Lecture 21
Basic Krylov Methods With Application to Transport

1 Introduction

Krylov methods are currently having an enormous impact upon the manner in which the S_N equations are solved. The first applications of Krylov methods to the S_N equations was made by applied mathematicians in the late 80's and early 90's, but the numerical transport community was initially somewhat slow to embrace them. However, over the last several years, Krylov methods have entered the mainstream of S_N solution techniques. Indeed, almost every S_N solution technique developed today can be expected to be based upon a preconditioned Krylov method. There are three basic reasons for the mainstream acceptance of Krylov methods within the numerical transport community. First, the recent recognition that DSA and DSA-like methods are not unconditionally effective in multidimensional problems has motivated a search both for modifications to the DSA method that would eliminate this deficiency and for fundamentally new unconditionally effective solution techniques. Second, it was demonstrated early on that an inconsistently discretized DSA method, when recast as a preconditioner for a Krylov method, produces a (presumably) unconditionally effective and efficient S_N solution technique. Third, it has been demon-

strated that Krylov methods, even with relatively simple preconditioners, can be much more effective than simple source iteration. In this paper, we give a basic description of Krylov methods, and then we consider some of the specific issues that arise when applying preconditioned Krylov methods to the S_N equations. This is not intended as a review of all forms of Krylov methods, but rather as a basic introduction to Krylov concepts with a focus on those types of Krylov methods that appear to be most relevant for transport calculations.

2 The Central Theme of Krylov Methods

Let us assume that we want to iteratively solve the following matrix equation:

$$\mathbf{A}\overrightarrow{x} = \overrightarrow{b} \quad , \tag{1}$$

where $\mathbf{A}$ is a non-singular $n \times n$ matrix, $\overrightarrow{x}$ is a solution vector of length n, and $\overrightarrow{b}$ is a source vector of length N. We will assume an initial solution guess for Eq. (1) of $\overrightarrow{x}_0 = \overrightarrow{0}$. However, if $\overrightarrow{x}_0$ is not zero, one can always solve the following equivalent system with an initial zero guess:

$$\mathbf{A}\overrightarrow{x}' = \overrightarrow{b}' \quad , \tag{2a}$$

where

$$\overrightarrow{x}' = \overrightarrow{x} - \overrightarrow{x}_0 \quad , \tag{2b}$$

and

$$\overrightarrow{b}' = \overrightarrow{b} - \mathbf{A}\overrightarrow{x}_0 \ . \tag{2c}$$

Fundamental to the central theme of a Krylov method is a Krylov space of dimension m, which is defined with respect to the matrix $\mathbf{A}$ and the vector $\overrightarrow{b}$. We denote this space by $K_m(\mathbf{A}, \overrightarrow{b})$. A basis for this space consists of M vectors formed by applying successive powers of $\mathbf{A}$ to $\overrightarrow{b}$:

$$K_m(\mathbf{A}, \overrightarrow{b}) = span\{ \overrightarrow{b}, \mathbf{A}\overrightarrow{b}, \mathbf{A}^2\overrightarrow{b}, \ldots, \mathbf{A}^{m-1}\overrightarrow{b} \} \ . \tag{3}$$

We refer to these basis vectors as the *Krylov vectors*. The basic idea of Krylov methods is to approximate the solution to Eq. (1) by a linear combination of Krylov vectors. Before we give more details on how this is done, we first show that the solution to Eq. (1) lies within a Krylov space. The minimal polynomial of a matrix, denoted by $P_d(\mathbf{A})$, is defined as the monic polynomial of minimum degree for which $\mathbf{A}$ is a root:

$$P_d(\mathbf{A}) = c_0 + c_1\mathbf{A}^1 + \ldots + c_d\mathbf{A}^d = \mathbf{0} \ . \tag{4}$$

If $\mathbf{A}$ is nonsingular then zero is not a root of its minimal polynomial, which implies that $c_0 \neq 1$. Thus, the minimal polynomial can be scaled so that $a_0 = 1$, yielding:

$$P_d^*(\mathbf{A}) = I + a_1\mathbf{A}^1 + \ldots + a_d\mathbf{A}^d = \mathbf{0} \ , \tag{5}$$

where the superscript "*" denotes the scaled polynomial. Equation (5) implies:

$$\mathbf{I} = -\left(a_1 + \ldots + a_2\mathbf{A}^1 + \ldots + a_d\mathbf{A}^{d-1}\right)\mathbf{A} \ . \tag{6}$$

Multiplying Eq. (6) from the right by $\mathbf{A}^{-1}$ gives

$$\mathbf{A}^{-1} = -\left(a_1 + a_2\mathbf{A}^1 + \ldots + a_d\mathbf{A}^{d-1}\right) \quad . \tag{7}$$

Thus, the solution to Eq. (1) can be written as

$$\overrightarrow{x} = \mathbf{A}^{-1}\overrightarrow{b} = -\left(a_1\overrightarrow{b} + \ldots + a_2\mathbf{A}^1\overrightarrow{b} + a_d\mathbf{A}^{d-1}\overrightarrow{b}\right) \quad , \tag{8}$$

which, by comparison with Eq. (3), is an element of $K_d(\mathbf{A}, \overrightarrow{b})$. Thus it is logical to seek an approximation to $\overrightarrow{x}$ from the Krylov space $K_n(\mathbf{A}, \overrightarrow{b})$, where $n \leq d$.

A typical Krylov method can be described as follows. The m'th solution iterate, $\overrightarrow{x}_m$, is an element of $K_m(\mathbf{A}, \overrightarrow{b})$. The first step in the algorithm is to build an orthogonal basis for $K_m(\mathbf{A}, \overrightarrow{b})$. Orthogonalization of the Krylov vectors is necessary because these vectors become less linearly independent with increasing m. This follows from the fact that $\mathbf{A}^j\overrightarrow{b}$ approaches the fundamental eigenvector (the eigenvector associated with the eigenvalue of largest magnitude) as j increases. Different methods achieve orthogonality in different ways, and in some cases, the vectors are orthogonal with respect to some particular inner product. For illustrative purposes we will simply assume that the vectors are orthogonal with respect to the standard dot product. For reasons that will be explained later, the orthogonal basis vectors are also normalized. The m'th solution iterate is expressed as a linear combination of the orthonormal basis vectors:

$$\overrightarrow{x}_m = \mathbf{V}_m\overrightarrow{y}_m \quad , \tag{9}$$

where $\mathbf{V}_m$ is an $n \times m$ matrix whose j'th column consists of the j'th orthonormal basis vector, and $\overrightarrow{y}_m$ is the vector of expansion coefficients of length m. The expansion coefficients are uniquely determined via a residual orthogonality condition. More specifically, an m-dimensional weighting space of vectors is first chosen,

$$W_m = \text{span}\{\overrightarrow{w}_1, \overrightarrow{w}_2, \ldots, \overrightarrow{w}_m\} \ , \tag{10}$$

and then expansion coefficients are chosen so that the residual vector associated with $\overrightarrow{x}_m$,

$$\overrightarrow{r}_m = \overrightarrow{b} - \mathbf{A}\overrightarrow{x}_m = \overrightarrow{b} - \mathbf{A}\mathbf{V}_m\overrightarrow{y}_m \ , \tag{11}$$

is orthogonal to the weighting space, i.e.,

$$\mathbf{W}_m^T\overrightarrow{r}_m = 0 \ , \tag{12}$$

where $\mathbf{W}_m$ is an $m \times m$ matrix whose j'th column consists of the j'th weighting vector. Substituting from Eq. (11) into Eq. (12), we obtain the following $m \times m$ matrix equation for the coefficient vector:

$$\mathbf{W}_m^T\mathbf{A}\mathbf{V}_m\overrightarrow{y}_m = \mathbf{W}_m^T\overrightarrow{b} \ . \tag{13}$$

Different Krylov methods arise from different choices of the weighting space. For instance, if $\mathbf{A}$ is positive-definite, $W_m = K_m(\mathbf{A}, \overrightarrow{b})$ is a good choice. This results in a classic Ritz-Galerkin approximation because the weighting space and the approximate solution space are identical:

$$\mathbf{V}_m^T\mathbf{A}\mathbf{V}_m\overrightarrow{y}_m = \mathbf{V}_m^T\overrightarrow{b} \ , \tag{14}$$

The **A**-norm of the error is minimized under this approximation. The **A**-norm of the error is defined as follows:

$$\| \overrightarrow{x} - \overrightarrow{x}_m \|_{\mathbf{A}} = \left(\overrightarrow{x} - \overrightarrow{x}_m \right)^T \mathbf{A} \left(\overrightarrow{x} - \overrightarrow{x}_m \right) \quad . \tag{15}$$

The quantity defined by Eq. (14) is always minimized, but it is not a true norm unless **A** is positive-definite. The conjugate-gradient (CG) method uses the Ritz-Galerkin approximation.

If **A** is not positive-definite, a good choice for the weighting space is $\mathbf{A}K_m(\mathbf{A}, \overrightarrow{b})$, i.e., the space spanned by the **A** times the Krylov vectors. In this case, Eq. (13) becomes

$$(\mathbf{A}\mathbf{V}_m)^T (\mathbf{A}\mathbf{V}_m) \overrightarrow{y}_m = (\mathbf{A}\mathbf{V}_m)^T \overrightarrow{b} \quad . \tag{16}$$

Equation (16) is easily recognized as the least-squares approximation to the over-determined linear system

$$\mathbf{A} \left(\mathbf{V}_m \overrightarrow{y}_m \right) = \overrightarrow{b} \quad . \tag{17}$$

Thus, the residual associated with this approximation is minimized with respect to the L_2 norm, defined as

$$\| \overrightarrow{r}_m \|_{L_2} = \sqrt{\overrightarrow{r}_m^T \overrightarrow{r}_m} \quad . \tag{18}$$

The generalized minimum-residual (GMRES) and minimum-residual (MINRES) methods use the least-squares approximation.

Weighting spaces other than those that we have discussed can be used, but when this is done, one usually finds that neither the error nor the residual are minimized in any rigorous sense. However, there are advantages to using other weighting spaces that relate to the amount of computational work and the amount of data storage required to solve for $\overrightarrow{x}_m$.

In general, Eq. (13) is not actually solved to obtain $\overrightarrow{y}_m$. For instance, in the GMRES method, an equivalent matrix equation based upon an Arnoldi decomposition of $\mathbf{A}$ is used. The details of the process used to obtain $\overrightarrow{x}_m$ are not important for our purposes, but what is important is the amount of computational work and the amount of data storage required to obtain $\overrightarrow{x}_m$. The storage costs associated with the Krylov vectors can be prohibitive if one is solving very large linear systems for which many iterations are required to achieve convergence. This is not a difficulty if $\mathbf{A}$ is symmetric positive-definite and the CG method is used, because $\overrightarrow{x}_m$ can be calculated using a three-term recursion formula. As a result, one need only save the previous solution iterate and one other vector of similar length to compute a new solution iterate. In this case, the required data storage does not increase with the number of iterations as it otherwise does. This is a great advantage associated with the CG method. If $\mathbf{A}$ is not symmetric positive-definite, one can avoid the storage issue by choosing the weighting space such that a recursion formula can be used to compute $\overrightarrow{x}_m$. However, as previously noted, this approach generally fails to minimize the error or the residual at each iteration step, which can result in erratic convergence

properties. The biconjugate-gradient (BCG) and quasi-minimum residual (QMR) methods use this type of approach. If a recursion formula cannot be used, a *restart* strategy must generally be used. Under such a strategy, one chooses a maximum number of Krylov iterations to perform within each "stage" before restarting the Krylov process with the last iterate from the previous stage serving as the initial solution guess for the next stage. The difficulty with this approach is that convergence of the restarted process is not guaranteed for general matrices. However, we note that convergence of the restarted GMRES method is guaranteed if $\mathbf{A}$ is positive-definite, i.e., if $\overrightarrow{x}^T \mathbf{A} \overrightarrow{x} > 0$ for all $\overrightarrow{x} \neq 0$, or equivalently, if the eigenvalues of $\mathbf{A} + \mathbf{A}^T$ are all positive and non-zero. As is later explained, this property is relevant for S_N calculations.

In summary, all Krylov methods use a Krylov space of dimension m to approximate the m'th solution iterate, and force the residual associated with that iterate to be orthogonal to an m-dimensional weighting space of vectors. The main differences between the various types of Krylov methods arise from the nature of the weighting space of vectors and the computational cost (both in CPU time and memory) of obtaining a solution iterate. Some Krylov methods are designed for general linear systems, while others are specifically designed to be optimal for some particular type of linear system. For instance, the conjugate-gradient method is designed for symmetric positive-definite systems while the minimum-residual method is designed for symmetric indefinite systems.

3 Convergence and Preconditioning of Krylov Methods

The speed of convergence of Krylov methods very much depends upon the characteristics of the coefficient matrix associated with the linear system being solved, i.e., the characteristics of the matrix $\mathbf{A}$ in Eq. (1). The coefficient matrix $\mathbf{A}$ is normal if $\mathbf{A}\mathbf{A}^T = \mathbf{A}^T\mathbf{A}$. The convergence properties associated with normal coefficient matrices are fairly well understood, but characterizing the convergence properties associated with non-normal coefficient matrices remains an open problem. A property associated with the convergence of the conjugate-gradient (CG) method is the *condition number* κ of the coefficient matrix, the ratio of its largest singular value to its smallest singular value:

$$\kappa = \frac{\sigma_{max}}{\sigma_{min}} \quad . \tag{19}$$

If κ is not large, the convergence of the CG method will be rapid. However, if κ is large, no conclusion can be made. The condition number is not particularly relevant to the convergence of the generalized minimum-residual (GMRES) method. More relevant factors are the distribution of the eigenvalues of $\mathbf{A}$ in the complex plane, and assuming that $\mathbf{A}$ is diagonalizable, the condition number of the matrix of eigenvectors of $\mathbf{A}$. An eigenvalue spectrum clustered away from zero and entirely contained in either half-plane, and an eigenvector matrix of $\mathbf{A}$ with a small condition number, are desirable. If any single property

of the coefficient matrix is desirable with a Krylov method, it is to have eigenvalues clustered away from zero. A set of eigenvalues is considered to be *clustered* if the distance between any two eigenvalues is much smaller than the distance of any eigenvalue from the origin. We stress that all of the eigenvalues need not be in one cluster or a collection of clusters. Any clustering of eigenvalues can be advantageous relative to a completely unclustered eigenvalue distribution.

The maximum number of iterations required for a Krylov method is N, the dimension of $\mathbf{A}$. However, it is desirable to take many fewer iterations, both from the viewpoint of efficiency and from the fact that roundoff errors can be problematic when N is large. If one has a linear system for which Krylov methods are inefficient, one can attempt to remedy the problem via *preconditioning*. A preconditioner for $\mathbf{A}$ is a matrix that approximates $\mathbf{A}^{-1}$ in some sense. Given a preconditioning matrix, which we denote by $\mathbf{C}$, the preconditioning process can take two forms. The first, called left-preconditioning, corresponds to solving

$$\mathbf{CA}\overrightarrow{x} = \mathbf{C}\overrightarrow{y} \quad . \tag{20}$$

In the absence of round-off error, The above equation clearly has the same solution as Eq. (1). The second is called right-preconditioning and corresponds to solving

$$\mathbf{AC}\overrightarrow{z} = \overrightarrow{y} \quad , \tag{21}$$

where

$$\overrightarrow{x} = \mathbf{C}\overrightarrow{z} \quad . \tag{22}$$

We note from Eq. (22) that once Eq. (21) has been solved, $\overrightarrow{x}$ is obtained from $\overrightarrow{z}$ via the action of $\mathbf{C}$ on $\overrightarrow{z}$. If $\mathbf{C} = \mathbf{A}^{-1}$, both Eqs. (20) and (21) can be solved in a single Krylov iteration. This is why a preconditioner should approximate $\mathbf{A}^{-1}$ in some sense. However, from a practical point of view, a preconditioner must approximate $\mathbf{A}^{-1}$ in some limited sense, since one would generally not expect to be able to find a nearly exact approximation to $\mathbf{A}^{-1}$ at a low computational cost.

Experience solving the transport equation with Krylov methods indicates that the best preconditioners move the the smallest eigenvalues significantly away from zero, while leaving the largest eigenvalues relatively unaffected. Assuming a preconditioned transport system that is SPD, the effectiveness of such preconditioners is easily explained when they are used in conjunction with the CG method. In particular, the eigenvalues and singular values of an SPD matrix are identical, so the condition number of such a matrix is just the ratio of the largest eigenvalue to the smallest eigenvalue. Thus, moving the eigenvalues nearest zero away from zero while leaving the largest eigenvalues essentially unaffected will decrease the condition number of the coefficient matrix and thereby result in faster convergence of the CG method. However, even though such preconditioners are observed to remain effective when used in conjunction with nonsymmetric transport systems and more general Krylov

methods, the reasons for their effectiveness are not yet understood. To move the smallest eigenvalues away from zero, the preconditioner must accurately approximate the inverse of the coefficient matrix when operating upon the eigenvectors associated with the smallest eigenvalues. To leave the largest eigenvalues relatively unaffected, the preconditioner must act approximately as the identity matrix when operating upon the eigenvectors associated with the largest eigenvalues. Such preconditioners are analogous to the coarse-grid operators used in multigrid methods and the low-order operators used in synthetic acceleration schemes. Thus it is somewhat clear how such preconditioners should be constructed.

As previously stated, the convergence of Krylov methods can be significantly improved by clustering some of the eigenvalues. Unfortunately, the properties required for a preconditioner to cluster eigenvalues are generally much less clear than those required to move the smallest eigenvalues away from zero. Nonetheless, as we shall later see, moderately effective preconditioners have been found for transport calculations that cluster some of the eigenvalues while leaving the smallest eigenvalues essentially unaltered.

Finally, moderately-effective preconditioners for transport calculations have been found that move the smallest eigenvalues away from zero, but also significantly increase and spread out (uncluster) the largest eigenvalues. One can give a plausible explanation for the effectiveness of such preconditioners when they are used in conjunction with the CG method, but such an explanation is not currently possible with more general Krylov methods. For

instance, if we assume a preconditioned transport system that is SPD, moving the smallest eigenvalues away from zero decreases the condition number of the coefficient matrix, while increasing the large eigenvalues increases the condition number of the coefficient matrix. We can conjecture that these preconditioners are moderately effective when used in conjunction with the CG method because the decrease in the condition number associated with moving the smallest eigenvalues away from zero weakly dominates the increase in the condition number associated with increasing the largest eigenvalues. This domination results in a net decrease in the condition number of the coefficient matrix and an attendant increase in the convergence rate of the CG method. Preconditioners of this type are analogous to unstable acceleration schemes that strongly attenuate the low-frequency Fourier error modes but weakly amplify the high-frequency Fourier error modes. We emphasize that not every acceleration scheme that attenuates low-frequency Fourier error modes and amplifies high-frequency Fourier error modes can be expected to be effective when recast as a preconditioned Krylov method. If the amplification factors for the high-frequency Fourier error modes are too large, the corresponding preconditioner may increase the largest eigenvalues too much relative to the movement of the smallest eigenvalues away from zero, presumably resulting in a net decrease in convergence rate.

4 Applying Krylov Methods to the S_N Equations

The optimal application of Krylov methods to the transport equation is not necessarily obvious. For instance, let us assume a monoenergetic approximation with isotropic scattering in 1-D slab geometry. The transport equation can be expressed as:

$$\left(\mathbf{L} - \frac{1}{2}\Sigma_s \mathbf{P}\right)\psi = Q \quad , \tag{23}$$

where

$$\mathbf{L}\psi = \mu\frac{\partial\psi}{\partial x} + \Sigma_t\psi \quad , \tag{24}$$

and

$$\mathbf{P}\psi = \phi = \int_{-1}^{+1} \psi \, d\mu \quad . \tag{25}$$

A Krylov method could be used to solve a fully discretized S_N version of Eq. (23), but it would not be optimal. A better approach is to left-precondition Eq. (24) with $\mathbf{L}^{-1}$:

$$\left(\mathbf{I} - \frac{1}{2}\mathbf{L}^{-1}\Sigma_s \mathbf{P}\right)\psi = \mathbf{L}^{-1}Q \quad . \tag{26}$$

This approach is better for three reasons. First, the analytic operator on the left side of Eq. (26) represents the integral transport operator, which is bounded, whereas the differential transport operator is unbounded. This implies that the discretized S_N version of the integral operator will have eigenvalues restricted to a much smaller region of the complex plane than the discretized S_N version of the differential operator. Furthermore,

the integral transport operator is a compact perturbation of the identity operator, so many (but not all) of the eigenvalues of the discretized S_N version of the integral operator will be clustered about one. Thus, Krylov methods will generally converge faster when used to solve Eq. (26) than when used to solve Eq. (23). The second reason for solving Eq. (26) is that the action of $\mathbf{L}^{-1}$ is easily and economically obtained via a sweep. Thus, the improved convergence rate will not be outweighed by the cost of preconditioning with $\mathbf{L}^{-1}$. We note that $\mathbf{L}^{-1}$ is an example of a preconditioner that clusters eigenvalues but does not significantly move the smallest eigenvalues away from zero. The improvement in convergence rate is significant, but preconditioning with $\mathbf{L}^{-1}$ will certainly not result in rapid convergence for all problems. The third reason for solving Eq. (26) is that it has been observed (but not rigorously proven) that the matrices associated with S_N discretizations of Eq. (26) are positive-definite. As previously noted, the restarted GMRES algorithm is guaranteed to converge when applied to linear systems with this property. This is a very significant property of Eq. (26).

A further improvement can be obtained by multiplying Eq. (26) by $\mathbf{P}$. This results in an integral equation for the scalar flux (also known as Peierls' equation):

$$\left(\mathbf{I} - \frac{1}{2}\mathbf{P}\mathbf{L}^{-1}\Sigma_s\right)\phi = \mathbf{P}\mathbf{L}^{-1}Q \quad . \tag{27}$$

The main advantage of this approach is that the dimension of the matrix associated with a discrete version of Eq. (27) is much less than that associated with a discrete version of

Eq. (26), greatly reducing the amount of computation required per Krylov iteration. Once the scalar flux is obtained from the Krylov solve, angular fluxes can be obtained with a single sweep:

$$\psi = \mathbf{L}^{-1}\frac{1}{2}\Sigma_s\mathbf{P}\psi + \mathbf{L}^{-1}Q \quad . \tag{28}$$

The derivation yielding Eq. (27) can be generalized for any order of anisotropic scattering, resulting in an integral equation for the Legendre (or spherical-harmonic) flux moments. Solving an integral moments equation will be advantageous relative to Eq. (26) whenever the number of moments is less than the number of discrete angular flux directions. Another potential advantage of the integral moments approach is that such equations can be made self-adjoint and positive-definite (SAPD) by defining an appropriate preconditioner in conjunction with a non-standard inner product. If a discretization for such an equation preserves the SAPD property, then the conjugate-gradient method can be used to solve the corresponding discrete system. SAPD has been observed to be preserved for Eq. (27) on orthogonal meshes with a wide variety S_N spatial discretization schemes, but it is not preserved with the linear-discontinuous spatial discretization scheme on unstructured tetrahedral meshes. The conditions under which SAPD is preserved are not currently well understood, but for reasons given later, this is not as important an issue as it might appear to be.

Finally, perhaps the optimal approach would be to multiply Eq. (27) by the operator

$(I + \mathbf{D}^{-1}\Sigma_s)$:

$$\left(I + \mathbf{D}^{-1}\Sigma_s\right)\left(\mathbf{I} - \frac{1}{2}\mathbf{P}\mathbf{L}^{-1}\Sigma_s\right)\phi = \left(I + \mathbf{D}^{-1}\Sigma_s\right)\mathbf{P}\mathbf{L}^{-1}Q \quad , \tag{29}$$

where $\mathbf{D}$ is the diffusion operator:

$$\mathbf{D}\phi = -\frac{\partial}{\partial x}\frac{1}{3\Sigma_t}\frac{\partial \phi}{\partial x} + \Sigma_a\phi \quad . \tag{30}$$

This is the analog of diffusion-synthetic acceleration, and it (presumably) results in rapid convergence under all conditions. The deficiency of DSA in multidimensions with large discontinuities in the cross sections is greatly mitigated when it is recast as a preconditioned Krylov method. This is the great advantage of Krylov methods. It only takes one eigenvalue of the iteration matrix very near unity to ruin the performance of an acceleration scheme, but a relatively small number of large eigenvalues associated with the system being solved will generally have a small effect upon a Krylov method.

As previously noted, modern S_N codes generally use discontinuous spatial discretization schemes. If a discontinuous diffusion discretization is used, the left side of Eq. (29) will not be symmetric. A great advantage of preconditioned Krylov methods is that the diffusion discretization does not have to be fully consistent with the S_N spatial discretization to be effective. However, if a discontinuous S_N spatial discretization is used, some capability for calculating a discontinuous solution must be included in the diffusion approximation to ensure unconditional effectiveness, and this will almost certainly cause the overall diffusion

operator to be either non-symmetric or symmetric-indefinite. Thus, even if a discontinuous discretization of Eq. (27) were to preserve the symmetry and positive-definiteness of Peierls' operator, applying a DSA-like preconditioner would result in a non-symmetric positive-definite system, precluding the use of the conjugate-gradient method to solve that system. This is why the failure of discontinuous methods to preserve the SAPD property of Peierls' operator on unstructured meshes is not as important an issue as one might expect it to be. On the other hand, even when a discontinuous diffusion approximation is used, the operator on the left side of Eq. (29) remains positive-definite, ensuring the convergence of the restarted GMRES method. This is a very important property of Eq. (29).

Next, we review the operations required to solve a fully discretized S_N version of Eq. (29) and show that they are quite similar to the operations that take place in a standard S_N code with diffusion-synthetic acceleration. We first note that to solve a general linear system such as Eq. (1), most Krylov routines do not require the formation of the matrix $\mathbf{A}$. Rather, it is required that the calling routine provide the Krylov routine with a vector that represents the action of $\mathbf{A}$. Specifically, the Krylov routine provides the calling program with a vector, which we denote by $\overrightarrow{v}$, and the calling routine returns the vector $\overrightarrow{v}'$, where

$$\overrightarrow{v}' = \mathbf{A}\,\overrightarrow{v} \quad . \tag{31}$$

It is not difficult to see that the Krylov space can be built up this way with one matrix-vector multiply per iteration. The simplest Krylov methods (e.g., the conjugate-gradient

method) require only one matrix-vector multiply per iteration, but others may require more (e.g., the biconjugate-gradient method requires two).

Let us now consider the solution of Eq. (29) in detail. Since the unknown in Eq. (29) is the monoenergetic or one-group scalar flux, a Krylov method used to solve this equation works with "scalar flux" vectors, i.e., vectors with a length equal to the number of spatial cells times the number of spatial unknowns per cell. The latter number clearly depends upon the spatial discretization used for Eq. (29). In the process of generating the scalar flux vectors required by the Krylov solver, intermediate "angular flux vectors" arise. These vectors have a length equal to the length of a scalar flux vector times the number of discrete directions in the quadrature set. In the descriptions that follow, angular flux vectors carry a superscript "a" to differentiate them from scalar flux vectors. The first vectors that must be provided to a Krylov solver are the initial solution guess, and the vector corresponding to the right side of Eq. (29). The latter "source" vector is generated as follows:

1. Perform a sweep with Q as the source to obtain

$$\overrightarrow{v}^{a}_{1} = \mathbf{L}^{-1}Q \quad .$$

2. Integrate $\overrightarrow{v}^{a}_{1}$ over all directions to obtain

$$\overrightarrow{v}_{2} = \mathbf{P}\mathbf{L}^{-1}Q \quad .$$

3. Multiply $\overrightarrow{v}^{s}_{2}$ by Σ_s (a diagonal matrix in this context) to obtain

$$\overrightarrow{v}_{3} = \Sigma_s \mathbf{P}\mathbf{L}^{-1}Q \ .$$

4. Solve the diffusion equation with $\overrightarrow{v}_{3}$ as the source to obtain

$$\overrightarrow{v}_{4} = \mathbf{D}^{-1}\Sigma_s \mathbf{P}\mathbf{L}^{-1}Q \ .$$

5. Add $\overrightarrow{v}_{2}$ to $\overrightarrow{v}_{4}$ to obtain the desired source vector:

$$\overrightarrow{v}_{5} = \left(\mathbf{I} + \mathbf{D}^{-1}\Sigma_s\right) \mathbf{P}\mathbf{L}^{-1}Q \ .$$

Next, every iteration requires the calling routine to provide the Krylov solver with the action of the operator on the left side of Eq. (29). The action of this operator on the vector $\overrightarrow{v}$ is calculated as follows:

1. Multiply $\overrightarrow{v}$ by $\frac{1}{2}\Sigma_s$ to obtain

$$\overrightarrow{v}_{1} = \frac{1}{2}\Sigma_s \overrightarrow{v} \ .$$

2. Perform a sweep with $\overrightarrow{v}_{1}$ as the source to obtain

$$\overrightarrow{v}^{a}_{2} = \frac{1}{2}\mathbf{L}^{-1}\Sigma_s \overrightarrow{v} \ .$$

3. Integrate $\overrightarrow{v}^{a}_{2}$ over all angles to obtain

$$\overrightarrow{v}_{3} = \frac{1}{2}\mathbf{P}\mathbf{L}^{-1}\Sigma_s \overrightarrow{v} \ .$$

4. Subtract $\overrightarrow{v}_3$ from $\overrightarrow{v}'$ to obtain

$$\overrightarrow{v}_4 = \left(\mathbf{I} - \frac{1}{2}\mathbf{P}\mathbf{L}^{-1}\Sigma_s\right)\overrightarrow{v} \quad .$$

5. Multiply $\overrightarrow{v}_4$ by Σ_s to obtain

$$\overrightarrow{v}_5 = \Sigma_s\left(\mathbf{I} - \frac{1}{2}\mathbf{P}\mathbf{L}^{-1}\Sigma_s\right)\overrightarrow{v} \quad .$$

6. Perform a diffusion calculation with $\overrightarrow{v}_5$ as the source to obtain

$$\overrightarrow{v}_6 = \mathbf{D}^{-1}\Sigma_s\left(\mathbf{I} - \frac{1}{2}\mathbf{P}\mathbf{L}^{-1}\Sigma_s\right)\overrightarrow{v} \quad .$$

7. Add $\overrightarrow{v}_4$ to $\overrightarrow{v}_5$ to obtain the desired action on $\overrightarrow{v}$

$$\overrightarrow{v}_7 = \left(I + \mathbf{D}^{-1}\Sigma_s\right)\left(\mathbf{I} - \frac{1}{2}\mathbf{P}\mathbf{L}^{-1}\Sigma_s\right)\overrightarrow{v} \quad .$$

Thus we see from the above description that nothing other than sweeps, diffusion solves, and vector additions and subtractions are required to solve the transport equation via a uniformly effective preconditioned Krylov method. Since these same operations occur in standard S_N codes with diffusion-synthetic acceleration, then it is not difficult to modify existing S_N codes to use preconditioned Krylov solution techniques.