Iterative Methods in TAXI (GMRES and BICGSTAB)

Tim Smith
Computer Science
May 11, 2005
Background and Motivation

- Source Iteration (SI) can be slow to converge
- TSA methods used to accelerate SI
  - $\beta$-TSA, $\varepsilon$-TSA, Stretched and Filter TSA implemented
  - SI with TSA can be slow to converge
- Using Krylov methods may improve convergence
  - Easily wrapped around existing transport methods
  - TSA can be used as a preconditioner for the solver
  - GMRES implemented by Jae Chang
Loop Construct in Infrastructure
Where Iterative Methods Fit

- We’ve chosen
  - Problem Type
    - Transport of some sort
  - Time discretization
    - May have been done
    - Not considered
  - Energy Discretization
    - Multigroup
  - Spatial discretization
    - Not considered
- Solve across groupsets
  - Iterate until solution converges in all energy groups
Solve All Energy Group Sets

- Iteration is a loop over all groupsets
Solve Each Energy Group Set

Diagram:

- Solve All Group Sets
  - Solve Each Groupset
    - Each groupset processed
    - No Prep
    - No Output

- Solve Each Groupset
  - Set sources: Scattering from other groupsets
  - Solve within groupset
Solve within Energy Group Set

- Still not committed to a spatial discretization method
- Choice of method used here can be based on characteristics of problem input
Solver Methods

- Iteration in each method is iteration to a fixed point
- BICGSTAB has two transport steps per iteration
Transport Step

Result from sweep is 
\((I - A_l^{-1} A) \cdot \text{input} \cdot A_l^{-1} \cdot q\)

Result from preconditioner is 
\((I - PA_l^{-1} A) + R_l^{-1} \cdot q\)

Precondition

Choose Method

TSA
None
Other
$x_0 = 0; r_0 = b ; \text{rhat} = r_0$

$\rho_0 = \alpha = \omega_0 = 1; v_0 = p_0 = 0; i = 0;$

While (!converged && i < maxiterations)

$\rho_i = (\text{rhat}, r_{i-1}); \beta = (\rho_i/\rho_{i-1})(\alpha/\omega_{i-1});$

$p_i = r_{i-1} + \beta(p_{i-1} - \omega_{i-1}v_{i-1});$ // find search vector

$v_i = Ap_i;$

$\alpha = \rho_i/(\text{rhat}/v_i);$

$s = r_{i-1} - \alpha v_i;$

$t = As;$

$\omega_i = (t,s)/(t,t);$  

$x_i = x_{i-1} + \alpha p_i + \omega_i s;$ // update solution

if $x_i$ is accurate enough then converged = true;

$r_i = s - \omega_i t;$

End
## Memory Usage

<table>
<thead>
<tr>
<th></th>
<th>Extra pArrays</th>
<th>Solution vector storage</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMRES</td>
<td>4</td>
<td>Restart value</td>
<td>(4 + restart) * total # elems</td>
</tr>
<tr>
<td>BICGSTAB</td>
<td>8</td>
<td>None</td>
<td>8 * total # elems</td>
</tr>
</tbody>
</table>

- BICGSTAB uses less than GMRES with restart values > 4
- This memory is in addition to SI data structures
  - Grid, Quadrature objects, etc.
## Operations

<table>
<thead>
<tr>
<th></th>
<th>Setup</th>
<th>Per Iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sweeps</td>
<td>Norms (dotprod, 2-norm)</td>
</tr>
<tr>
<td>BICGSTAB</td>
<td>1 Sweep</td>
<td>2</td>
</tr>
<tr>
<td>GMRES Every Iteration</td>
<td>1 Sweep</td>
<td>1</td>
</tr>
<tr>
<td>GMRES Every mth iteration</td>
<td>1</td>
<td>i+3</td>
</tr>
</tbody>
</table>

- i in GMRES counts is the current size of the iteration matrix
  - varies from 1 to restart value
  - BICGSTAB uses less memory than GMRES with restart value $\geq 4$
- Each sweep and norm operation uses a global barrier
Breakdown Conditions

- Restarted GMRES break down
  - Due to unavailability of previous search vectors
  - Progress towards solution with new search vectors is slowed

- BICGSTAB breaks down when rhat is poorly chosen
  - $\rho_i = (\text{rhat}, r_{i-1}) = 0$ (or very small)
  - Progress towards solution is not made.
  - May choose new rhat or use GMRES.
We test the methods on a homogenous problem.

- 100 x 100 cells
- 100 x 100 mfp
- $\sigma_t = 1.0$
- Scattering ratio = 0.99
- Incident source on the left face
- No volumetric source
- Convergence tolerance = 1.0E-07
- Diamond difference spatial discretization
## Numerical Results for BICGSTAB

<table>
<thead>
<tr>
<th>Method</th>
<th>CPU Time (seconds)</th>
<th>Iterations</th>
<th>GMRES CPU Time (seconds)</th>
<th>GMRES Iterations</th>
<th>BICGSTAB CPU Time (seconds)</th>
<th>BICGSTAB Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>SI</td>
<td>3841</td>
<td>2270</td>
<td>396</td>
<td>99</td>
<td>322</td>
<td>51</td>
</tr>
<tr>
<td>SFTSA (γ=0)</td>
<td>1320</td>
<td>182</td>
<td>482</td>
<td>50</td>
<td>427</td>
<td>24</td>
</tr>
<tr>
<td>SFTSA (γ=0.3)</td>
<td>1927</td>
<td>267</td>
<td>389</td>
<td>40</td>
<td>411</td>
<td>23</td>
</tr>
<tr>
<td>SFTSA (γ=1.0)</td>
<td>3729</td>
<td>519</td>
<td>261</td>
<td>26</td>
<td>258</td>
<td>14</td>
</tr>
<tr>
<td>SFTSA (γ=2.0)</td>
<td>3196</td>
<td>444</td>
<td>194</td>
<td>19</td>
<td>190</td>
<td>10</td>
</tr>
<tr>
<td>SFTSA (γ=3.0)</td>
<td>443</td>
<td>59</td>
<td>194</td>
<td>19</td>
<td>208</td>
<td>11</td>
</tr>
<tr>
<td>SFTSA (γ=10.0)</td>
<td>1529</td>
<td>211</td>
<td>270</td>
<td>27</td>
<td>258</td>
<td>14</td>
</tr>
</tbody>
</table>
Parallel Scalability

- Weekly Timing Problem
  - Added input section so BICGSTAB or GMRES used
  - No TSA Acceleration

- Input Specs
  - 64x64x256 cells
  - S6, 48 angles (6 angles per angleset)
  - 6 energy groups (6 groups per groupset)
  - $\sigma_t = 10, \sigma_s = 9.9$

- Experiments run on frost at LLNL
  - 16 Power3 procs @ 300MHz per node
  - 16GB memory per node
Strong Scaling Results
## Execution Times

<table>
<thead>
<tr>
<th>Method</th>
<th>Iterations</th>
<th>32</th>
<th>64</th>
<th>128</th>
<th>256</th>
</tr>
</thead>
<tbody>
<tr>
<td>SI</td>
<td>1410</td>
<td>37169</td>
<td>17297</td>
<td>9961</td>
<td>6277</td>
</tr>
<tr>
<td>GMRES10</td>
<td>74</td>
<td>12930</td>
<td>5497</td>
<td>4172</td>
<td>3777</td>
</tr>
<tr>
<td>GMRES10 CGS</td>
<td>74</td>
<td>11408</td>
<td>3883</td>
<td>2907</td>
<td>2529</td>
</tr>
<tr>
<td>BICG</td>
<td>44</td>
<td>9362</td>
<td>2757</td>
<td>1974</td>
<td>1450</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Solve Times (seconds)</th>
<th>32</th>
<th>64</th>
<th>128</th>
<th>256</th>
</tr>
</thead>
<tbody>
<tr>
<td>SI</td>
<td>36848</td>
<td>17138</td>
<td>9874</td>
<td>6224</td>
<td></td>
</tr>
<tr>
<td>GMRES10</td>
<td>12601</td>
<td>5336</td>
<td>4085</td>
<td>3723</td>
<td></td>
</tr>
<tr>
<td>GMRES10 CGS</td>
<td>11093</td>
<td>3727</td>
<td>2819</td>
<td>2475</td>
<td></td>
</tr>
<tr>
<td>BICG</td>
<td>9038</td>
<td>2596</td>
<td>1887</td>
<td>1397</td>
<td></td>
</tr>
</tbody>
</table>
Conclusions

- Implementing BICGSTAB in our code was easy
  - GMRES used as a guide (e.g. how to do matvec)
- BICGSTAB scales better than GMRES
- Iterative Methods shouldn’t use TSA if problems use low order quadratures
Future Work

- Determine when SI, GMRES, or BICGSTAB is the better method
- Identify problem attributes we can use to decide which method to use
- Adaptively choose method to use at run-time