

Solution Procedure

Recall that we are interested in solving the operator form of the TE:

$$\mathbf{L}\psi = \mathbf{M}\mathbf{S}\phi + \mathbf{M}q_e \quad (1)$$

$$\phi = \mathbf{D}\psi \quad (2)$$

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

$$\begin{aligned} \mathbf{L}[\psi]_0 &= [\mathbf{M}]([\mathbf{S}]_{00}[\phi]_0 + [\mathbf{S}]_{01}[\phi]_1 + \dots + [\mathbf{S}]_{0,G-1}[\phi]_{G-1}) + [\mathbf{M}][q_e]_0, \\ \mathbf{L}[\psi]_1 &= [\mathbf{M}]([\mathbf{S}]_{10}[\phi]_0 + [\mathbf{S}]_{11}[\phi]_1 + \dots + [\mathbf{S}]_{1,G-1}[\phi]_{G-1}) + [\mathbf{M}][q_e]_1, \\ &\vdots \\ \mathbf{L}[\psi]_{G-1} &= [\mathbf{M}]([\mathbf{S}]_{G-1,0}[\phi]_0 + [\mathbf{S}]_{G-1,1}[\phi]_1 + \dots + [\mathbf{S}]_{G-1,G-1}[\phi]_{G-1}) + [\mathbf{M}][q_e]_{G-1}, \end{aligned} \quad (3)$$

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 (1) is to combine it with (2) 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:

$$(\mathbf{I} - \mathbf{D}\mathbf{L}^{-1}\mathbf{M}\mathbf{S})\phi = Q. \quad (4)$$

This is solved for ϕ from which ψ 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

$$[\psi]_g = \mathbf{L}^{-1}([\mathbf{M}]([\mathbf{S}]_{0 \rightarrow 0}[\phi]_0 + [\mathbf{S}]_{1 \rightarrow 0}[\phi]_1 + \dots + [\mathbf{S}]_{G-1 \rightarrow 0}[\phi]_{G-1}) + [\mathbf{M}][q_e]_g) .$$

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:

$$\mathbf{L}[\psi]_g^{k+1} = \mathbf{MS}[\phi]_g^k + [\bar{Q}]_g, \quad (5)$$

$$[\phi]_g^{k+1} = \mathbf{D}[\psi]_g^{k+1}. \quad (6)$$

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:

$$\mathcal{K}_k(\mathbf{A}, v_1) \equiv \text{span}\{v_1, \mathbf{A}v_1, \mathbf{A}^2v_1, \dots, \mathbf{A}^{k-1}v_1\}. \quad (7)$$

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-fxdsrc-solns/05-2-krylov.pdf> for more information.