

**NE 255, Fa16**  
**Operator Form and Iteration Methods**  
**October 20, 2016**

---

We have now fully discretized the transport equation:

$$\hat{\Omega}_a \cdot \nabla \psi_a^g(\vec{r}) + \Sigma_t^g(\vec{r}) \psi_a^g(\vec{r}) = \sum_{g'=1}^G \sum_{l=0}^N \Sigma_{sl}^{gg'}(\vec{r}) \left[ Y_{l0}^e(\hat{\Omega}_a) \phi_{l0}^{g'}(\vec{r}) + \sum_{m=1}^l (Y_{lm}^e(\hat{\Omega}_a) \phi_{lm}^{g'}(\vec{r}) + Y_{lm}^o(\hat{\Omega}_a) \vartheta_{lm}^{g'}(\vec{r})) \right] + q_{a,e}^g(\vec{r}), \quad (1)$$

$$\phi_{lm}^g = \int_{4\pi} Y_{lm}^e(\hat{\Omega}) \psi^g(\hat{\Omega}) d\hat{\Omega}, \quad m \geq 0, \quad (2)$$

$$\vartheta_{lm}^g = \int_{4\pi} Y_{lm}^o(\hat{\Omega}) \psi^g(\hat{\Omega}) d\hat{\Omega}, \quad m > 0. \quad (3)$$

Now that we have these equations (+ spatial discretization) we're going to look at how we actually solve this with a computer.

The next set of material comes from

R. N. Slaybaugh. *Acceleration Methods for Massively Parallel Deterministic Transport*, a PhD Dissertation. University of Wisconsin, Madison, WI (2011).

T. M. Evans, A. S. Stafford, R. N. Slaybaugh, and K. T. Clarno. "Denovo – New Three-Dimensional Parallel Discrete Ordinates Code in SCALE." *Nuclear Technology*, **volume 171**(2), pp. 171200 (2010).

## Operator Form

We start by expressing the TE in operator notation; which makes it easier to talk about solution techniques. In general, uppercase bolded letters will indicate matrices and lowercase italicized letters will indicate vectors and scalars. The following operators are used to express the transport equation:

$\mathbf{L} = \hat{\Omega} \cdot \nabla + \Sigma_t$  is the transport operator,

$\mathbf{M}$  is the operator that converts harmonic moments into discrete angles,

$\mathbf{S}$  is the scattering matrix,

$q_e$  contains the external source,  
 $f$  contains the fission source,  $\nu\Sigma_f$ ;  $\mathbf{F} = \chi f^T$ ,  
 $\mathbf{D} = \mathbf{M}^T \mathbf{W} = \sum_{a=1}^n Y_{lm}^{e/o} w_a$  is the discrete-to-moment operator.

With this notation, Equation 1 can be written as Equation 4; it can be formulated as an eigenvalue problem by replacing the fixed source term with the fission term,  $\frac{1}{k} \mathbf{M} \mathbf{F} \phi$ . This has two unknowns, the angular flux and the moments, which are related by the discrete-to-moment operator as seen in Equation (5) [NOTE:  $\phi$  are the flux moments, **not** the scalar flux].

$$\mathbf{L}\psi = \mathbf{M}\mathbf{S}\phi + \mathbf{M}q_e \quad (4)$$

$$\phi = \mathbf{D}\psi \quad (5)$$

The size of the operators can be defined in terms of the granularity of discretization:

$G$  = number of energy groups,  
 $N$  = number of moments,  
 $n$  = number of angular unknowns,  
 $c$  = number of cells,  
 $u$  = number of unknowns per cell, which is determined by spatial discretization.

These can be combined to define

$\alpha = G \times n \times c \times u$  and  
 $\beta = G \times N \times c \times u$ .

Using  $\alpha$  and  $\beta$ , Equation (4) can be presented in terms of operator size:

$$(\alpha \times \alpha)(\alpha \times 1) = (\alpha \times \beta)(\beta \times \beta)(\beta \times 1) + (\alpha \times \beta)(\beta \times 1) .$$

The index variables, their meaning, and their ranges are shown in Table 1.

Table 1: Meaning and Range of Indices Used in Transport Discretization

Variable	Symbol	First	Last
Energy	$g$	1	$G$
Solid Angle	$a$	1	$n$
Space	suppressed	n/a	n/a
Legendre moment ( $P_N$ )	$l$	0	$N$
Spherical harmonic moment ( $Y$ )	$m$	0	$l$

The structures of the vectors and matrices are shown in the next few equations as this can make the whole thing easier to understand and visualize. The angular flux vector is explicitly written first, where for each discrete angle,  $a$ , and energy group,  $g$ , the set of angular fluxes,  $\psi_a^g$ , includes all

spatial unknowns.

$$\psi = \left( [\psi]_1 \quad [\psi]_2 \quad \cdots \quad [\psi]_g \quad \cdots \quad [\psi]_G \right)^T, \quad \text{and} \quad (6)$$

$$[\psi]_g = \left( \psi_1^g \quad \psi_2^g \quad \cdots \quad \psi_a^g \quad \cdots \quad \psi_n^g \right)^T. \quad (7)$$

$$\mathbf{M} = \begin{pmatrix} [\mathbf{M}]_{11} & 0 & 0 & \cdots & 0 \\ 0 & [\mathbf{M}]_{22} & 0 & \cdots & 0 \\ 0 & 0 & [\mathbf{M}]_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & [\mathbf{M}]_{GG} \end{pmatrix}, \quad \mathbf{S} = \begin{pmatrix} [\mathbf{S}]_{11} & [\mathbf{S}]_{12} & [\mathbf{S}]_{13} & \cdots & [\mathbf{S}]_{1G} \\ [\mathbf{S}]_{21} & [\mathbf{S}]_{22} & [\mathbf{S}]_{23} & \cdots & [\mathbf{S}]_{2G} \\ [\mathbf{S}]_{31} & [\mathbf{S}]_{32} & [\mathbf{S}]_{33} & \cdots & [\mathbf{S}]_{3G} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ [\mathbf{S}]_{G1} & [\mathbf{S}]_{G2} & [\mathbf{S}]_{G3} & \cdots & [\mathbf{S}]_{GG} \end{pmatrix},$$

$$\mathbf{F} = \begin{pmatrix} \chi^1[\nu\Sigma_f]^1 & \chi^1[\nu\Sigma_f]^2 & \cdots & \chi^1[\nu\Sigma_f]^G \\ \chi^2[\nu\Sigma_f]^1 & \chi^2[\nu\Sigma_f]^2 & \cdots & \chi^2[\nu\Sigma_f]^G \\ \vdots & \vdots & \ddots & \vdots \\ \chi^G[\nu\Sigma_f]^1 & \chi^G[\nu\Sigma_f]^2 & \cdots & \chi^G[\nu\Sigma_f]^G \end{pmatrix}, \quad [\mathbf{S}]_{gg'} = \begin{pmatrix} \Sigma_{s0}^{gg'} & 0 & \cdots & 0 \\ 0 & \Sigma_{s1}^{gg'} & \cdots & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & \Sigma_{sN}^{gg'} \end{pmatrix},$$

$$[\mathbf{M}]_{gg} = \begin{pmatrix} Y_{00}^e(\hat{\Omega}_1) & Y_{10}^e(\hat{\Omega}_1) & Y_{11}^o(\hat{\Omega}_1) & Y_{11}^e(\hat{\Omega}_1) & Y_{20}^e(\hat{\Omega}_1) & \cdots & Y_{NN}^o(\hat{\Omega}_1) & Y_{NN}^e(\hat{\Omega}_1) \\ Y_{00}^e(\hat{\Omega}_2) & Y_{10}^e(\hat{\Omega}_2) & Y_{11}^o(\hat{\Omega}_2) & Y_{11}^e(\hat{\Omega}_2) & Y_{20}^e(\hat{\Omega}_2) & \cdots & Y_{NN}^o(\hat{\Omega}_2) & Y_{NN}^e(\hat{\Omega}_2) \\ Y_{00}^e(\hat{\Omega}_3) & Y_{10}^e(\hat{\Omega}_3) & Y_{11}^o(\hat{\Omega}_3) & Y_{11}^e(\hat{\Omega}_3) & Y_{20}^e(\hat{\Omega}_3) & \cdots & Y_{NN}^o(\hat{\Omega}_3) & Y_{NN}^e(\hat{\Omega}_3) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ Y_{00}^e(\hat{\Omega}_n) & Y_{10}^e(\hat{\Omega}_n) & Y_{11}^o(\hat{\Omega}_n) & Y_{11}^e(\hat{\Omega}_n) & Y_{20}^e(\hat{\Omega}_n) & \cdots & Y_{NN}^o(\hat{\Omega}_n) & Y_{NN}^e(\hat{\Omega}_n) \end{pmatrix}.$$

Note that  $[\mathbf{M}]_{11} = [\mathbf{M}]_{22} = \cdots = [\mathbf{M}]_{GG} = [\mathbf{M}]$ . These are written with subscripts to simplify the visualization of which blocks correspond to which equations and multiply which other blocks.

The lower triangular part of  $\mathbf{S}$  represents downscattering, the diagonal represents in-group scattering, and the upper diagonal is upscattering.

Finally, the vector of flux moments for group  $g$  has  $t$  entries per spatial unknown, and

$$[\phi]_g = \left( \phi_{00}^g \quad \phi_{10}^g \quad \vartheta_{11}^g \quad \phi_{11}^g \quad \phi_{20}^g \quad \cdots \quad \vartheta_{NN}^g \quad \phi_{NN}^g \right)^T, \quad (8)$$

for each group.

Note also that the total size of  $\mathbf{D}$  over all groups and spatial unknowns is  $(\beta \times \alpha)$  (where  $\mathbf{W}$  is an  $(n \times n)$  diagonal matrix of diagonal matrices of quadrature weights). Also, even though  $\mathbf{M}$  maps angular flux moments onto discrete angular fluxes, in general  $\psi \neq \mathbf{M}\phi$ . This constraint can be removed by using Galerkin quadrature in which  $\mathbf{D} = \mathbf{M}^{-1}$ .

## Solution Procedure

Once the matrices are multiplied together, a series of single-group equations that are each only a function of space and angle result:

$$\begin{aligned} \mathbf{L}[\psi]_1 &= [\mathbf{M}]([\mathbf{S}]_{11}[\phi]_1 + [\mathbf{S}]_{12}[\phi]_2 + \dots + [\mathbf{S}]_{1G}[\phi]_G) + [\mathbf{M}][q_e]_1, \\ \mathbf{L}[\psi]_2 &= [\mathbf{M}]([\mathbf{S}]_{21}[\phi]_1 + [\mathbf{S}]_{22}[\phi]_2 + \dots + [\mathbf{S}]_{2G}[\phi]_G) + [\mathbf{M}][q_e]_2, \\ &\vdots \\ \mathbf{L}[\psi]_G &= [\mathbf{M}]([\mathbf{S}]_{G1}[\phi]_1 + [\mathbf{S}]_{G2}[\phi]_2 + \dots + [\mathbf{S}]_{GG}[\phi]_G) + [\mathbf{M}][q_e]_G, \end{aligned} \tag{9}$$

Each **within-group** equation is solved for that group's flux.

If the groups are coupled together through upscattering, as they often are, then multiple **multigroup** solves over the coupled portion of the energy range may be required.

If there is fission and the eigenvalue is desired, an additional **eigenvalue** solve is needed as well.

Each of these levels of iteration typically uses a completely different solver.

Note that the typical strategy for solving Equation (4) is to combine it with (5) and form one equation involving only the moments, where  $Q = \mathbf{D}\mathbf{L}^{-1}\mathbf{M}q_e$  is comprised of the fixed (or fission) source:

$$(\mathbf{I} - \mathbf{D}\mathbf{L}^{-1}\mathbf{M}\mathbf{S})\phi = Q. \tag{10}$$

This is solved for  $\phi$  from which  $\psi$  can be determined at the end of the calculation. This format looks like what we're used to seeing in numerical linear algebra:  $\mathbf{A}x = b$ .

## Solver Basics

There are two categories of methods for solving  $\mathbf{A}x = b$ , **direct** and **iterative**.

Direct methods solve the problem by inverting  $\mathbf{A}$  and setting  $x = \mathbf{A}^{-1}b$ . If  $\mathbf{A}$  is invertible this

can be done explicitly.  $\mathbf{A}$  is often not invertible, so most direct methods are based on factoring the coefficient matrix  $\mathbf{A}$  into matrices that are easy to invert. The problem is then solved in pieces where each factored matrix is inverted to get the final solution. An example where this is done is LU factorization. These methods are often robust and require a predictable amount of time and storage resources. However, direct methods scale poorly with problem size, becoming increasingly expensive as problems grow large.

Iterative methods compute a sequence of increasingly accurate approximations to the solution. They generally require less storage and take fewer operations than direct methods, though they may not be as reliable. Iterative methods are highly advantageous for large problems because direct methods become intractable for systems of the size of those of interest here. For this reason the nuclear energy industry tends to use iterative methods for transport calculations.

Within the category of iterative methods, two further subdivisions can be made that will be useful in this work. Some methods only place data from previous iterations on the right hand side of the equation. The order in which those equations are solved is then irrelevant because only the previous iterate is needed. The other class of methods place data from both the previous and current iteration on the right hand side. These methods are fundamentally sequential and must be solved in order. These two categories will be referred to as *order independent* and *order dependent*, respectively.

## Within Group Iterative Methods

The methods that functionally perform a mesh sweep

$$[\psi]_g = \mathbf{L}^{-1}([\mathbf{M}]([\mathbf{S}]_{11}[\phi]_1 + [\mathbf{S}]_{12}[\phi]_2 + \dots + [\mathbf{S}]_{1G}[\phi]_G) + [\mathbf{M}][q_e]_1) .$$

### Richardson iteration

The simplest iteration scheme used by the nuclear community is source iteration (SI), also known as **Richardson iteration**. SI is applied to the within-group space-angle iterations. Richardson iteration can be thought of as a two-part process for the neutron transport equation, where  $\bar{Q}$  includes all sources and  $k$  is the inner iteration index:

$$\mathbf{L}[\psi]_g^{k+1} = \mathbf{MS}[\phi]_g^k + [\bar{Q}]_g , \quad (11)$$

$$[\phi]_g^{k+1} = \mathbf{D}[\psi]_g^{k+1} . \quad (12)$$

The spectral radius,  $c = \Sigma_s / \Sigma_t$ , determines the speed of convergence. For problems dominated by scattering, SI will converge very slowly (good explanation of how to think of this physically on p. 2, 3 of <https://github.com/rachelslaybaugh/NE250/blob/master/23-iterations/Nov-30Class.tex>, which you can build using pdflatex).

## Krylov methods

**Krylov methods** are a powerful class of subspace methods that can be ideal for solving various types of linear and eigenvalue problems. A Krylov method solves  $\mathbf{A}x = b$  by building a solution from a Krylov subspace generated by an iteration vector  $v_1$ . At iteration  $k$ , the subspace is:

$$\mathcal{K}_k(\mathbf{A}, v_1) \equiv \text{span}\{v_1, \mathbf{A}v_1, \mathbf{A}^2v_1, \dots, \mathbf{A}^{k-1}v_1\}. \quad (13)$$

The choice of  $v_1$  varies, but  $v_1 = b$  is common.

The dimension of a Krylov space is bounded by  $n$  because Krylov methods will give the exact solution after  $n$  iterations in the absence of roundoff error. Interestingly, this technically makes Krylov methods a hybrid of direct and iterative methods because an exact answer can be obtained in a predetermined number of steps. Krylov subspace methods are nevertheless generally classified as iterative methods.

Krylov methods are particularly useful in a few pertinent cases. One is when  $\mathbf{A}$  is very large because fewer operations are required than traditional inversion methods like Gaussian elimination. Another is when  $\mathbf{A}$  is not explicitly formed because Krylov methods only need the action of  $\mathbf{A}$ . Finally, Krylov methods are ideal when  $\mathbf{A}$  is sparse because the number of operations are low for each matrix-vector multiplication. For deterministic transport codes,  $\mathbf{A}$  is typically quite large, fairly sparse, and only its action is needed. The action of  $\mathbf{A}$  is implemented through the transport sweeps.

In the last few decades Krylov methods have been used widely to solve problems with appropriate properties for several reasons. Krylov methods are robust; the existence and uniqueness of the solution can be established; typically far fewer than  $n$  iterations are needed when they are used as iterative solvers; they can be preconditioned to significantly reduce time to solution; only matrix-vector products are required; explicit construction of intermediate residuals is not needed; and they have been found to be highly efficient in practice.

There are, however, a few drawbacks. In some cases Krylov methods can be very slow to converge, causing large subspaces to be generated and thus becoming prohibitively expensive in terms of

storage size and cost of computation. Some methods can be restarted after  $m$  steps to alleviate this problem, keeping the maximum subspace size below  $\mathcal{K}_{m+1}$ . The relatively inexpensive restart techniques can reduce the storage requirements and computational costs associated with a slow-converging problem such that they are tractable. Preconditioners can also help by reducing the number of iterations needed.

See aside about Krylov Methods here: <https://github.com/rachelslaybaugh/NE255/blob/gh-pages/05-fxsrc-solns/05-2-krylov.pdf> for more information.

## Outer Iterations

Once we have solved the space-angle component inside of a group, we need to loop over groups. There are several strategies for this, with Jacobi and Gauss Seidel being the most common (and multigroup Krylov becoming more common).

### Jacobi

The Jacobi method is known as the method of *simultaneous displacements*. This could be applied as the outer iteration method over energy and can be written as follows, where  $j$  is the outer iteration index:

$$\mathbf{L}[\psi]_g^{j+1} = [\mathbf{M}][\mathbf{S}]_{gg}[\phi]_g^j + [\mathbf{M}]\left(\sum_{g'=1}^{g-1} [\mathbf{S}]_{gg'}[\phi]_{g'}^j + \sum_{g'=g+1}^G [\mathbf{S}]_{gg'}[\phi]_{g'}^j + [q_e]_g\right). \quad (14)$$

The Jacobi method is order *independent* since all terms on the right are at the old iteration level. Jacobi is unconditionally stable and linearly convergent, but may converge very slowly. The convergence of Jacobi is governed by its spectral radius.

---

#### Aside:

The spectrum of eigenvalues of a matrix  $\mathbf{A}$  are defined formally as

$$\sigma(\mathbf{A}) = [\lambda \in \mathbb{C} : \det(\mathbf{A} - \lambda \mathbf{I}) = 0]$$

and an eigenvalue is

$$\lambda \in \sigma(\mathbf{A}).$$

The spectral radius of  $\mathbf{A} \in \mathbb{C}^{n \times n}$  is defined as

$$\rho(\mathbf{A}) \equiv \max\{|\lambda|, \lambda \in \sigma(\mathbf{A})\}$$


---

## Gauss Seidel

Gauss Seidel (GS) is known as the method of *successive displacements*. It is commonly used as the outer iteration method over energy and can be written as follows:

$$\mathbf{L}[\psi]_g^{j+1} = [\mathbf{M}][\mathbf{S}]_{gg}[\phi]_g^{j+1} + [\mathbf{M}]\left(\sum_{g'=1}^{g-1} [\mathbf{S}]_{gg'}[\phi]_{g'}^{j+1} + \sum_{g'=g+1}^G [\mathbf{S}]_{gg'}[\phi]_{g'}^j + [q_e]_g\right). \quad (15)$$

The GS method is order *dependent*, containing terms on the right hand side at both the new and old iterates. Once the space-angle iterations for each upscattering group are completed for one outer iteration, the right hand side is recalculated and the within-group calculations are repeated. This goes on until the entire system converges.

Gauss Seidel is also unconditionally stable and linearly convergent, but may converge very slowly. The convergence of GS is also governed by its spectral radius,  $\rho$ , though it converges twice as fast as the Jacobi method.

The spectral radius of Gauss Seidel has been found for various materials by solving the eigenvalue problem  $(\mathbf{T} - \mathbf{S}_D)^{-1}\mathbf{S}_U\xi = \rho\xi$ , where  $\mathbf{T}$  is the matrix of total cross sections,  $\mathbf{S}_U$  is the upper triangular portion of  $\mathbf{S}$ ,  $\mathbf{S}_D$  contains the diagonal and lower triangular portion, and  $\xi$  is the eigenvector. This eigenvalue problem comes from Fourier analysis. The spectral radii for some materials of practical interest determined using cross sections with 41 thermal upscattering groups are: graphite = 0.9984, heavy water = 0.9998, and iron = 0.6120. Problems containing graphite or heavy water would converge very slowly if GS were used.