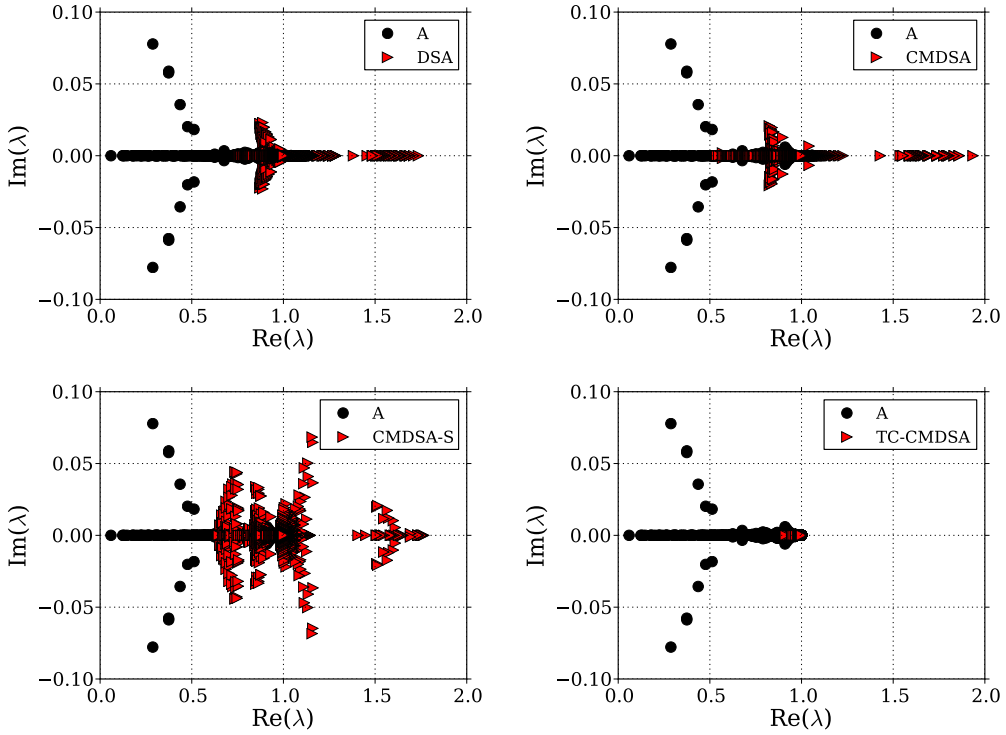


Figure 2: A comparison of the spectrum of $\mathbf{A}_{\mathrm{MG}}$ to several preconditioned spectra for $k = 1.0$.

	std(λ)	$\|\|\mathbf{I} - \mathbf{AP}^{-1}\|\|_F$	iterations
A	0.1253	17.1369	36
DSA	0.0917	19.7099	20
CMDSA	0.1179	29.0933	28
CMDSA-S	0.1018	20.9580	25
TC-CMDSA	0.0049	0.9037	9

Table 1: Spectrum diagnostic information

constant flux, but more accurate, potentially problem-dependent shape functions are possible and may lead to a better preconditioner.

4.2.1. Test Problem and Shape Function Descriptions

For application to reactor problems with arrays of fuel pins, shape functions can be generated for each unique pin type in isolation (i.e., a single pin subject to reflective boundary conditions) or for each of several pins in an array of pins (e.g., an assembly). To test these approaches, a simple 1-D model of 10 UO_2 and 10 MOX pins was developed as shown in Figure 4. The model was discretized using the diamond-difference approximation in space and a 16-angle, double Gauss-Legendre (DP_N) quadrature. The pins were represented by 12 spatial meshes. The UO_2, MOX(4.3%), and moderator data from the 7-group, 2-D C5G7 problem were used [29]. The left surface was subject to an isotropic source in the fast group, and vacuum conditions were imposed on the right.

In addition to a flat shape function, two problem-specific approaches were studied. The first approach used distinct spatial shapes for the UO_2 and MOX pincells based on solving the k-eigenvalue problem (i.e., the homogeneous form of Eq. (1) in which k is an eigenvalue) for each pincell with reflective boundaries. The resulting flux spatial distribution in each energy group was used for homogenization and prolongation. The second approach used the solution of the full test problem to define shape functions in each pin, i.e., each of the 20 pincells had unique shape functions.

4.2.2. Results

Right-preconditioned, non-restarted GMRES was used to solve the test problem with multigroup CMDSA and each shape function approach. No

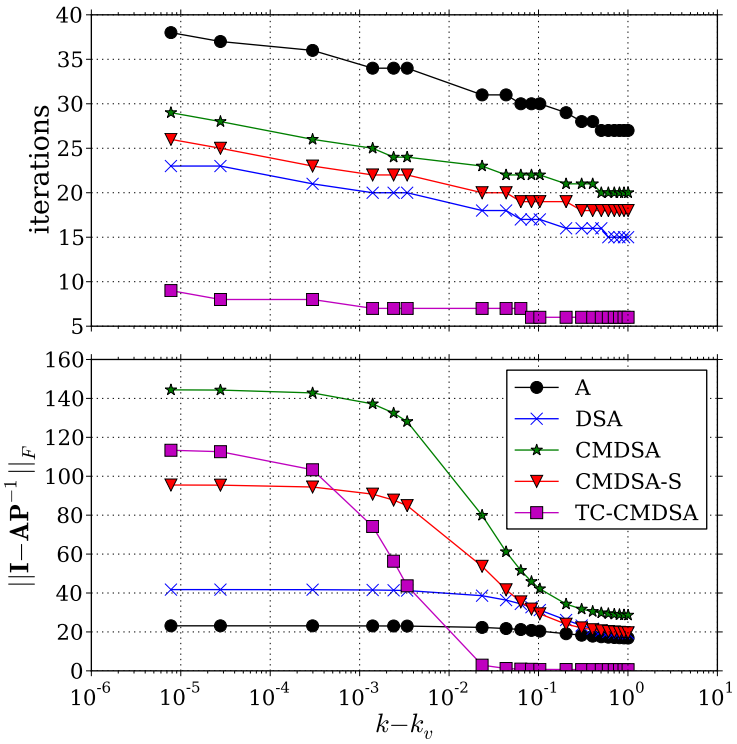


Figure 3: A comparison of two spectral parameters as a function of $k - k_v$, the difference between the parameter k and the k-eigenvalue of the BWR bundle subject to vacuum boundary conditions.

smoothing was applied, and the coarse meshes were taken to be full pincells. Table 2 provides the same spectral diagnostics discussed in Section 4.1 for the preconditioned operators constructed using each shape function. The results indicate that the use of an accurate spatial shape has little effect on the preconditioning process and can even reduce the efficacy. An extended study using multigroup CMDSA with smoothing, various ω values, and different coarse-mesh levels showed that the preconditioner based on the actual solution yielded at most a savings of one GMRES iteration.

4.3. Assessing Solver and Preconditioner Performance

4.3.1. Test Problem Description

For a preliminary performance assessment of solvers and preconditioners for fixed source multiplying transport problems, a single UO_2 assembly based

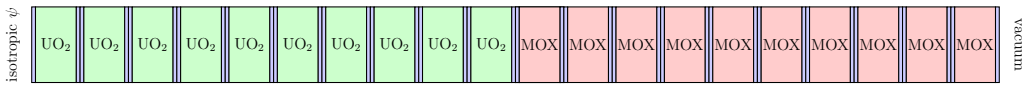


Figure 4: 1-D pin cell lattice schematic. The pin diameter is 1.08 cm, while the pin pitch is 1.28 cm.

	std(λ)	$\|\|\mathbf{I} - \mathbf{AP}^{-1}\|\|_F$	iterations
A	0.1182	7.7968	42
flat	0.0568	5.8662	16
pin	0.0567	6.6568	17
solution	0.0560	5.0343	16

Table 2: Effect of spatial shape function on CMDSA-preconditioned spectrum.

on the seven group 2-D C5G7 benchmark [29] was considered. The model was discretized in angle using an 8×8 angle-per-octant Chebyshev-DP$_N$ product quadrature, and in space using the diamond-difference discretization and a volume-conserving 7×7 mesh for each pin. A uniform, isotropic boundary source on the left side in the fast group was used. For reference, the k-eigenvalue of the assembly subject to vacuum boundary conditions is approximately 0.534, while the eigenvalue for the fully-reflected assembly is approximately 1.3. This model represents an approximate upper bound for the types of problems characteristic of 2-D response matrix analyses.

Three basic solver combinations were studied: Gauss-Seidel (GS) for the multigroup equations with source iteration (SI) for the within-group equations, GS with GMRES(30) for the within-group equations, and GMRES(30) for the multigroup equations. Recall that GMRES(N) refers to GMRES with a restart of N. A value of $N = 30$ was selected because it has worked well in our experience and, for the problems studied, led to few restarts in most cases.

For SI, two cases were considered. The first used a maximum of 20 inner iterations (denoted by SI(20)), while the second used a maximum of just one iteration (denoted by SI(1)), yielding what has been called the "flattened" transport operator [30]. For GMRES applied to the within-group

equations, DSA preconditioning was applied to the right using either LU or ILU-preconditioned GMRES(30) to invert the diffusion operator. The use of LU represents an "exact" inversion, while the GMRES solver was set with a tolerance equal to that of the transport solver. In addition, ILU was applied using two levels of factorization, with all other remaining parameters set to their PETSc default values.

For GMRES applied to the multigroup equations, several preconditioners were investigated. A fine-mesh, multigroup DSA was studied, using the same LU and GMRES(30)+ILU settings as the within-group DSA preconditioner. In addition, CMDSA was tested with and without smoothing. The smoothed version used three iterations and $\omega = 0.7$. Homogenization was performed over the pincell with a flat spectrum. Smoothed CMDSA (CMDSA+S) was used as the initial preconditioner for the transport corrected preconditioner (TC-CMDSA), for which two cases were considered. In the first case (TC^1), three updates with the coarse operator were used followed by one update using the full transport operator. In the second case (TC^2), just one update were performed with the coarse operator, and the full transport operator was not applied.

Because a preconditioner based on the use of GMRES(30)+ILU is not exactly inverted, its action is generally not the same at each iteration, and, hence, the convergence of GMRES is not guaranteed. Although no numerical difficulties were encountered for the problems studied, convergence would only be guaranteed if a flexible variant of GMRES (i.e., FGMRES) were used [20].

4.3.2. Measuring Performance and Convergence

In this study, the number of space-angle sweeps and the total wall time were used as metrics for performance. A space-angle transport sweep, denoted by the operator $\mathbf{T} = \mathbf{D}\mathbf{L}^{-1}$ in Section 2, usually is the single most computationally-intensive part of the iterative schemes analyzed. The application of $\mathbf{T}$ is especially costly when the number of angles is large, rendering the essentially space- and energy-dependent expenses related to source construction insignificant. However, as the overhead of Krylov solvers or their preconditioners grows, the sweeps alone fail to tell the whole story, and the computational time becomes the deciding factor.

In addition to a meaningful performance metric, a consistent criterion for convergence is needed to ensure each method leads to the same result within some tolerance. The typical convergence criteria employed for SI and

GS (i.e., the difference between successive flux iterates) differ from that of GMRES (i.e., a residual norm). By using a tightly converged solution x_{ref} as a reference, the tolerance of each method was set to produce an approximate solution x that satisfies

$$||x - x_{\text{ref}}||_2 / ||x_{\text{ref}}||_2 < \tau, \tag{51}$$

where x is a vector that contains the full multigroup flux, and τ is the desired tolerance on the error.

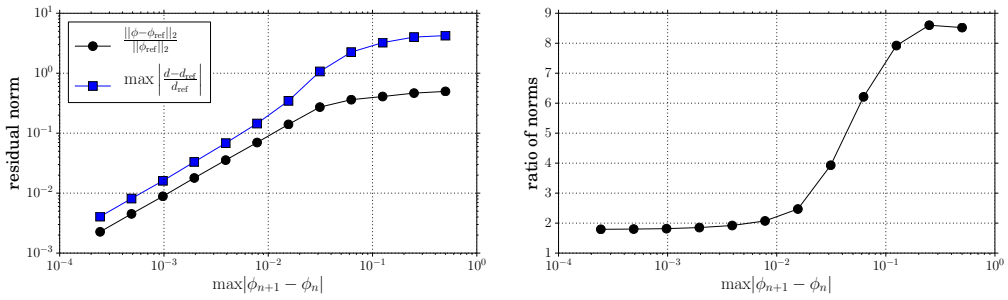
Each of the methods was tested using the tolerance defined by Eq. (51) for values of $\tau = 10^{-4}$, 10^{-6}, and 10^{-8}. A tolerance on the multigroup flux directly, though easy to implement, is not necessarily easy to interpret physically. In reality, the analyst cares about reaction rates, and for the given example, pin powers are a relevant metric. Here, pin power errors are measured by the maximum relative error in the pin fission rate, defined by

$$\epsilon_{\text{pin}} = \max_{p \in \text{pins}} \left| \frac{d_p - d_{p,\text{ref}}}{d_{p,\text{ref}}} \right|, \tag{52}$$

where d_p represents the total fission rate in any pin. To help correlate flux errors to pin fission rate errors, Figure 5a provides both errors as a function of the successive flux tolerance as used in GS+SI(20), while Figure 5b shows the ratio of the norms as a function of GS+SI(20) tolerance. Figure 5a shows that the flux norm is numerically less conservative than the pin-based norm, i.e., given an actual error in the flux, the actual error in the pin fission rates is larger. Figure 5b shows that the ratio of the norms is not monotonic, but in the limit of tight convergence, the ratio of the flux and pin power error norms approaches an asymptotic value just under two. This behavior is likely problem-specific but serves to illustrate that error norms for one quantity may not be entirely applicable to another quantity. Similarly, the norm of the successive flux difference, used as the GS convergence criterion, is even less conservative by an additional order of magnitude. This fact is sometimes overlooked, and tolerances that seem appropriately tight are known to yield premature convergence [31]. Hence, the use of a consistent metric for comparison is crucial.

4.3.3. Results

Tables 3-5 provide the number of sweeps, execution time, residual norm tolerance, and final relative error for each solver for $k = 1.0$. In addition,



(a) Error norms.

(b) Ratio of error norms.

Figure 5: Flux and pin fission rate errors (5a) and their ratio (5b) as functions of successive flux difference norm.

Figures 6 and 7 show the number of sweeps (divided by the number of groups) and the computational time for several, representative methods as a function of $k - k_v$ for $\tau = 10^{-4}$.

The results provided by Tables 3–5 support our general observation that converging the inner, within-group iterations leads to wasted effort in the classical GS+SI scheme, with the flattened scheme becoming more effective with a tighter tolerance. The results also show that (unpreconditioned) GS+GMRES led to a small improvement over GS+SI(20) with respect to both sweeps and time, but diminishingly so for tighter tolerances. Only at loose tolerances did GS+GMRES outperform GS+SI(1). This trend is essentially the same as observed for GS+SI(20) and GS+SI(1); sweeps used to converge the inner iterations are not as valuable as additional outer iterations. However, because the sweeps applied by GS+GMRES for the within-group problem are comparatively more effective than GS+SI(20) sweeps, the performance degradation is not as great.

The application of DSA(LU) preconditioning to GS+GMRES for the within-group problem further reduced the computational time to just over half the time of GS+SI(20), with a similar decrease in sweep count. The ILU-based implementation outperformed GS+SI(1) and GS+SI(20) in all cases but was more expensive than the LU-based implementation, which means that the small, one-group diffusion problem representing the DSA process is easily solved directly by LU elimination. This observation would likely not be true for large, 3-D problems.

The use of GMRES for the full multigroup equations led to a reduction

Table 3: Sweeps, timings, residual norm tolerance, and actual relative error for a C5G7 assembly with $\tau = 1.0 \times 10^{-4}$

	sweeps	$\frac{\text{sweeps}}{\text{groups}}$	time (s)	residual	error
GS, SI(20)	1684	240	68.4	1.1×10^{-5}	9.8×10^{-5}
GS, SI(1)	1589	227	67.9	1.8×10^{-5}	9.8×10^{-5}
GS, GMRES	1266	180	51.6	1.3×10^{-5}	8.0×10^{-5}
GS, DSA(LU)	858	122	36.8		1.5×10^{-5}
GS, DSA(ILU)	858	122	43.8		1.5×10^{-5}
GMRES	168	24	7.0	3.8×10^{-5}	6.1×10^{-5}
DSA(LU)	42	6	10.9		1.4×10^{-5}
DSA(ILU)	42	6	6.6		1.3×10^{-5}
CMDSA	63	9	2.8		3.3×10^{-5}
CMDSA+S	56	8	2.5		2.9×10^{-5}
TC^1-CMDSA+S	49	7	3.9		9.1×10^{-6}
TC^2-CMDSA+S	42	6	2.1		3.0×10^{-5}

in sweeps and computational time by roughly an order of magnitude compared to GS+SI(20), with slight improvement for tighter convergence. All of the multigroup preconditioners successfully reduced the number of sweeps. However, the computational time of MG-DSA(LU) was more expensive than unpreconditioned GMRES for all tolerances and $k = 1$, and MG-DSA(ILU) was less expensive than unpreconditioned GMRES only for $\tau = 10^{-4}$.

Compared to multigroup DSA, multigroup CMDSA led to approximately 1.5–2 times as many sweeps as MG-DSA(ILU) but with execution times just under that of MG-DSA(ILU). Because the CMDSA process used homogenization over 7×7 cells, the inversion of the corresponding coarse mesh diffusion operator was small, even by LU factorization. By adding smoothing (denoted CMDSA+S), the cost of applying the fine mesh diffusion operator (*not* its inverse) was incurred, but the reduction in sweeps led to a slight yet increasing reduction in execution time with tighter tolerance.

Finally, the addition of transport correction to CMDSA+S further reduced the number of sweeps but at the cost of applying the low-order trans-

Table 4: Sweeps, timings, residual norm tolerance, and actual relative error for a C5G7 assembly with $\tau = 1.0 \times 10^{-6}$

	sweeps	$\frac{\text{sweeps}}{\text{groups}}$	time (s)	residual	error
GS, SI(20)	2960	422	121.6	1.1×10^{-7}	9.9×10^{-7}
GS, SI(1)	2569	367	108.0	9.1×10^{-8}	5.0×10^{-7}
GS, GMRES	2435	347	102.2	1.8×10^{-7}	8.0×10^{-7}
GS, DSA(LU)	1517	216	65.2		1.8×10^{-7}
GS, DSA(ILU)	1517	216	84.3		1.8×10^{-7}
GMRES	217	31	9.1	2.7×10^{-7}	4.4×10^{-7}
DSA(LU)	56	8	12.0		1.4×10^{-7}
DSA(ILU)	56	8	12.8		2.3×10^{-7}
CMDSA	98	14	4.3		9.6×10^{-8}
CMDSA+S	84	12	3.8		1.2×10^{-7}
TC^1-CMDSA+S	63	9	5.2		9.2×10^{-8}
TC^2-CMDSA+S	63	9	3.2		6.9×10^{-8}

port operator and, possibly, the full transport operator. Because the low-order operator used 4 angles per octant, and the high-order operator used 64 per octant, one high-order space-angle sweep should be equivalent to 16 low-order space-angle sweeps. This is not entirely true in practice because the construction of the scattering source is independent of the quadrature. Even so, for $\tau = 10^{-8}$, TC^1-CMDSA+S required 588 low order sweeps, equivalent to 36 high-order sweeps. Hence, a more representative sweep count is 127 rather than 91. However, its cost at 7.6 s is higher than the cost of CMDSA at about the same sweep count, and this apparent discrepancy can be attributed to the angle-independent costs of the low order operator in addition to the cost of the single application of the high-order operator. On the other hand, the omission of the high-order operator and the reduction from three to just one application of the low-order operator led to a slight reduction in the number of sweeps and a reduction in the time by nearly half.

Figures 6 and 7 provide further evidence that the coarse-mesh, multigroup preconditioners were effective—even when the underlying multigroup

Table 5: Sweeps, timings, residual norm tolerance, and actual relative error for a C5G7 assembly with $\tau = 1.0 \times 10^{-8}$

	sweeps	$\frac{\text{sweeps}}{\text{groups}}$	time (s)	residual	error
GS, SI(20)	4345	620	178.0	1.0×10^{-9}	9.4×10^{-9}
GS, SI(1)	3297	471	135.0	1.8×10^{-9}	9.9×10^{-9}
GS, GMRES	3681	525	151.4	3.7×10^{-9}	9.8×10^{-9}
GS, DSA(LU)	2214	316	95.6		3.8×10^{-9}
GS, DSA(ILU)	2214	316	132.5		3.8×10^{-9}
GMRES	280	40	11.9	3.0×10^{-9}	9.7×10^{-9}
DSA(LU)	70	10	12.9		1.6×10^{-9}
DSA(ILU)	70	10	20.5		2.2×10^{-9}
CMDSA	126	18	5.5		1.7×10^{-9}
CMDSA+S	105	15	4.7		2.7×10^{-9}
TC[1]-CMDSA+S	91	13	7.6		3.5×10^{-11}
TC[2]-CMDSA+S	77	11	3.9		1.8×10^{-9}

GMRES solver was challenged. For $k \approx k_v$, the spectral radius of the transport operator approaches unity, which renders GS prohibitively expensive for $k - k_v < 0.1$ regardless of the within-group solver. Even unpreconditioned GMRES was significantly challenged, requiring nearly 2000 sweeps to converge in the worst case. A restart value larger than 30 would likely yield some improvement in the unpreconditioned case but not the others because all of the multigroup preconditioners reduced the number of iterations to less than 20 (the number of sweeps divided by the number of groups is equal to one plus the number of GMRES iterations). Each of the coarse-mesh, multigroup preconditioners significantly reduced the computational time and the number of sweeps compared to unpreconditioned GMRES, and all of the preconditioners effectively reduced the number of sweeps. The coarse-mesh, multigroup preconditioners were especially effective for small values of $k - k_v$, for which the reduction in the computational time was over an order of magnitude compared to unpreconditioned GMRES. Although the performance of each multigroup preconditioner with respect to sweeps and computational

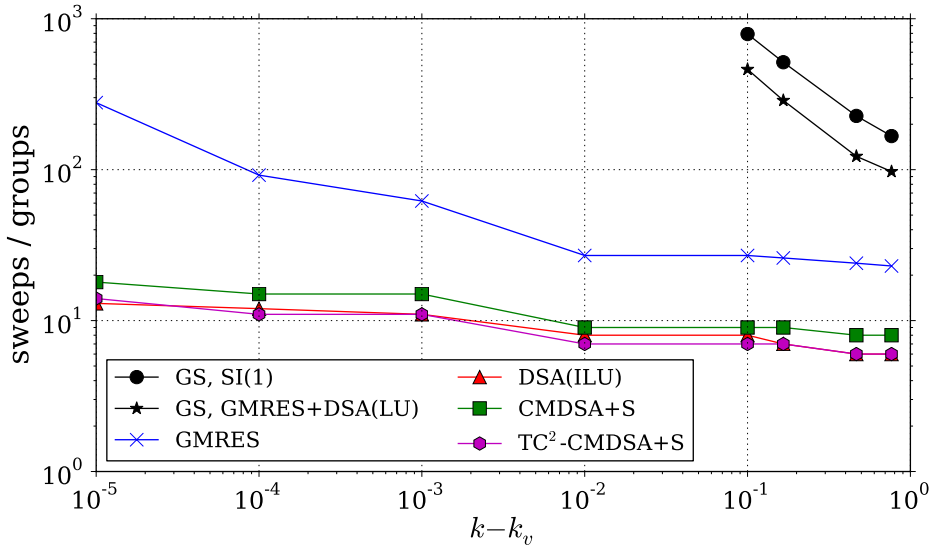


Figure 6: The number of sweeps (divided by the number of groups) as a function of $k - k_v$, the difference between the parameter k and the k-eigenvalue of the PWR assembly subject to vacuum boundary conditions.

time degraded somewhat for smaller $k - k_v$ (relative to the performance of the same preconditioner at, e.g., $k = 1$), the relative increases in the number of sweeps and computational time was small (approximately a factor of 2–3) compared to the increases observed for unpreconditioned GMRES (approximately a factor of 10). Based on the results of Figures 6 and 7 and Tables 3–5, TC^2-CMDSA-S led to the best overall performance.

5. Conclusion

Several transport preconditioners based on the diffusion approximation have been developed and applied to representative fixed-source, multiplying problems. In particular, it was found that the two-grid, coarse-mesh diffusion preconditioner performs quite well, especially when used with a fine-mesh diffusion smoother and transport-correction. Although several combinations were considered, it is recognized that many possible parameter permutations were omitted. Some of these were eliminated based on past experience, and others were simply deemed outside the intended scope. Of the latter, a more careful study of the coarse-mesh parameters (meshing, spectra, and smooth-

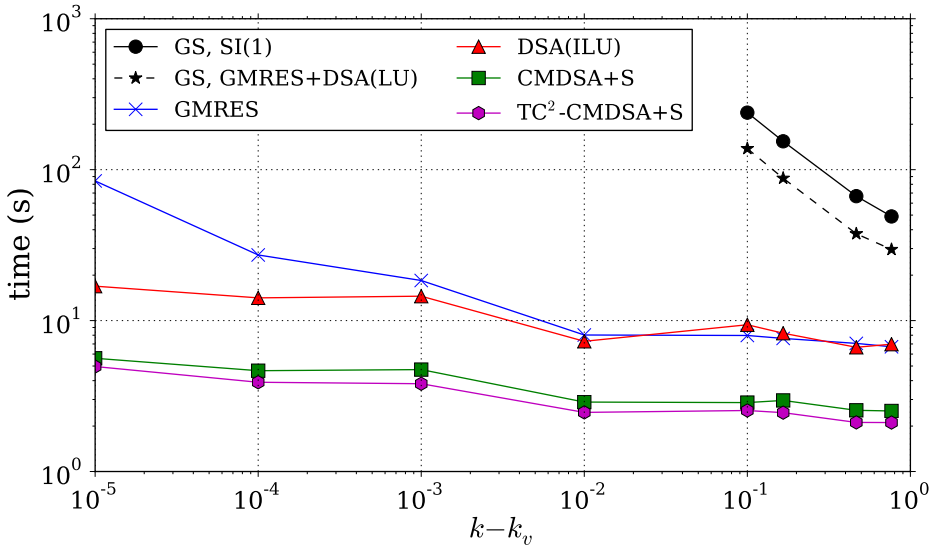


Figure 7: The computational time (divided by the number of groups) as a function of $k - k_v$, the difference between the parameter k and the k-eigenvalue of the PWR assembly subject to vacuum boundary conditions.

ing, *etc.*) and preconditioners for the fine-mesh diffusion solver would be of value. Our efforts have largely been in support of related work on response matrix methods, but the methods developed and results obtained should be of interest to the broader transport and reactor physics communities.

Future efforts should explore at least two areas not addressed by this work. First, the energy variable should be incorporated as an additional variable subject to homogenization. Although our scoping studies proved unsuccessful, recent work on energy-based multigrid methods shows significant promise [4]. Second, the application of the methods developed has been limited to the relatively small problems of importance to response matrix analyses. More work should examine the utility of diffusion-based preconditioners for large-scale, parallel computation in all areas of radiation transport for which multigroup Krylov methods have yielded so much success.

Acknowledgements

Appendix A. Three Group BWR Bundle

To provide a relatively small transport problem for examining the transport operator in detail, a small BWR bundle was modeled, illustrated in Figure A.8. The model contains just two materials, a smeared fuel-clad mixture and a moderator, and consists of 1.15 cm square pins with a 1.4 cm pitch and an exaggerated water gap. The cross-section data are provided in Tables A.6 and A.7.

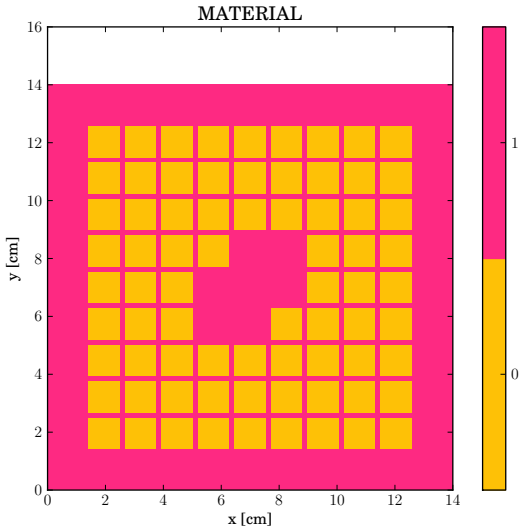


Figure A.8: Simplified BWR bundle. Zero indexes fueled regions while one indexes moderator regions.

References

[1] P. K. Romano, B. Forget, The OpenMC Monte Carlo Particle Transport Code, Annals of Nuclear Energy 51 (2013) 274 – 281.

[2] T. M. Evans, G. G. Davidson, R. N. Slaybaugh, Three-Dimensional Full Core Power Calculations for Pressurized Water Reactors, in: Journal of Physics: Conference Series, SciDAC, Vol. 68, 2010.

	$g = 1$	$g = 2$	$g = 3$
Σ_{tg}	0.2822058997	0.4997685502	0.4323754911
Σ_{fg}	0.0028231045	0.0096261203	0.1123513981
ν_g	2.7202775245	2.4338148428	2.4338000000
χ_g	0.9996892490	0.0003391680	0.0000000000
$\Sigma_{s1\leftarrow g}$	0.2760152893	0.0000000000	0.0000000000
$\Sigma_{s2\leftarrow g}$	0.0011230014	0.4533430274	0.0000378305
$\Sigma_{s3\leftarrow g}$	0.0000000000	0.0014582502	0.2823864370

Table A.6: Material 0 (fuel) cross sections.

	$g = 1$	$g = 2$	$g = 3$
Σ_{tg}	0.3333694980	0.5887110656	1.6512246291
$\Sigma_{s1\leftarrow g}$	0.2432946408	0.0000000000	0.0000000000
$\Sigma_{s2\leftarrow g}$	0.0898364840	0.4413753398	0.0000203109
$\Sigma_{s3\leftarrow g}$	0.0000387911	0.1465683257	1.6300848232

Table A.7: Material 1 (moderator) cross sections.

[3] J. S. Warsa, T. A. Wareing, J. E. Morel, Krylov Iterative Methods and the Degraded Effectiveness of Diffusion Synthetic Acceleration for Multidimensional S_N Calculations in Problems with Material Discontinuities, Nuclear Science and Engineering 147 (3) (2004) 218–248.

[4] R. N. Slaybaugh, T. M. Evans, G. G. Davidson, P. P. H. Wilson, Multigrid in Energy Preconditioner for Krylov Solvers, Journal of Computational Physics 242 (2013) 405–419.

[5] S. P. Hamilton, Numerical Solution of the k-Eigenvalue Problem, Ph.D. thesis, Emory University (2011).

[6] J. A. Roberts, B. Forget, Solving Eigenvalue Response Matrix Equations with Nonlinear Techniques, Annals of Nuclear Energy 69 (2014) 97–107.

[7] J. A. Roberts, B. Forget, Krylov Subspace Iteration for Eigenvalue Response Matrix Equations , in: Proceedings of the International Con-

ference on the Physics of Reactors (PHYSOR 2012), American Nuclear Society, 2012.

[8] J. A. Roberts, B. Forget, Solving Eigenvalue Response Matrix Equations using Jacobian-Free Newton-Krylov Methods, in: Proceedings of the International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2011), American Nuclear Society, 2011.

[9] J. J. Duderstadt, L. J. Hamilton, Nuclear Reactor Analysis, John Wiley and Sons, Inc., 1976.

[10] E. E. Lewis, W. F. Miller, Computational Methods of Neutron Transport, American Nuclear Society, 1993.

[11] E. W. Larsen, J. E. Morel, Advances in Discrete-Ordinates Methodology, in: Nuclear Computational Science, Springer, 2010, pp. 1–84.

[12] B. T. Adams, J. E. Morel, A Two-Grid Acceleration Scheme for the Multigroup S_N Equations with Neutron Upscattering, Nuclear Science and Engineering 115 (3) (1993) 253–264.

[13] Y. Saad, Iterative Methods for Sparse Linear Systems, SIAM, 2003.

[14] Y. Saad, M. H. Schultz, GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems, SIAM Journal on Scientific and Statistical Computing 7 (3) (1986) 856–869.

[15] J. E. Morel, J. M. McGhee, A Fission-Source Acceleration Technique for Time-Dependent Even-Parity S_N Calculations, Nuclear Science and Engineering 116 (2) (1994) 73–85.

[16] K. S. Smith, Nodal Method Storage Reduction by Nonlinear Iteration, Transactions of the American Nuclear Society 44 (1984) 265.

[17] K. S. Smith, J. D. Rhodes III, Full-Core, 2-D, LWR Core Calculations with CASMO-4E, in: Proceedings of the International Conference on the Physics of Reactors (PHYSOR 2002), 2002, pp. 7–10.

[18] W. L. Briggs, V. E. Henson, S. F. McCormick, A Multigrid Tutorial, SIAM, 2000.

[19] K. S. Smith, Assembly Homogenization Techniques for Light Water Reactor Analysis, Progress in Nuclear Energy 17 (3) (1986) 303–335.

[20] Y. Saad, A Flexible Inner-Outer Preconditioned GMRES Algorithm, SIAM Journal on Scientific Computing 14 (2) (1993) 461–469.

[21] J. S. Warsa, T. A. Wareing, J. E. Morel, Solution of the Discontinuous P1 Equations in Two-Dimensional Cartesian Geometry with Two-Level Preconditioning, SIAM Journal on Scientific Computing 24 (6) (2003) 2093–2124.

[22] B. W. Patton, J. P. Holloway, Application of Preconditioned GMRES to the Numerical Solution of the Neutron Transport Equation, Annals of Nuclear Energy 29 (2) (2002) 109–136.

[23] H. Zhang, H. Wu, L. Cao, An Acceleration Technique for 2D MOC Based on Krylov Subspace and Domain Decomposition Methods, Annals of Nuclear Energy 38 (12) (2011) 2742 – 2751.

[24] G. Schulz, Iterative Berechnung der Reziproken Matrix, Zeitschrift für Angewandte Mathematik und Mechanik 13 (1933) 57–59.

[25] A. S. Householder, The Theory of Matrices in Numerical Analysis, Dover, 1975.

[26] E. Chow, Y. Saad, Approximate Inverse Preconditioners via Sparse-Sparse Iterations, SIAM Journal on Scientific Computing 19 (3) (1998) 995–1023.

[27] R. Alcouffe, Diffusion Synthetic Acceleration Methods For the Diamond-Differenced Discrete-Ordinates Equations, Nuclear Science and Engineering 64 (1977) 344–355.

[28] 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.

[29] 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 (C5G7 MOX), NEA/NSC.

[30] D. F. Gill, Y. Y. Azmy, J. S. Warsa, J. Densmore, Newton's Method for the Computation of k-Eigenvalues in S_N Transport Applications, Nuclear Science and Engineering 168 (1) (2011) 37–58.

[31] M. L. Adams, E. W. Larsen, Fast iterative methods for Discrete-Ordinates Particle Transport Calculations, Progress in Nuclear Energy 40 (1) (2002) 3–159.