SP_N Update

Stuart R. Slattery
Engineering Physics Department
University of Wisconsin - Madison

March 20, 2013

Neutron Transport Solution Methods

$$\hat{\Omega} \cdot \vec{\nabla}\psi(\vec{r}, \hat{\Omega}, E) + \sigma(\vec{r}, E)\psi(\vec{r}, \hat{\Omega}, E) = \int \int \sigma_s(\vec{r}, E' \rightarrow E, \hat{\Omega}' \rightarrow \hat{\Omega})\psi(\vec{r}, \hat{\Omega}', E')d\Omega' dE' + q(\vec{r}, \hat{\Omega}, E) \quad (1)$$

- S_N transport is expensive
  - Difficult to parallelize: sweeps in space, pipelining in angle, energy decoupling
  - Large storage requirements
  - Ray effects
- P_N equations still expensive
  - Complicated system: $(N+1)^2$ equations in 3D
  - Coupling of equations through both angular moments and spatial derivatives

SP_N Approximation

- Ad-hoc generalization of planar P_N equations by Gelbard in the 1960's
- Rigorous formulation through asymptotic and variational analysis in 1990's and 2000's
- Simpler system $(N+1)/2$ equations in $3D$
- Yields elliptic, diffusion-like equations
- Applicable when diffusion theory is applicable: reasonable flux gradients, full-core transport
- Typically does not converge to transport solution as $N \rightarrow \infty$
- **Can build the full linear operator**
- **Parallelism through the linear solver**

SP_N Approximation

SP_N Equations

$$
\begin{aligned}
-\nabla \cdot \Bigg[& \frac{n}{2n+1} \frac{1}{\Sigma_{n-1}} \nabla \left(\frac{n-1}{2n-1} \phi_{n-2} + \frac{n}{2n-1} \phi_n \right) \\
& + \frac{n+1}{2n+1} \frac{1}{\Sigma_{n+1}} \nabla \left(\frac{n+1}{2n+3} \phi_n + \frac{n+2}{2n+3} \phi_{n+2} \right) \Bigg] \\
& + \Sigma_n \phi_n = q \delta_{n0} \qquad n = 0, 2, 4, \cdots, N \, , \quad (2)
\end{aligned}
$$

SP_N Numerical Spectral Analysis

- Monte Carlo methods for have strong restrictions on the eigenvalues of the operator for convergence
- MCSA has the same restrictions on the outer stationary iteration
- We need to compute these eigenvalues for various forms of the SP_N equations to verify convergence of these methods.

We need eigenvalues for $\mathbf{A}$, $\mathbf{H_J}$, and $\mathbf{H_{GS}}$ with:

$$\mathbf{H_J} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{A} \tag{3}$$

where $\mathbf{D} = diag(\mathbf{A})$ and

$$\mathbf{H_{GS}} = (\mathbf{L} + \mathbf{D})^{-1}\mathbf{U} \tag{4}$$

SP_7, P_3, 3 groups, reflecting boundaries

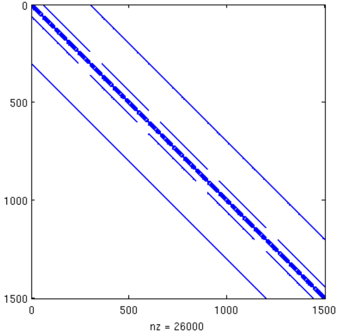

Figure: **Linear operator sparsity pattern**

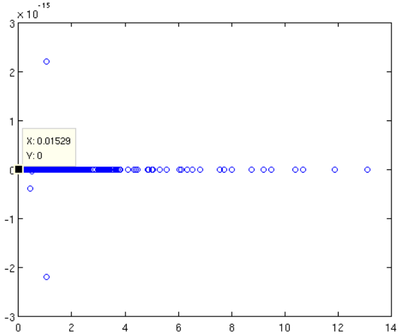

Figure: **Linear operator eigenvalues**

SP_7, P_3, 3 groups, reflecting boundaries

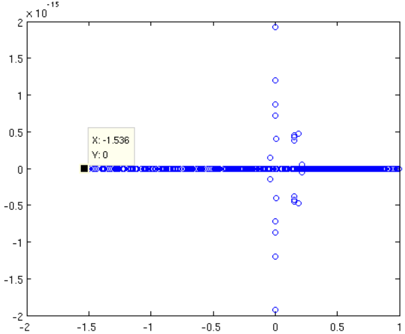

Figure: **Jacobi iteration matrix eigenvalues**

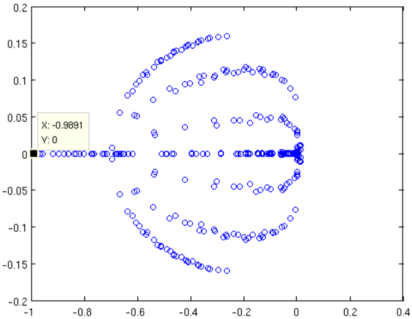

Figure: **Gauss-Seidel iteration matrix eigenvalues**

Oh No!

- The Jacobi method won't converge - all that stuff I said in my prelim won't work

Solution

- Bug fix in Denovo SP_N implementation
- A new kind of preconditioning $\cdots$

Multigroup Matrix Pattern

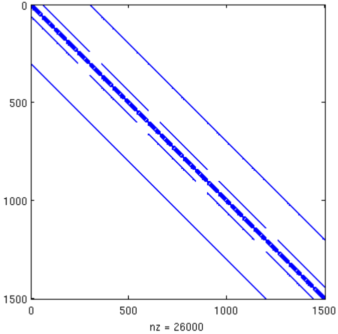

Figure: **Linear operator sparsity pattern**

Block Jacobi Preconditioning

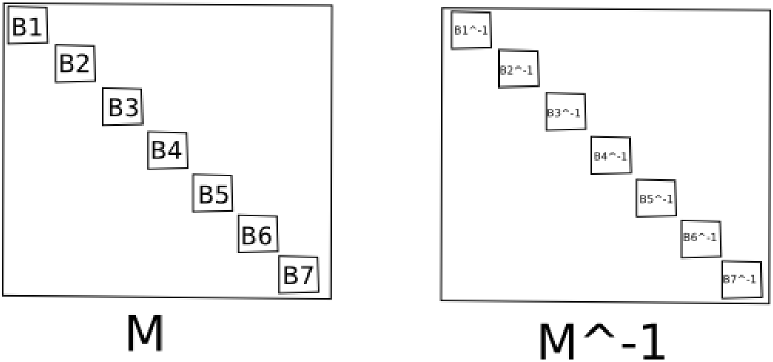

Figure: **Block Jacobi Preconditioner**

Point Jacobi Results

Block Jacobi Results

Conclusions

- Block Jacobi preconditioning is a simple and appropriate solution for preconditioning the SP_N equations
- Implementation is general - slides right into ANA framework
- Implementation is scalable - local operations only
- Implementation works - can move on with research