Nuclear Mathematical and Computational Sciences: A Century in Review, A Century Anew
Gatlinburg, Tennessee, April 6-11, 2003, on CD-ROM, American Nuclear Society, LaGrange Park, IL (2003)

KRYLOV ITERATIVE METHODS APPLIED TO MULTIDIMENSIONAL S_N CALCULATIONS IN THE PRESENCE OF MATERIAL DISCONTINUITIES

James S. Warsa, Todd A. Wareing, Jim E. Morel
Transport Methods Group
Los Alamos National Laboratory
Los Alamos, NM 87545-0001
warsa@lanl.gov, wareing@lanl.gov, jim@lanl.gov

ABSTRACT

We show that a Krylov iterative method, preconditioned with DSA, can be used to efficiently compute solutions to diffusive problems with discontinuities in material properties. We consider a lumped, linear discontinuous discretization of the S_N transport equation with a "partially consistent" DSA preconditioner. The Krylov method can be implemented in terms of the original S_N source iteration coding with little modification. Results from numerical experiments show that replacing source iteration with a preconditioned Krylov method can efficiently solve problems that are virtually intractable with accelerated source iteration.

Key Words: Krylov iterative methods, discrete ordinates, deterministic transport methods, diffusion synthetic acceleration

1 INTRODUCTION

A spatial discretization of the DSA diffusion equations that is consistent with the discretization of the transport equation is usually considered a sufficient condition for a DSA method to be unconditionally effective [1–3]. However, the degradation of DSA methods – even fully consistent ones – in problems with discontinuities in material properties means that consistency is not enough to guarantee the effectiveness of a DSA method. This was first identified in [4] and [5] and revealed as a general deficiency of DSA in a paper at this conference.

For this paper, we follow on the work of Ashby, et al. [6], Brown [7], and Guthrie, et al. [8], where Krylov methods preconditioned by DSA replace traditional source iteration on the scalar flux, and extend their approach to our linear discontinuous finite element method (DFEM) on unstructured tetrahedral grids. This discretization, including a discussion of compatible DSA methods, is presented in Sec. 2.

We find that using a more powerful iteration, like a Krylov subspace iterative method [9], significantly improves convergence for problems in which the convergence of accelerated source iteration degraded in the presence of material discontinuities. A nice feature is that the Krylov iterative method can be "wrapped around" the source iteration code so that only minor changes to the original inner iteration coding is necessary. A brief discussion of the formulation and implementation of the preconditioned Krylov iterative solution method, including an overview of related work, is presented in Sec. 3

We show numerical results for a realistic, unstructured mesh problem with a range of differing material properties in Sec. 4. We find that (a) the Krylov method alone, without DSA preconditioning, accelerates the transport solution, although not always as well as accelerated source iteration, (b) convergence is vastly improved if the Krylov method is preconditioned with DSA, and (c) the Krylov method restores the effectiveness of the partially consistent, but inexpensive, simplified WLA (S-WLA) DSA scheme [10, 11]. This last observation is also true for the fully consistent DSA method, but the high costs we encountered in the fully consistent equations is unacceptable [12, 13], particularly costly in problems with material discontinuities. Therefore we only consider the less costly S-WLA method for preconditioning the Krylov method.

2 DISCONTINUOUS FINITE ELEMENT DISCRETIZATION ON TETRAHEDRAL MESHES

We present the linear discontinuous finite element method (DFEM) for the S_N transport equation on unstructured tetrahedral meshes, followed by a brief overview of DSA. Further details on the fully consistent DSA scheme can be found in [13] and details of the partially consistent DSA method can be found in [10] and [11].

2.1 Discontinuous Finite Element Discretization

The notation used here has the usual meaning [14] and we assume cgs units. Given an angular quadrature set with N specified nodes and weights $\{\hat{\Omega}_m, w_m\}$, an isotropic distributed source of particles $Q_0(\boldsymbol{r})$ and anisotropic scattering of order L, the monoenergetic, steady-state S_N transport equation in the three-dimensional domain $\boldsymbol{r} \in V$ with boundary $\boldsymbol{r}_s \in \partial V$, is

$$\hat{\Omega}_m \cdot \nabla \psi_m(\boldsymbol{r}) + \sigma_t(\boldsymbol{r})\psi_m(\boldsymbol{r}) = \sum_{l=0}^{L} \sigma_{s,l} \sum_{n=-l}^{l} Y_{ln}(\hat{\Omega}_m)\phi_l^n(\boldsymbol{r}) + Q_0(\boldsymbol{r}), \quad m = 1, \ldots, N. \tag{1a}$$

Here, $Y_{ln}(\hat{\Omega})$ are the normalized spherical harmonics functions and the scalar flux moments are

$$\phi_l^n(\boldsymbol{r}) = \sum_{m=1}^{N} w_m Y_{ln}(\hat{\Omega}_m)\psi_m(\boldsymbol{r}). \tag{1b}$$

The linear DFEM discretization is specified by the following variational formulation. Given an angular flux expansion in terms of the four independent linear basis functions on a tetrahedral cell T_k,

$$\psi_{m,k} = \sum_{j=1}^{4} \psi_{m,j,k} L_j(\boldsymbol{r}), \tag{2a}$$

find the linear approximation for each angle $\hat{\Omega}_m$ that satisfies

$$\begin{aligned} \hat{\Omega}_m \cdot \left(\int_{\partial T_k} \hat{n}\, \psi_m^b\, u_j\, dS - \int_{T_k} \psi_{m,k} \nabla u_j\, dV \right) + \sigma_{tk} \int_{T_k} \psi_{m,k}\, u_j\, dV \\ = \sum_{l=0}^{L} \sigma_{s,l,k} \sum_{n=-l}^{l} Y_{ln}(\hat{\Omega}_m) \int_{T_k} \phi_{l,k}\, u_j\, dV + \int_{T_k} Q_0 u_k\, dV\ , \end{aligned} \tag{2b}$$

$$\phi_{l,k} = \sum_{m=1}^{N} w_m Y_{ln}(\hat{\Omega}_m)\psi_{m,k} \,, \tag{2c}$$

for all linear trial functions $u_j, j = 1, \ldots, 4$ on cell T_k. The Galerkin approximation takes the trial functions to be the basis functions L_j, and the above expressions can be evaluated for each of these four functions. This gives four equations for the four unknowns $\psi_{m,j,k}$ on the cell. Before carrying out the integrations in (2b), however, we first introduce the discontinuous approximation. Considering a cell k with face p whose outward normal is $\hat{n}_p$, the boundary terms ψ_m^b are defined as

$$\Big(\hat{\Omega}_m \cdot \hat{n}_p\Big)\psi_m^b = \begin{cases} \Big(\hat{\Omega}_m \cdot \hat{n}_p\Big)\psi_{m,i(p),k}, & \hat{\Omega}_m \cdot \hat{n}_p > 0, \quad \hat{n}_p \text{ in } V \\ \Big(\hat{\Omega}_m \cdot \hat{n}_p\Big)\psi_{m,i(p),l}, & \hat{\Omega}_m \cdot \hat{n}_p < 0, \quad \hat{n}_p \text{ in } V \backslash \partial V \\ \Big(\hat{\Omega}_m \cdot \hat{n}_p\Big)\Gamma(\hat{\Omega}_m), & \hat{\Omega}_m \cdot \hat{n}_p < 0, \quad \hat{n}_p \text{ on } \partial V \end{cases} \tag{2d}$$

where l is the cell that shares face p with cell k. The subscript $i(p)$ denotes three vertices i on a face p of a given cell. Simply put, if $\hat{n}_p$ is on the boundary of the problem domain V, then the boundary condition is used to define the incoming angular flux for the three points on a face; otherwise the internal or external values angular fluxes are used depending on the orientation of the cell face with respect to the quadrature direction. The discrete boundary conditions are vacuum, $\Gamma(\hat{\Omega}_m) = 0$, or $\Gamma(\hat{\Omega}_m) = \psi_{m',i(p),s}$ for reflective boundary conditions, where m' is determined by the relationship

$$\hat{\Omega}_{m'} = \hat{\Omega}_m - 2\,\hat{n}\,\Big(\hat{\Omega}_m \cdot \hat{n}\Big)\,, \tag{2e}$$

for $\hat{\Omega}_m$ and $\hat{n} = \hat{n}_p$. In our application, reflection is implemented only for boundary faces aligned parallel to the x, y or z coordinate axes so that the standard quadrature sets we use contain the reflected angles $\hat{\Omega}_{m'}$ that satisfy this relationship.

The integrals in (2) are evaluated, either analytically or by quadrature approximation, for every cell in the mesh. The angular flux, $\psi_{m,j,k}$, can then be computed for all vertices $j = 1, 4$ of every cell k, one cell at a time over the entire mesh in a predetermined order for every quadrature angle $\hat{\Omega}_m$. Note that we use a fully lumped version of (2). Describing it goes beyond the scope of this work, but suffice it to say that this lumping preserves the diffusion limit in thick, diffusive regimes (see [15]).

2.2 Source Iteration in Operator Notation

In this section, we formulate the transport equation in an operator notation to facilitate our presentation. We assume that we are using a standard S_N angular discretization. Postponing discussion of the boundary conditions, the discretized transport problem reads

$$\boldsymbol{L}\psi = \boldsymbol{MSD}\psi + q. \tag{3}$$

We use an N-point quadrature and there are N_c spatial cells in the problem. Let ψ be the vector of angular fluxes for every angle and every vertex (four of them) in each cell, so ψ is of length $n = 4NN_c$. The vector q is source vector also of length n. The $(n \times n)$ operator $\boldsymbol{L}$ represents the discretized streaming and removal operator for all angles. The vector ϕ contains the N_m (this number depends on L and the particular quadrature) of scalar flux moments at the four vertices of each cell, so that is is of length $t = 4N_mN_c$. The operator $\boldsymbol{D}$ maps the vector ψ onto ϕ, that is, $\phi = \boldsymbol{D}\psi$. The operator $\boldsymbol{M}$ maps a vector

of scalar flux moments onto angular fluxes although it should be noted that $\psi \neq \boldsymbol{M}\phi$. These are in general rectangular operators, $\boldsymbol{D}$ being $(t \times n)$ and $\boldsymbol{M}$ is $(n \times t)$. In the standard S_N scattering treatment $\boldsymbol{D} = \boldsymbol{M}^T\boldsymbol{W}$, where $\boldsymbol{W}$ is an $(n \times n)$ diagonal operator of the quadrature weights distributed to all vertices of all the cells. The operator $\boldsymbol{S}$ is a $(t \times t)$ diagonal operator of the scattering cross sections on the cells, distributed to all vertices on all cells.

Rearranging (3) and introducing an iteration index ℓ, we get traditional source (or Richardson) iteration:

$$\psi^{\ell+1} = \boldsymbol{L}^{-1}\left(\boldsymbol{M}\boldsymbol{S}\phi^{\ell} + q\right) \tag{4a}$$

$$\phi^{\ell+1} = \boldsymbol{D}\psi^{\ell+1}. \tag{4b}$$

with the boundary conditions for ψ:

$$\boldsymbol{B}_0\psi = g \quad \text{on } \Gamma_0 \tag{5a}$$

$$\boldsymbol{B}_R\psi = 0 \quad \text{on } \Gamma_R. \tag{5b}$$

The problem boundary $\Gamma = \Gamma_0 \cup \Gamma_R$ has been separated into Γ_0, the part of the boundary with incoming source or a vacuum conditions, and the remaining part, Γ_R, the specular reflection part.

We first discuss the boundary conditions on Γ_0, (5a) (see also [16]). Because the problem is linear, we first initialize a source b by solving the problem

$$\boldsymbol{L}b = q, \quad \boldsymbol{B}_0 b = g, \tag{6}$$

for b, whose physical interpretation is the uncollided flux. Then the iteration (4) can be computed according to

$$\psi^{\ell+1} = \boldsymbol{L}^{-1}\boldsymbol{M}\boldsymbol{S}\phi^{\ell} + b, \tag{7a}$$

$$\phi^{\ell+1} = \boldsymbol{D}\psi^{\ell+1}, \tag{7b}$$

Once the source b is calculated, homogeneous, or vacuum, boundary conditions are applied during the iterations.

Reflection is implemented by modifying the operator $\boldsymbol{L}$. We treat reflective conditions implicitly, so no special angular S_N sweep ordering is necessary. This greatly simplifies the implementation with no significant extra computations since the solution is calculated iteratively anyway. We split the operator $\boldsymbol{L}$ into two parts, $\boldsymbol{L} = \boldsymbol{L}_0 + \boldsymbol{L}_R$, where the second part, $\boldsymbol{L}_R$, represents the coupling terms on the reflective boundary faces based on (2e). All other spatial and angular coupling terms remain in the first part, $\boldsymbol{L}_0$. This enables us to "lag" the angular fluxes on the reflective boundary faces and the iteration (7) is modified to be

$$\psi^{\ell+1} = \boldsymbol{L}_0^{-1}\left(\boldsymbol{M}\boldsymbol{S}\phi^{\ell} - \boldsymbol{L}_R\psi^{\ell}\right) + b, \tag{8a}$$

$$\phi^{\ell+1} = \boldsymbol{D}\psi^{\ell+1}. \tag{8b}$$

where b is again computed by solving (6). Let the vector ψ_R consist only of the angular fluxes on the points of reflective boundary faces. The operator $\boldsymbol{P}$ maps ψ_R into the full length vector representation ψ. The vector ψ_R is of length $v = N_R \cdot N$, where N_R is the total number of spatial points on the reflective boundary faces of the problem, so that $\boldsymbol{P}$ is a $(n \times v)$ operator. This enables us to use ϕ as the primary

working vector and save only the vector ψ_R for subsequent iterations. A complete description, then, in operator notation describing precisely how source iteration is implemented in our transport code is

$$\psi^{\ell+1} = \boldsymbol{L}_0^{-1}\left(\boldsymbol{MS}\phi^{\ell} - \boldsymbol{L}_R\boldsymbol{P}\psi_R^{\ell}\right) + b, \tag{9a}$$

$$\phi^{\ell+1} = \boldsymbol{D}\psi^{\ell+1}. \tag{9b}$$

2.3 DSA Methods

We briefly review the DSA method to put it in context. We ignore reflective boundary conditions for purposes of discussion because we can write the iteration for the scalar flux

$$\phi^{\ell+1} = \boldsymbol{TS}\phi^{\ell} + b \tag{10}$$

where $\boldsymbol{T} = \boldsymbol{DL}^{-1}\boldsymbol{M}$ and $b = \boldsymbol{DL}^{-1}q$. If ϕ is the exact solution to (10) then the error $f^{\ell+1} = (\phi - \phi^{\ell+1})$ satisfies

$$(\boldsymbol{I} - \boldsymbol{TS})f^{\ell+1} = \boldsymbol{TS}r^{\ell}, \tag{11}$$

where $r^{\ell} = (\phi^{\ell+1} - \phi^{\ell})$ is the residual and $\boldsymbol{I}$ is the $(t \times t)$ identity operator. Equation (11) suggests that we can use an approximation to the operator $(\boldsymbol{I} - \boldsymbol{TS})^{-1}\boldsymbol{TS}$ to estimate the error and correct the current iterate. This leads to a more efficient iteration if the approximate operator is relatively easy to setup and invert and if the approximate operator adequately reduces the spectral radius.

In the case of DSA, the approximate operator involves the diffusion operator, $\boldsymbol{C}$. This is an appropriate choice because the diffusion equation is the asymptotic limit of the transport operator in highly diffusive regimes [17, 18]. This is just the situation for which we need acceleration. The diffusion operator is effective because it can represent the errors that are poorly attenuated by source iteration and which can be seen to be nearly diffusive [2]. Whether the diffusion operator can be inverted easily and result in a more efficient algorithm depends on the spatial discretization of both the transport equation and the diffusion equation. Introducing an intermediate correction step in the source iteration algorithm, the DSA algorithm is

$$\phi^{\ell+1/2} = \boldsymbol{T}\phi^{\ell} + b \tag{12a}$$

$$f^{\ell+1/2} = \boldsymbol{ES}\left(\phi^{\ell+1/2} - \phi^{\ell}\right) \tag{12b}$$

$$\phi^{\ell+1} = \phi^{\ell+1/2} + f^{\ell+1/2}. \tag{12c}$$

The operator $\boldsymbol{E}$ represents the inverse of the diffusion equation, which we write as $\boldsymbol{C}^{-1}$. It may also include projection or interpolation operators, $\boldsymbol{P}$ and $\boldsymbol{R}$, respectively, that may be required by the particular spatial discretization, or $\boldsymbol{E} = \boldsymbol{RC}^{-1}\boldsymbol{P}$. It is the properties of all the computations represented by the $\boldsymbol{E}$ operator that determines how effective, efficient and robust the overall DSA algorithm will be in practice. Note that additional operations are necessary to treat reflective boundary conditions. For our purposes here, we consider only the partially consistent scheme, the S-WLA method [10, 13]. A method that is fully consistent with our DFEM spatial discretization is simply too costly to be used in general implementations, especially considering the way its effectiveness is degraded in problems with discontinuous material properties.

3 KRYLOV SUBSPACE ITERATIVE METHODS

We now describe the implementation of the Krylov iterative methods, starting from the source iteration that is already implemented in our transport code. The previous work with Krylov iterative methods for transport applications that we find most relevant is briefly reviewed.

3.1 Reformulating Source Iteration

Source iteration is better known mathematically as a stationary, one-step Richardson iteration or as a fixed-point iteration for the scalar flux. It is the simplest possible iteration. We wish to replace source iteration with another, more powerful, iteration, like a Krylov subspace iterative method. To do so, first we write source iteration in a form that we can use with another iterative method.

We postpone discussion of reflective boundary conditions. We start with (7) and assume that we have already calculated the source term b as previously discussed. Eliminating iteration indices and collapsing the iteration into a single expression for the scalar flux moments gives

$$\left(\boldsymbol{I}_t - \boldsymbol{D}\boldsymbol{L}^{-1}\boldsymbol{M}\boldsymbol{S}\right)\phi = \boldsymbol{D}b, \tag{13}$$

where $\boldsymbol{I}_k$ is a $(k \times k)$ identity operator.

We now account for reflective boundary conditions, working with the operators $\boldsymbol{L}_O^{-1}$ and $\boldsymbol{L}_R$ to minimize additions and changes to a code for which source iteration has already been implemented and verified. So, we augment the working vector ϕ with the vector of angular fluxes on the boundary, ψ_R:

$$\widetilde{\phi} = \begin{bmatrix} \phi \\ \psi_R \end{bmatrix}. \tag{14}$$

The source term is similarly augmented, $\widetilde{b} = [\boldsymbol{D}b \;\; 0]^T$. We again collapse the iteration in (8) into a single expression by writing it in the augmented form

$$\widetilde{\phi}^{\ell+1} = \begin{bmatrix} \boldsymbol{D} & 0 \\ 0 & \boldsymbol{I}_v \end{bmatrix} \begin{bmatrix} \boldsymbol{I}_n \\ \boldsymbol{P}^T \end{bmatrix} \boldsymbol{L}_0^{-1} \left[\boldsymbol{I}_n \;\; (-\boldsymbol{L}_R\boldsymbol{P})\right] \begin{bmatrix} \boldsymbol{M} & 0 \\ 0 & \boldsymbol{I}_v \end{bmatrix} \begin{bmatrix} \boldsymbol{S} & 0 \\ 0 & \boldsymbol{I}_v \end{bmatrix} \widetilde{\phi}^{\ell} + \widetilde{b}. \tag{15}$$

We can eliminate the iteration index and bring things from the right to the left hand side to find an expression in the same form as (13):

$$\left(\widetilde{\boldsymbol{I}} - \widetilde{\boldsymbol{D}}\widetilde{\boldsymbol{L}}^{-1}\widetilde{\boldsymbol{M}}\widetilde{\boldsymbol{S}}\right)\widetilde{\phi} = \widetilde{b} \tag{16a}$$

where

$$\widetilde{\boldsymbol{L}}^{-1} = \begin{bmatrix} \boldsymbol{I}_n \\ \boldsymbol{P}^T \end{bmatrix} \boldsymbol{L}_0^{-1} \left[\boldsymbol{I}_n \;\; (-\boldsymbol{L}_R\boldsymbol{P})\right], \quad \text{and} \quad \widetilde{\boldsymbol{I}} = \begin{bmatrix} \boldsymbol{I}_t & 0 \\ 0 & \boldsymbol{I}_v \end{bmatrix}. \tag{16b}$$

Correspondence of the remaining operators with the previous augmented form can be deduced easily by comparison.

It is as simple as that. Existing coding is used to compute the action of the $\boldsymbol{D}$, $\boldsymbol{M}$, $\boldsymbol{S}$ and $\boldsymbol{L}_0^{-1}$ operators at every Krylov iteration. Note that a more powerful iterative method has to utilize, or at least make better use of, information that is available from one or more previous iterates. Some additional storage and computational overhead is needed to implement an advanced iterative strategy. The expectation is that any extra costs are outweighed by an improvement in the iterative convergence rate.

3.2 Preconditioning with DSA

We show that the DSA algorithm is equivalent to a preconditioning the transport operator as follows. See [19] for a related discussion. First, consider that Richardson iteration for some linear system $\boldsymbol{A}x = y$ is simply

$$x^{\ell+1} = x^{\ell} + r^{\ell} = x^{\ell} + \left(y - \boldsymbol{A}x^{\ell}\right) = \left(\boldsymbol{I} - \boldsymbol{A}\right) x^{\ell} + y. \tag{17}$$

Comparing this with (10), we find the operator corresponding to source iteration is $\boldsymbol{A} = (\boldsymbol{I} - \boldsymbol{T}\boldsymbol{S})$. Richardson iteration for the (left) preconditioned linear system $\boldsymbol{M}^{-1}\boldsymbol{A}x = \boldsymbol{M}^{-1}y$ is

$$x^{\ell+1} = x^{\ell} + \boldsymbol{M}^{-1}r^{\ell} = x^{\ell} + \boldsymbol{M}^{-1}\left(y - \boldsymbol{A}x^{\ell}\right) = \left(\boldsymbol{I} - \boldsymbol{M}^{-1}\boldsymbol{A}\right) x^{\ell} + \boldsymbol{M}^{-1}y. \tag{18}$$

Recall that preconditioning is effective if $\boldsymbol{M}^{-1}$ is in some sense an approximation to $\boldsymbol{A}^{-1}$. The preconditioner may be computed explicitly in advance or it may involve the solution of another linear system, $\boldsymbol{M}w = z$, for example, which might need to be computed iteratively or approximately at every iteration. The overall solution can be computed more efficiently only if the preconditioning system can be computed relatively easily.

Collapsing the DSA algorithm (12) into a single operation gives

$$\begin{aligned} \phi^{\ell+1} &= \boldsymbol{T}\boldsymbol{S}\phi^{\ell} + b + \boldsymbol{E}^{-1}\boldsymbol{S}\left(\boldsymbol{T}\boldsymbol{S}\phi^{\ell} + b - \phi^{\ell}\right) \\ &= \left[\boldsymbol{I} + \left(\boldsymbol{I} + \boldsymbol{E}^{-1}\boldsymbol{S}\right)\left(\boldsymbol{T}\boldsymbol{S} - \boldsymbol{I}\right)\right]\phi^{\ell} + \left(\boldsymbol{I} + \boldsymbol{E}^{-1}\boldsymbol{S}\right) b \end{aligned} \tag{19}$$

which, by comparison with (18) shows that the DSA algorithm is just Richardson iteration for the preconditioned system

$$\left(\boldsymbol{I} + \boldsymbol{E}^{-1}\boldsymbol{S}\right)\left(\boldsymbol{I} - \boldsymbol{T}\boldsymbol{S}\right)\phi = \left(\boldsymbol{I} + \boldsymbol{E}^{-1}\boldsymbol{S}\right) b. \tag{20}$$

We can see that $\left(\boldsymbol{I} + \boldsymbol{E}^{-1}\boldsymbol{S}\right)$ is in fact an approximation to the inverse of $(\boldsymbol{I} - \boldsymbol{T}\boldsymbol{S})$ as follows. The computation of the error estimate in (12), $f^{\ell+1/2} = \boldsymbol{E}^{-1}\boldsymbol{S}\left(\phi^{\ell+1/2} - \phi^{\ell}\right)$, is an approximation to the error equation

$$(\boldsymbol{I} - \boldsymbol{T}\boldsymbol{S}) f^{\ell+1/2} = \boldsymbol{T}\boldsymbol{S}r^{\ell}, \tag{21}$$

where $f^{\ell+1/2} = (\phi - \phi^{\ell+1/2})$, $r^{\ell} = (\phi^{\ell+1/2} - \phi^{\ell})$, and ϕ is the exact solution to (10). Examining (12) it is evident that

$$\boldsymbol{E}^{-1}\boldsymbol{S} \approx \left(\boldsymbol{I} - \boldsymbol{T}\boldsymbol{S}\right)^{-1}\boldsymbol{T}\boldsymbol{S} = \left(\boldsymbol{I} - \boldsymbol{T}\boldsymbol{S}\right)^{-1} - \boldsymbol{I}, \tag{22}$$

or, in other words,

$$\left(\boldsymbol{I} + \boldsymbol{E}^{-1}\boldsymbol{S}\right) \approx \left(\boldsymbol{I} - \boldsymbol{T}\boldsymbol{S}\right)^{-1}. \tag{23}$$

3.3 Previous Work

There has been a fair amount of work during the past decade in applying Krylov methods to transport problems in various contexts (see [20–29]. For a definition and description of Krylov subspace and other iterative methods, see [9].

The ideas most influential on the work we are presenting here those of Ashby, et al., [6, 16], Brown [7], and Guthrie, et al. [8]. These papers show similar numerical results and make similar conclusions regarding the effectiveness and efficiency of GMRES preconditioned with DSA. Ashby, et al.[16], and Brown [7], put

emphasis on the analysis of DSA as a preconditioner and the linear algebraic formalism that facilitates their analysis. Guthrie, et al. [8] focused primarily on how to implement the iterative solution in terms of transport sweeps and give an enlightening discussion on the optimality of GMRES.

In [6], the authors considered an inconsistent DSA scheme for linear discontinuous discretizations in one dimension, whose effectiveness degraded in the presence of two very different materials, unlike the consistent method to which it was compared [6]. While this DSA method was totally ineffective in accelerating source iteration, convergence improved significantly when it was applied as a preconditioner to GMRES. Their work actually presaged the situation in which we now find ourselves. They state in their conclusion, "These results have possible implications for problems in higher spatial dimensions, for which a consistent preconditioner is difficult to obtain and/or impractical to apply" [6]. Furthermore, Brown points out in his work on DSA for diamond-difference methods in 3D orthogonal grids that if the diffusion equation in the DSA algorithm is not solved exactly or some other approximations are made, "…the use of these methods in three dimensions will be crucial to the overall usefulness of DSA in 3-D problems," referring to "…more powerful iterative methods such as Bi-CGSTAB …" [7]. Indeed, this is the case for our linear DFEM S_N transport discretization. The FCDSA algorithm is prohibitively expensive to employ and only the partially consistent S-WLA DSA method remains feasible for general purpose. This is because when used as a preconditioner to a Krylov iterative method, it remains effective even in the presence of material discontinuities. The overall solution can be computed efficiently because the S-WLA method is inexpensive, precisely because of the approximations made that make it only partially consistent.

3.4 Implementation and Convergence of Krylov Subspace Iterative Methods

We now make some brief remarks on the choice of Krylov iterative method, convergence of the Krylov methods, and the implementation of the Krylov method in the transport code AttilaV2, indicating how the structure and properties of the transport operator $\boldsymbol{I} - \boldsymbol{TS}$ influences these issues.

3.4.1 Choosing a Krylov iterative method.

The obvious iterative method to choose when an operator is s.p.d. (symmetric, positive definite) is the method of conjugate gradients (CG). Symmetrization of the scalar flux formulation is possible, in the sense that it is possible to define an inner product for which the operator is symmetric [24, 28]. This is all that is needed for isotropic scattering. The operator in (13) or (16) is a discrete form of integral equation for the vector of scalar flux moments. By examining the continuous form of the integral equation for anisotropic scattering, Santandrea and Sanchez very cleverly define an inner product *and* preconditioner that makes the preconditioned operator s.p.d. [24].

This implies that we might use CG as an iterative solution technique, provided the spatial discretization does not induce a nonsymmetric operator. We found that our LDFEM discretization on tetrahedra does in fact makes the scalar flux formulation nonsymmetric, even for isotropic scattering with constant material properties. Furthermore, two potential DSA schemes that are compatible with that discretization, the fully consistent, discontinuous diffusion discretization, and the partially consistent, S-WLA method, are either nonsymmetric, or symmetric and indefinite. So, we cannot guarantee that (preconditioned) CG will converge under these circumstances and we have use a nonsymmetric Krylov iteration like GMRES.

3.4.2 A few words about convergence.

To limit memory use, we use the restarted version of GMRES, GMRES(m), which restarts the iteration every m iterations. We have found that the transport operator, while nonsymmetric, is positive definite, in which case restarted GMRES is guaranteed to converge [9, 27].

We have also found that our operator is non-normal as well as being nonsymmetric. This can complicate the prediction and analysis of convergence [27, 30]. The convergence behavior of simple iterative methods like Richardson iteration or Krylov iterative methods for symmetric systems is well-understood in terms of the eigenvalues of the linear system. In contrast, there is very little that can be done to accurately predict or estimate the convergence of nonsymmetric Krylov iterative methods, except under special circumstances. Even if available, the estimated bounds may not be sharp. We do not consider this lack of a priori knowledge about convergence to be restrictive because our experience so far indicates that the nonsymmetric Krylov methods GMRES and BiCGStab, for example, are very robust for this application.

We can, however, make a few statements about convergence. In exact arithmetic, GMRES will converge in a number of iterations less than or equal to the degree of the minimal polynomial of $\boldsymbol{A}$, denoted $d(\boldsymbol{A})$, except for some very special cases. The minimal polynomial is the unique monic polynomial that satisfies $p_d(\boldsymbol{A}) = \mathbf{0}$ of lowest degree. It can be seen that such a polynomial will, generally speaking, have low degree if the eigenvalues of $\boldsymbol{A}$ are non-defective and of high multiplicity. The Krylov subspace from which the solution at iteration p, x_p, is chosen is $K_m(\boldsymbol{A}, r_0) = \text{span}\{r_0, \boldsymbol{A}r_0, \boldsymbol{A}^2 r_0, \ldots \boldsymbol{A}^{m-1} r_0\}$. Obviously, the maximum dimension that the Krylov subspace can attain is the degree of the minimal polynomial. When $m = d(\boldsymbol{A})$ the Arnoldi process, which computes an orthogonal basis for $K_m(\boldsymbol{A}, r_0)$ halts and GMRES suffers what is called a "lucky breakdown", "lucky" because the solution has converged at that point [9]. In finite precision arithmetic, however, the eigenvalues are perturbed such that they form small clusters of distinct eigenvalues about the exact values and the degree of the minimal polynomial will be larger than it would be otherwise. Nonetheless, GMRES still has the opportunity to converge quickly if the eigenvalues make up a small number of small clusters. This is often observed in practice [30].

Now, if the matrix $\boldsymbol{A}$ is normal ($\boldsymbol{A}$ has a complete orthonormal eigendecomposition $\boldsymbol{A} = \boldsymbol{Q}^* \boldsymbol{\Lambda} \boldsymbol{Q}$ or, equivalently, $\boldsymbol{A}\boldsymbol{A}^T = \boldsymbol{A}^T \boldsymbol{A}$) then its spectrum will be insensitive to perturbations, or well-conditioned, and the effects of finite precision arithmetic should be nominal. In that case, GMRES should converge in approximately $d(\boldsymbol{A})$ iterations. Non-normality does not imply that the spectrum of $\boldsymbol{A}$ is ill-conditioned, although an operator whose spectrum is ill-conditioned has to be non-normal. If GMRES is applied to a non-normal matrix and it converges in the number of iterations predicted by $d(\boldsymbol{A})$, then everything is as it should be, and the spectrum is probably well-conditioned. However, when GMRES does *not* converge in the predicted number of iterations when the matrix is non-normal, then it might be that the spectrum is ill-conditioned.

We have attempted to explore the properties of the transport operator by explicitly constructing $\boldsymbol{I} - \boldsymbol{T}\boldsymbol{S}$ for some simple, homogeneous problems, including anisotropic scattering, on a few very small meshes. Having found the operator is non-normal, GMRES still converged within one or two iterations of the degree of the minimal polynomial, which was typically quite small, on the order of 10-15. This indicates that the eigenvalues are well-conditioned and GMRES has the opportunity to converge quickly.

A number of recent texts provide further information on the theoretical background behind Krylov iterative

methods; see [9, 27], for example.

3.4.3 Applying DSA as a preconditioner.

The preconditioned linear system, in the form given in (20), is what we solve with a Krylov iterative method. Assume for the moment that we are solving the preconditioned linear system $\boldsymbol{M}^{-1}\boldsymbol{A}x = \boldsymbol{M}^{-1}y$. At every iteration, the Krylov method supplies a vector v to which the linear system is applied, that is, the vector $z = \boldsymbol{A}v$ is computed and returned to the Krylov solver. Subsequently, the linear system $\boldsymbol{M}w = z$ is "solved" and the vector w is returned to the solver. This allows us to compute the "action" of the preconditioner on the vector z without the inverse matrix $\boldsymbol{M}^{-1}$ being available.

However, we do not actually use the preconditioned version of the Krylov algorithm where we would have to first apply the linear system to v, $z = (\boldsymbol{I} - \boldsymbol{T}\boldsymbol{S})v$, and subsequently compute the action of the preconditioner on z, $w = (\boldsymbol{I} + \boldsymbol{E}^{-1}\boldsymbol{S})z$, as we would if DSA were being used as a preconditioner. Instead, we use the unpreconditioned version of the Krylov method. At every iteration we set $\phi^{\ell} = v$, compute the sequence of operations shown in the DSA algorithm, Eqs. 12, with the original source iteration code. We then return $w = v - \phi^{\ell+1}$ to the Krylov solver. Collapsing the solution process into a single operation shows that this approach is fully equivalent to preconditioning in the usual sense but requires much less code modification because the Krylov solver can just be "wrapped around" the source iteration code with just the addition of the last step $w = v - \phi^{\ell+1}$.

3.4.4 Inner-outer iterations.

The S-WLA DSA equations involve solving an s.p.d. linear system (the linear continuous finite element discretization of the diffusion equation) with conjugate gradients (CG). We would like to save computational effort in the combined inner-outer iteration by varying the tolerance of the inner iteration, which in this case is the CG iteration, without affecting the accuracy or convergence of the outer iteration, which in this case is a Krylov subspace method. Recently, it was observed in [31] that the inner iterations need to be computed to a strict convergence tolerance in the early part of the outer Krylov iterations. The tolerance can then be relaxed as the outer iteration proceeds. We set the inner tolerance in inverse proportion to the norm of the outer residual vector in accordance with [31]. This approach has recently been understood and justified theoretically [32]. If restarting, however, the inner solution should again be computed with high precision and the tolerance subsequently relaxed. The inner tolerance for the DSA CG solution at outer iteration k, with residual r_k, is set according to

$$\gamma = \begin{cases} \frac{1}{10}\epsilon & \text{if } k \bmod m = 0 \\ \frac{1}{10}\max(\epsilon, \min(1, \epsilon/\min(\|r_k\|_2, 1))) & \text{otherwise} \end{cases}$$

where ϵ is the tolerance for the outer iteration. We have found the factor of $1/10$ to be conservative in that this choice did not affect convergence or accuracy in any of the problems we have tried.

4 NUMERICAL RESULTS

In this section we compare the Krylov iterations to source (Richardson) iterations, with and without preconditioning using the partially consistent S-WLA DSA method. The results are computed using

AttilaV2 [33] for a realistic, two-material "duct" problem. They illustrate both how computational effort depends on scattering ratio and the total cross section in the two regions in the problem.

We found through experimentation on a wide variety of problems that the reasonably-sized restart parameter of $m = 10$ did not reduce the outer convergence rate compared to higher values. There was some slight sensitivity for values smaller than this.

We refer to traditional source iteration as "SI", and to source iteration accelerated by the S-WLA DSA method as "ASI".

We refer to the Krylov method without preconditioning as, "GMRES", and to the Krylov method preconditioned with the S-WLA algorithm as, "PGMRES". In both cases, GMRES(10) is the Krylov method used.

The following stopping criterion is used in all results presented here:

$$\frac{\|\phi^{\ell+1} - \phi^{\ell}\|_2}{\|\phi^{\ell+1}\|_2} \leq 10^{-5}, \tag{24}$$

or 2000 iterations. The CG convergence tolerance for the DSA method is fixed to 10^{-6} only for the source iteration solutions. The strategy discussed in Sec. 3.4 is used for the PGMRES results. Initial scalar fluxes are zero and we use a triangular, S_4 Chebyshev-Legendre quadrature.

The problem consists of a reflected quarter-cylinder, 25 cm in radius and 50 cm long. The "thin" duct region is 5 cm in radius. It is surrounded by a "thick" region and bends around a central disc of thick material. There is a unit isotropic boundary source incident on the left face of the duct. Vacuum boundary conditions are specified the outer surfaces. The values of the total cross sections in the thin region, $\sigma_{t,1}$, and the thick region, $\sigma_{t,2}$, as well as the scattering ratio c, are varied to examine the effect of material heterogeneities in a realistic problem. An isotropic source of strength 1.0^{-6} particles$/$cm^3 is distributed throughout the problem to help smooth the solution. The unstructured, tetrahedral mesh is illustrated in Figs. 1. It consists of 31,481 cells.

The number of iterations and the measured number of floating point operations (FLOP) on a single SGI Origin 2000 250 MHz CPU are tabulated below. The results for ASI and PGMRES on the Tet Mesh are shown in Tables I and II. These results may be compared to the results for SI and GMRES (no DSA acceleration or preconditioning) on the Tet Mesh shown in Tables III and IV. Each table contains data for both solution methods for various values of thin ($\sigma_{t,1}$) and thick ($\sigma_{t,2}$) region cross sections and scattering ratios $c = 1.0000, 0.9999, 0.999, 0.99, 0.9$.

The most apparent and important observation is that PGMRES significantly reduces the number of iterations needed for convergence compared to ASI. This improvement in iteration count offsets any extra computational effort. The savings in compuational effort compares more favorably as the scattering ratio c approaches 1.0 and S-WLA becomes less effective. While PGMRES needs less computational effort even for the lowest value of $c = 0.9$ considered here, there is most likely going to be a point at which c is small enough that ASI, will compute a solution with less effort because of its lower per iteration cost. However, as c approaches zero DSA is not really needed anyway and unaccelerated source iteration is perfectly adequate. It is notable that the results that GMRES alone, without the S-WLA preconditioner, does in fact

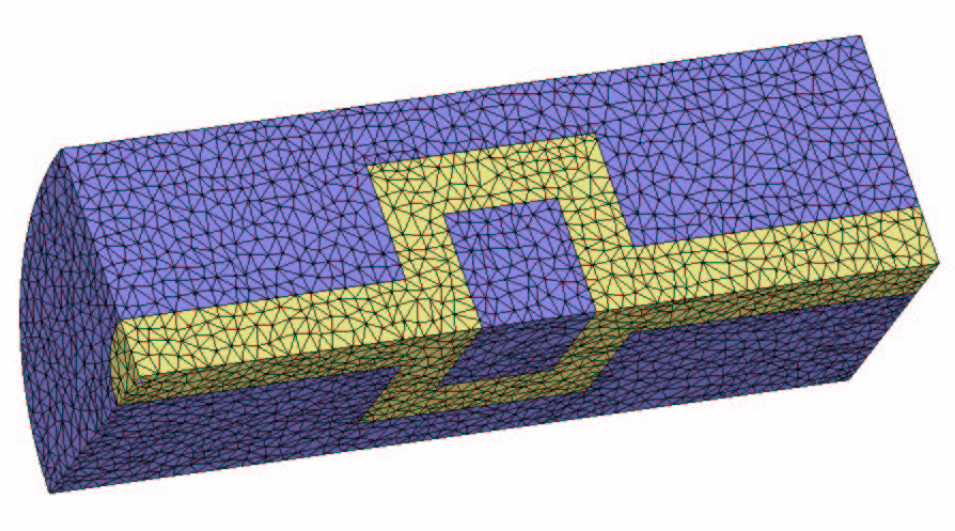

Figure 1. The unstructured, tetrahedral mesh used for the numerical computations. The total cross section in the lightly-shaded, "thin" duct region is $\sigma_{t,2}$ and $\sigma_{t,1}$ in the darker, "thick" surrounding region.

accelerate the transport iterations. In some cases it performs as well as DSA-accelerated source iteration, although we cannot conclude that this will be true in general. The results reported here are encouraging, especially since PGMRES outperforms source iteration when material discontinuities are not present.

Finally, as shown in Table V the additional memory required for GMRES(10) is not unreasonable, only about 50% greater than what is needed for the source iteration implementation.

5 CONCLUSIONS

A Krylov subspace iterative method can significantly improve the efficiency of S_N transport calculations in problems for which source iteration accelerated with DSA schemes will be ineffective. Such problems have discontinuities in material properties that render even fully consistent methods ineffective. Most importantly, our results show that a partially consistent DSA method like the S-WLA method can be used as an efficient preconditioner for a Krylov iterative method, despite being ineffective as an acceleration scheme for traditional source iteration. The rate of convergence is substantially faster when the Krylov iteration is preconditioned with the S-WLA DSA method, relative to either source iteration accelerated with S-WLA, or GMRES without preconditioning.

Computational measurements indicate that the extra computational and storage overhead associated with the GMRES Krylov iterative method is acceptable given that these problems may not even converge in a reasonable amount of time using accelerated source iteration.

Although we only considered steady-state, one-group problems with isotropic scattering, we successfully tested this approach on problems with anisotropic scattering and on multigroup, criticality eigenvalue problems. We found that preconditioned Krylov methods outperformed accelerated source iteration in almost all the problems we tested. We feel that Krylov iterative methods are well-suited for calculating within-group solutions in transport codes that are intended for general purpose use.

Table I. Computational results for the Tet Mesh. Number of iterations are tabulated for total cross sections $\sigma_{t,1}$ and $\sigma_{t,2}$ (cm^{-1}) and a range of scattering ratio c.

		$\sigma_{t,2}$							
		10^3		10^2		10^1		10^0	
$\sigma_{t,1}$	c	PGMRES	ASI	PGMRES	ASI	PGMRES	ASI	PGMRES	ASI
10^{-3}	1.0000	45	618	36	403	14	71	5	14
	0.9999	43	513	34	378	14	71	5	14
	0.999	25	222	30	291	14	69	5	14
	0.99	10	41	19	136	13	56	5	14
	0.9	4	9	8	28	8	28	5	12
10^{-2}	1.0000	36	379	29	267	13	59	5	13
	0.9999	31	305	28	251	13	58	5	13
	0.999	19	121	24	193	13	57	5	12
	0.99	9	34	16	85	12	46	5	12
	0.9	4	8	7	23	8	23	5	11
10^{-1}	1.0000	19	112	17	86	9	27	4	9
	0.9999	19	106	17	81	9	27	4	9
	0.999	17	83	15	67	9	26	4	9
	0.99	9	29	13	53	8	23	4	9
	0.9	4	8	7	19	6	17	4	8
10^{0}	1.0000	14	58	11	37	6	14	4	8
	0.9999	15	66	11	38	6	14	4	8
	0.999	14	59	12	39	6	14	4	8
	0.99	9	25	10	34	6	14	4	8
	0.9	5	10	6	16	5	12	4	7
10^{1}	1.0000	11	30	9	26	6	13		
	0.9999	10	36	9	27	6	13		
	0.999	10	37	9	30	6	13		
	0.99	8	24	9	27	6	13		
	0.9	6	15	7	16	6	12		
10^{2}	1.0000	11	35	11	34				
	0.9999	13	43	11	34				
	0.999	13	48	11	34				
	0.99	11	37	10	31				
	0.9	7	18	7	18				
10^{3}	1.0000	21	49						
	0.9999	13	50						
	0.999	12	44						
	0.99	9	28						
	0.9	4	10						

Table II. Computational results for the Tet Mesh. FLOP counts (in billions) are tabulated for total cross sections $\sigma_{t,1}$ and $\sigma_{t,2}$ (cm^{-1}) and a range of scattering ratio c.

		$\sigma_{t,2}$							
		10^3		10^2		10^1		10^0	
$\sigma_{t,1}$	c	PGMRES	ASI	PGMRES	ASI	PGMRES	ASI	PGMRES	ASI
10^{-3}	1.0000	8.17	90.05	6.54	58.79	2.94	10.43	1.30	2.13
	0.9999	7.86	74.31	6.24	55.11	2.94	10.43	1.30	2.13
	0.999	4.75	32.06	5.48	42.18	2.94	10.15	1.30	2.13
	0.99	2.14	5.92	3.60	19.66	2.75	8.23	1.30	2.13
	0.9	1.18	1.35	1.74	4.07	1.84	4.12	1.29	1.84
10^{-2}	1.0000	6.46	54.70	5.27	38.51	2.66	8.57	1.28	1.96
	0.9999	5.69	43.79	5.11	36.12	2.66	8.43	1.28	1.96
	0.999	3.57	17.33	4.47	27.71	2.65	8.28	1.28	1.81
	0.99	1.87	4.87	3.09	12.19	2.51	6.68	1.27	1.81
	0.9	1.05	1.17	1.56	3.32	1.73	3.35	1.27	1.66
10^{-1}	1.0000	3.62	16.01	3.23	12.29	1.95	3.89	1.07	1.32
	0.9999	3.61	15.10	3.22	11.58	1.95	3.89	1.07	1.32
	0.999	3.27	11.73	2.90	9.56	1.95	3.75	1.07	1.32
	0.99	1.86	3.96	2.59	7.51	1.79	3.32	1.07	1.32
	0.9	1.05	1.06	1.48	2.63	1.46	2.44	1.07	1.18
10^{0}	1.0000	2.70	8.15	2.25	5.19	1.34	1.96	1.02	1.12
	0.9999	2.79	9.11	2.26	5.31	1.34	1.96	1.02	1.12
	0.999	2.59	7.82	2.37	5.35	1.33	1.96	1.02	1.12
	0.99	1.64	3.09	1.89	4.54	1.32	1.94	1.02	1.12
	0.9	1.06	1.25	1.23	2.00	1.14	1.61	1.01	0.99
10^{1}	1.0000	2.54	4.11	1.72	3.50	1.35	1.73		
	0.9999	1.93	4.70	1.70	3.60	1.35	1.73		
	0.999	1.85	4.51	1.67	3.87	1.34	1.72		
	0.99	1.53	2.87	1.60	3.28	1.31	1.67		
	0.9	1.25	1.79	1.30	1.91	1.26	1.46		
10^{2}	1.0000	2.24	4.65	2.04	4.34				
	0.9999	2.21	5.14	1.99	4.17				
	0.999	2.17	5.55	1.94	3.97				
	0.99	1.90	4.20	1.68	3.54				
	0.9	1.27	2.07	1.27	2.07				
10^{3}	1.0000	4.48	6.07						
	0.9999	2.22	5.65						
	0.999	2.09	4.94						
	0.99	1.61	3.17						
	0.9	0.98	1.18						

Table III. Computational results for the Tet Mesh *without* S–WLA preconditioning or acceleration. Number of iterations are tabulated for total cross sections $\sigma_{t,1}$ and $\sigma_{t,2}$ (cm^{-1}) and a range of scattering ratio c. An entry "n/c" indicates that the problem did not converge in 2000 iterations.

		$\sigma_{t,2}$							
		10^3		10^2		10^1		10^0	
$\sigma_{t,1}$	c	GMRES	SI	GMRES	SI	GMRES	SI	GMRES	SI
10^{-3}	1.0000	112	n/c	350	n/c	43	606	8	28
	0.9999	88	n/c	316	n/c	42	601	8	28
	0.999	33	n/c	85	n/c	41	553	8	28
	0.99	12	486	30	441	28	310	8	27
	0.9	4	59	10	67	12	60	7	22
10^{-2}	1.0000	325	n/c	319	n/c	46	607	8	28
	0.9999	108	n/c	213	n/c	46	601	8	28
	0.999	42	n/c	92	n/c	44	553	8	28
	0.99	13	486	33	441	29	310	8	27
	0.9	5	59	10	67	13	60	7	22
10^{-1}	1.0000	156	n/c	322	n/c	45	609	8	28
	0.9999	93	n/c	269	n/c	44	603	8	28
	0.999	50	n/c	116	n/c	42	555	8	28
	0.99	18	485	35	441	29	311	8	27
	0.9	6	59	12	67	13	60	7	22
10^{0}	1.0000	1133	n/c	393	n/c	50	635	10	35
	0.9999	320	n/c	291	n/c	50	628	10	35
	0.999	73	n/c	128	n/c	47	575	10	35
	0.99	26	477	39	441	33	313	9	34
	0.9	9	59	14	66	14	60	8	25
10^{1}	1.0000	920	n/c	660	n/c	57	850		
	0.9999	401	n/c	392	n/c	57	835		
	0.999	99	n/c	170	n/c	52	720		
	0.99	37	439	45	439	33	319		
	0.9	14	57	16	65	14	61		
10^{2}	1.0000	n/c	n/c	769	n/c				
	0.9999	549	n/c	363	n/c				
	0.999	136	n/c	131	n/c				
	0.99	37	441	37	452				
	0.9	12	68	11	68				
10^{3}	1.0000	146	n/c						
	0.9999	49	n/c						
	0.999	25	1776						
	0.99	11	511						
	0.9	5	81						

Table IV. Computational results for the Tet Mesh *without* S–WLA preconditioning or acceleration. FLOP counts (in billions) are tabulated for total cross sections $\sigma_{t,1}$ and $\sigma_{t,2}$ (cm^{-1}) and a range of scattering ratio c. An entry "n/c" indicates that the problem did not converge in 2000 iterations.

		$\sigma_{t,2}$							
		10^3		10^2		10^1		10^0	
$\sigma_{t,1}$	c	GMRES	SI	GMRES	SI	GMRES	SI	GMRES	SI
10^{-3}	1.0000	14.73	n/c	44.43	n/c	5.94	63.25	1.32	3.00
	0.9999	11.60	n/c	40.14	n/c	5.83	62.72	1.32	3.00
	0.999	4.65	n/c	11.04	n/c	5.71	57.72	1.32	3.00
	0.99	1.98	50.74	4.09	46.05	3.98	32.39	1.32	2.90
	0.9	0.93	6.23	1.57	7.07	1.99	6.34	1.20	2.38
10^{-2}	1.0000	41.20	n/c	40.33	n/c	6.15	63.35	1.32	3.00
	0.9999	13.90	n/c	27.09	n/c	6.15	62.72	1.32	3.00
	0.999	5.66	n/c	11.93	n/c	5.91	57.72	1.32	3.00
	0.99	1.99	50.74	4.50	46.05	3.98	32.39	1.32	2.90
	0.9	0.96	6.23	1.56	7.07	2.01	6.34	1.20	2.38
10^{-1}	1.0000	20.21	n/c	40.75	n/c	6.18	63.56	1.32	3.00
	0.9999	12.27	n/c	34.04	n/c	6.06	62.93	1.32	3.00
	0.999	6.77	n/c	14.87	n/c	5.83	57.93	1.32	3.00
	0.99	2.69	50.63	4.72	46.05	4.10	32.50	1.32	2.90
	0.9	1.17	6.23	1.89	7.07	2.11	6.34	1.20	2.38
10^{0}	1.0000	142.68	n/c	49.88	n/c	6.63	66.27	1.57	3.73
	0.9999	40.51	n/c	37.05	n/c	6.63	65.54	1.57	3.73
	0.999	9.54	n/c	16.41	n/c	6.27	60.01	1.57	3.73
	0.99	3.59	49.80	5.22	46.05	4.54	32.71	1.45	3.63
	0.9	1.43	6.23	2.11	6.96	2.12	6.34	1.32	2.69
10^{1}	1.0000	118.08	n/c	83.66	n/c	7.70	88.68		
	0.9999	51.70	n/c	49.88	n/c	7.70	87.11		
	0.999	13.05	n/c	21.78	n/c	7.11	75.13		
	0.99	5.13	45.84	6.03	45.84	4.66	33.33		
	0.9	2.22	6.02	2.35	6.86	2.22	6.44		
10^{2}	1.0000	252.58	n/c	97.31	n/c				
	0.9999	69.78	n/c	46.15	n/c				
	0.999	17.47	n/c	16.93	n/c				
	0.99	4.99	46.05	5.00	47.19				
	0.9	1.89	7.17	1.78	7.17				
10^{3}	1.0000	19.14	n/c						
	0.9999	6.68	n/c						
	0.999	3.61	185.19						
	0.99	1.87	53.34						
	0.9	1.05	8.53						

Table V. Approximate memory requirements (MB) measured on a dedicated SGI Origin 2000 single processor.

PGMRES	GMRES	ASI	SI
276	246	181	178

Our conclusions could change if we consider a parallel implementation. However, as long as the S_N sweep algorithms can be implemented efficiently then there should be no reason why we would not see similar performance in parallel applications.

Acknowledgments

This work was performed under the auspices of the U.S. Department of Energy at the Los Alamos National Laboratory. The authors would like to thanks Marvin Adams, Yousry Azmy, Ed Larsen, James Holloway, John McGhee, and Bruce Patton for their very helpful discussions. And special thanks go to Michele Benzi for his careful review of Sec. 3 and for working with us on this topic.

REFERENCES

[1] R. E. Alcouffe, "Diffusion Synthetic Acceleration Methods for Diamond–Differenced Discrete–Ordinates Equations," *Nucl. Sci. and Engr.*, **64**, pp. 344–355 (1977).

[2] E. W. Larsen, "Unconditionally Stable Diffusion–Synthetic Acceleration Methods for Slab Geometry Discrete Ordinates Equations. Part I: Theory," *Nucl. Sci. and Engr.*, **82**, pp. 47–63 (1982).

[3] D. R. McCoy and E. W. Larsen, "Unconditionally Stable Diffusion–Synthetic Acceleration Methods for Slab Geometry Discrete Ordinates Equations. Part II: Numerical Results," *Nucl. Sci. and Engr.*, **82**, pp. 64–70 (1982).

[4] Y. Azmy, "Impossibility of Unconditional Stability and Robustness of Diffusive Accleration Schemes," in **1998 American Nuclear Society Radiation Protection and Shielding Division Topical Meeting**, 19–23 Apr, Vol. 1, Nashville, TN, p. 480 (1998).

[5] Y. Azmy, T. Wareing, and J. Morel, "Effect of Material Heterogeneity on the Performance of DSA for Even–Parity S_N Methods," in **International Conference on Mathematics and Computation, Reactor Physics, and Environmental Analysis in Nuclear Applications**, 27–30 Sep, Vol. 1, Madrid, Spain, pp. 55–63 (1999).

[6] S. F. Ashby, P. N. Brown, M. R. Dorr, and A. C. Hindmarsh, "Preconditioned Iterative Methods for Discretized Transport Equations," in **Proc. International Topical Meeting on Advances in Mathematics, Computations, Reactor Physics**, 28 April – 2 May, Vol. 2, Pittsburgh, Pennsylvania, pp. 6.1 2–1 (1991). (Also available as LLNL Report UCRL-JC-104901, July 1990.).

[7] P. N. Brown, "A Linear Algebraic Development of Diffusion Synthetic Acceleration for Three–Dimensional Transport Equations," *SIAM Journal on Numerical Analysis*, **32**, pp. 179–214 (1995).

[8] B. Guthrie, J. P. Holloway, and B. W. Patton, "GMRES as a Multi–Step Transport Sweep Accelerator," *Tran. Theory and Stat. Phys.*, **28**, n. 1, pp. 83–102 (1999).

[9] Y. Saad, **Iterative Methods for Sparse Linear Systems**. PWS Publishing Company: Boston (1996).

[10] T. A. Wareing, J. M. McGhee, J. E. Morel, and S. D. Pautz, "Discontinuous Finite Element S_n Methods on Three–Dimensional Unstructured Grids," *Nuclear Science and Engineering*, **138**, pp. 1–13 (2001).

[11] T. A. Wareing, "New Diffusion–Synthetic Accelerations Methods for the S_N Equations with Corner Balance Spatial Differencing," in **Joint International Conference on Mathematical Methods and Supercomputing in Nuclear Applications**, 19–23 April, Vol. 2, Karlsruhe, Germany, p. 500 (1993).

[12] J. S. Warsa, T. A. Wareing, and J. E. Morel, "Fully Consistent Linear Discontinuous Diffusion Synthetic Acceleration 3D Unstructured Meshes," in **Proceedings of the 2001 International Meeting on Mathematical Methods for Nuclear Applications**, 9–13 Sep., Salt Lake City, Utah (2001).

[13] J. S. Warsa, T. A. Wareing, and J. E. Morel, "Fully Consistent Diffusion Synthetic Acceleration of Linear Discontinuous Transport Discretizations on Three–Dimensional Unstructured Meshes," *Nuclear Science and Engineering*, **141**, pp. 236–251 (2002).

[14] E. E. Lewis and W. F. Miller, **Computational Methods of Neutron Transport**. Wiley & Sons: New York (1984).

[15] M. L. Adams, "Discontinuous Finite Element Methods in Thick Diffusive Problems," *Nucl. Sci. and Engr.*, **137**, pp. 298–333 (2001).

[16] S. F. Ashby, P. N. Brown, M. R. Dorr, and A. C. Hindmarsh, "A Linear Algebraic Analysis of Diffusion Synthetic Acceleration for the Boltzmann Transport Equation," *SIAM Journal on Numerical Analysis*, **32**, pp. 128–178 (1995).

[17] E. W. Larsen, J. E. Morel, and W. F. Miller, Jr., "Asymptotic Solutions of Numerical Transport Problems in Optically Thick, Diffusive Regimes I," *J. Comp. Phys.*, **69**, pp. 283–324 (1987).

[18] E. W. Larsen and J. E. Morel, "Asymptotic Solutions of Numerical Transport Problems in Optically Thick, Diffusive Regimes II," *J. Comp. Phys.*, **83**, pp. 212–236 (1989).

[19] M. L. Adams and E. W. Larsen, "Fast Iterative Methods for Discrete-Ordinates Particle Transport Calculations," *Prog. Nucl. Energy*, **40**, pp. 3–159 (2002).

[20] V. Faber and T. A. Manteuffel, "A Look at Transport Theory from the Point of View of Linear Algebra," in **Lecture Notes in Pure and Applied Mathematics** (P. Nelson, et al., Ed.), Vol. 115, pp. 31–61 Marcel Dekker: New York (1989).

[21] C. T. Kelley, "Multilevel Source Iteration Accelerators for the Linear Transport Equation in Slab Geometry," *Trans. Theory and Stat. Phys.*, **24**, n. 4&5, pp. 697–707 (1995).

[22] C. T. Kelley and Z. Q. Xue, "GMRES and Integral Operators," *SIAM J. Scientific Computing*, **17**, n. 1, pp. 217–226 (1996).

[23] S. Oliveira and Y. Deng, "Preconditioned Krylov Subspace Methods for Transport Equations," *Prog. Nucl. Energy*, **33**, n. 1/2, pp. 155–174 (1998).

[24] R. Sanchez and S. Santandrea, "Symmetrization of the Transport Operator and Lanzcos' Iterations," in **Proceedings of the 2001 International Meeting on Mathematical Methods for Nuclear Applications**, 9–13 Sep., Salt Lake City, Utah (2001).

[25] B. W. Patton and J. P. Holloway, "Application of Krylov Subspace Methods to the Slab Geometry Transport Equation," in **ANS Topical Meeting on Radiation Protection and Shielding**, 21–25 April, Vol. 1, Cape Cod, pp. 384–389 (1996).

[26] B. W. Patton and J. P. Holloway, "Application of Preconditioned GMRES to the Numerical Solution of the Neutron Transport Equation," *Ann. Nucl. Energy*, **29**, n. 2, pp. 109–136 (2002).

[27] A. Greenbaum, **Iterative Methods for Solving Linear Systems**. SIAM: Philadelphia (1997).

[28] G. L. Ramone, M. L. Adams, and P. F. Nowak, "A Transport Synthetic Acceleration Method for Transport Iterations," *Nucl. Sci. and Engr.*, **125**, pp. 257–283 (1997).

[29] M. R. Zika and M. L. Adams, "Transport Synthetic Acceleration with Opposing Reflecting Boundary Conditions," *Nucl. Sci. and Engr.*, **134**, pp. 159–170 (2000).

[30] S. L. Campbell, I. C. F. Ipsen, C. T. Kelley, and C. D. Meyer, "GMRES and the Minimal Polynomial," *BIT*, **36**, pp. 664–675 (1996).

[31] A. Bouras and V. Frayssé, "A Relaxation Strategy for Inexact Matrix–Vector Products for Krylov Methods," CERFACS TR/PA/00/15, European Centre for Research and Advanced Training in Scientific Computation, Toulouse, France (2000). Submitted to *SIAM J. Matrix Anal. Appl.*.

[32] V. Simoncini and D. Szyld, "Theory of Inexact Krylov Subspace Methods and Applications to Scientific Computing," Research Report 02–4–12, Temple University, Apr. (2002).

[33] T. A. Wareing, J. M. McGhee, and J. E. Morel, "ATTILA: A Three–Dimensional, Unstructured Tetrahedral Mesh Discrete Ordinates Transport Code," *Transactions of the American Nuclear Society*, **75**, pp. 146–147 (1996).