Improving Coarsening and Interpolation for Algebraic Multigrid

Jeff Butler
Hans De Sterck
Department of Applied Mathematics

(In Collaboration with Ulrike Meier Yang, LLNL)
Outline

• Introduction: Algebraic Multigrid (AMG)
• Modification 1: PMIS Greedy vs. PMIS
• Modification 2: PMIS restricted to finer grid levels
• Modification 3: FF and FF1 Interpolation
Introduction

• Solve: $Au = f$

• $A, f$ from PDE discretization
  → sparse

• Parallel
  → large problems: $10^9$ degrees of freedom

• Unstructured grid problems
AMG Structure

Setup Phase

On each level (m):
- Select coarse grid points (coarsen)
- Define interpolation operator, $P^{(m)}$
- Define restriction and coarse-grid operators

\[ R^{(m)} = P^{(m)} T \quad A^{(m+1)} = P^{(m)} T A^{(m)} P^{(m)} \]

Solve Phase

\[
\begin{align*}
\text{Relax} & \quad A^{(m)} \hat{u}^m = f^m \\
\text{Restrict} & \quad r^{m+1} = R^{(m)} r^m \\
\text{Compute} & \quad r^m = f^m - A^{(m)} \hat{u}^m \\
\text{Solve} & \quad A^{(m+1)} e^{m+1} = r^{m+1} \\
\text{Correct} & \quad \hat{u}^m \leftarrow \hat{u}^m + e^m \\
\text{Interpolate} & \quad e^m = P^{(m)} e^{m+1}
\end{align*}
\]
AMG Complexity & Scalability

• Goal: Scalable Algorithm
  – $O(n)$ operations per V-cycle
  – $\rho_{AMG}$ independent of $n$

• Operator Complexity:
  \[ C_{op} = \frac{\sum \text{nonzeros}(A_i)}{\text{nonzeros}(A_0)} \]

• Measure of memory use, work in solve phase, and work required in coarsening and interpolation process

• E.g. 3D geometric multigrid:
  \[ C_{op} = 1 + \frac{1}{8} + \frac{1}{64} + \cdots < \frac{8}{7} \]
Classical AMG Coarsening (Ruge, Stueben)

- (H1) All F-F connections require connection to a common C-point (good nearest neighbor interpolation)

- (H2) Maximal Independent Set:
  Independent: no two C-points are connected
  Maximal: If one more C-point is added, independence is lost
Classical AMG Coarsening (Ruge, Stueben)

- (H1) All F-F connections require connection to a common C-point (good nearest neighbor interpolation)

- (H2) Maximal Independent Set:
  Independent: no two C-points are connected
  Maximal: If one more C-point is added, independence is lost

→ Enforce H1 rigorously with H2 as a guide (change F-points to C-points)
→ More C-points = higher complexity
AMG Coarsenings

1. **Ruge-Stueben (RS)**
   - two passes: second pass to ensure that F-F have a common C
   - disadvantage: highly sequential

2. **CLJP**
   - based on parallel independent set algorithms developed by Luby and later by Jones & Plassman
   - ensures that F-F have a common C

3. **PMIS (De Sterck, Yang)**
   - Parallel Modified Independent Set (PMIS) (Luby)
   - do not enforce heuristic H1 (F-F without a common C)
PMIS Selection Step 1

- select C-points with maximal local measure
- make neighbors F-points
- remove neighbor edges

* This grid animation courtesy of Ulrike Yang, LLNL
PMIS Remove and Update Step 1

- select C-points with maximal local measure
- make neighbors F-points
- remove neighbor edges

*This grid animation courtesy of Ulrike Yang, LLNL*
PMIS Selection Step 2

- select C-points with maximal local measure
- make neighbors F-points
- remove neighbor edges

* This grid animation courtesy of Ulrike Yang, LLNL
PMIS Remove and Update Step 2

- select C-points with maximal local measure
- make neighbors F-points
- remove neighbor edges

* This grid animation courtesy of Ulrike Yang, LLNL
PMIS Final Coarsening

- select C-points with maximal local measure
- make neighbors F-points
- remove neighbor edges

*This grid animation courtesy of Ulrike Yang, LLNL*
PMIS Summary (De Sterck, Yang & Heys: 2006)

- PMIS coarsening worked well for many problems (with GMRES)
- for some problems, convergence degraded compared to RS
  - decreased accuracy in interpolation due to an inadequate amount of C-points
- One solution: add C-points (RS, CLJP)
- Other possibilities:
  - modify PMIS to (hopefully) improve convergence (PMIS Greedy)
  - combine PMIS coarsening with RS/CLJP coarsening
  - use distance-two C-points as long-range interpolation for F-F connections without a common C-point (F-F interpolation)
Modification 1: PMIS Greedy vs. PMIS

- **PMIS Greedy**
  - same procedure as in PMIS
  - increase the measure of an unassigned point if it is strongly connected to a newly assigned F-point in each pass
  - helps to remove some randomness in the grid structure
  - interpolation has a chance to be more accurate

5-pt Laplacian:  
- RS = finest grid  
- PMIS = second finest grid  
- PMIS Greedy = third finest grid

![Grid Diagrams]
Results for an “Easy” Problem

- 27-pt laplacian, 1 processor, $128^3$ dof
- AMG

### AMG

<table>
<thead>
<tr>
<th></th>
<th>$C_{op}$</th>
<th># iterations</th>
<th>$t_{setup}$ (s)</th>
<th>$t_{solve}$ (s)</th>
<th>$t_{total}$ (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RS</td>
<td>2.67</td>
<td>9</td>
<td>209.88</td>
<td>25.24</td>
<td>235.12</td>
</tr>
<tr>
<td>PMIS</td>
<td>1.10</td>
<td>47</td>
<td>33.67</td>
<td>65.08</td>
<td>98.75</td>
</tr>
<tr>
<td>PMIS Greedy</td>
<td>1.12</td>
<td>38</td>
<td>36.37</td>
<td>54.01</td>
<td>90.38</td>
</tr>
</tbody>
</table>

### AMG-GMRES(5)

<table>
<thead>
<tr>
<th></th>
<th>$C_{op}$</th>
<th># iterations</th>
<th>$t_{setup}$ (s)</th>
<th>$t_{solve}$ (s)</th>
<th>$t_{total}$ (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RS</td>
<td>2.67</td>
<td>6</td>
<td>208.94</td>
<td>74.72</td>
<td>283.66</td>
</tr>
<tr>
<td>PMIS</td>
<td>1.10</td>
<td>17</td>
<td>33.72</td>
<td>42.09</td>
<td>75.81</td>
</tr>
<tr>
<td>PMIS Greedy</td>
<td>1.12</td>
<td>15</td>
<td>36.39</td>
<td>36.99</td>
<td>73.38</td>
</tr>
</tbody>
</table>
Results for a More Difficult Problem

• 3D elliptic PDE with jumps in the coefficient $a$

\[(a u_x)_x + (a u_y)_y + (a u_z)_z = 1\]

• 1 processor, $80^3$ dof

• AMG-GMRES(5)

<table>
<thead>
<tr>
<th>Method</th>
<th>$C_\text{op}$</th>
<th># iterations</th>
<th>$t_{\text{setup}}$ (s)</th>
<th>$t_{\text{solve}}$ (s)</th>
<th>$t_{\text{total}}$ (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RS</td>
<td>21.54</td>
<td></td>
<td>Ran out of Memory</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PMIS</td>
<td>2.46</td>
<td>188</td>
<td>4.06</td>
<td>77.34</td>
<td>81.40</td>
</tr>
<tr>
<td>PMIS Greedy</td>
<td>2.54</td>
<td>144</td>
<td>4.19</td>
<td>59.90</td>
<td>64.09</td>
</tr>
</tbody>
</table>
PMIS vs. PMIS Greedy Summary

- better convergence and timing for some problems
- improvement not major
- for other problems, PMIS Greedy not much better (and sometimes even worse) than PMIS with respect to convergence and timing
- slightly higher operator complexities for PMIS Greedy
- PMIS Greedy will require more communication in parallel than PMIS
Modification 2: Restrict PMIS to Finer Grid Levels

- Perform PMIS on first $g$ grid levels and RS on all remaining levels
- Recall:
  
  \[
  3D \text{ geometric multigrid: } C_{op} = 1 + \frac{1}{8} + \frac{1}{64} + \cdots < \frac{8}{7}
  \]

- Advantage:
  - PMIS reduces operator complexity (compared to RS) where it makes the biggest difference (finer levels)
  - RS produces more “structured” grids leading to better interpolation on coarser levels where impact on operator complexity is reduced
Results: 27-pt Laplacian Problem

• 1 processor, $128^3$ dof

• AMG

<table>
<thead>
<tr>
<th></th>
<th>$C_{op}$</th>
<th># iterations</th>
<th>$t_{setup}$ (s)</th>
<th>$t_{solve}$ (s)</th>
<th>$t_{total}$ (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PMIS (all levels)</td>
<td>1.10</td>
<td>47</td>
<td>33.67</td>
<td>65.08</td>
<td>98.75</td>
</tr>
<tr>
<td>PMIS (first 2 levels)</td>
<td>1.12</td>
<td>27</td>
<td>34.55</td>
<td>38.05</td>
<td>72.60</td>
</tr>
</tbody>
</table>

• AMG-GMRES(5)

<table>
<thead>
<tr>
<th></th>
<th>$C_{op}$</th>
<th># iterations</th>
<th>$t_{setup}$ (s)</th>
<th>$t_{solve}$ (s)</th>
<th>$t_{total}$ (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PMIS (all levels)</td>
<td>1.10</td>
<td>17</td>
<td>33.72</td>
<td>42.09</td>
<td>75.81</td>
</tr>
<tr>
<td>PMIS (first 2 levels)</td>
<td>1.12</td>
<td>12</td>
<td>34.58</td>
<td>30.55</td>
<td>65.13</td>
</tr>
</tbody>
</table>
Results: 3D elliptic PDE with jumps in $a$

- $(a_{ux})_x + (a_{uy})_y + (a_{uz})_z = 1$

- **AMG, 1 processor, $120^3$ dof**

<table>
<thead>
<tr>
<th></th>
<th>$C_{op}$</th>
<th># iterations</th>
<th>$t_{setup}$ (s)</th>
<th>$t_{solve}$ (s)</th>
<th>$t_{total}$ (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PMIS (all levels)</td>
<td>2.44</td>
<td>$&gt;&gt; 200$</td>
<td><strong>Slow to converge</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PMIS (first level only)</td>
<td>7.2131</td>
<td>23</td>
<td>47.93</td>
<td>44.32</td>
<td>92.25</td>
</tr>
<tr>
<td>PMIS (first 2 levels)</td>
<td>3.49</td>
<td>168</td>
<td>20.58</td>
<td>190.16</td>
<td>210.74</td>
</tr>
</tbody>
</table>

- **AMG-GMRES(5), 1 processor, $80^3$ dof**

<table>
<thead>
<tr>
<th></th>
<th>$C_{op}$</th>
<th># iterations</th>
<th>$t_{setup}$ (s)</th>
<th>$t_{solve}$ (s)</th>
<th>$t_{total}$ (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PMIS (all levels)</td>
<td>2.46</td>
<td>188</td>
<td>4.06</td>
<td>77.34</td>
<td>81.40</td>
</tr>
<tr>
<td>PMIS (first level only)</td>
<td>6.97</td>
<td>11</td>
<td>12.76</td>
<td>8.38</td>
<td>21.14</td>
</tr>
<tr>
<td>PMIS (first 3 levels)</td>
<td>2.56</td>
<td>105</td>
<td>4.24</td>
<td>44.20</td>
<td>48.44</td>
</tr>
</tbody>
</table>

* $C_{op}(RS) = 21.54$
PMIS on Higher Levels Only: Summary

• Improvement in convergence properties for a variety of problems (although some only small)

• How many levels PMIS?
  – problem dependent

• Trade-off between memory (increased operator complexity) and speed of convergence/execution

• Similar approach for parallel case is possible
  – CLJP has similar performance characteristics as RS
Modification 3: Modified FF Interpolation

- FF Interpolation* (De Sterck, Yang: Copper 2005):
  - When a strong F-F connection is encountered, do not add a C-point, but extend interpolation to distance-two C-points
  - No C-points added, but get larger interpolation stencils (and therefore somewhat increased operator complexity)

*Closely related to Stueben’s Standard Interpolation
• Modified FF Interpolation (FF1)

• To reduce operator complexity, only include one distance-two C-point when a strong FF connection is encountered

• Setup time, complexity are reduced
Results: 7-pt Laplacian Problem

- PMIS coarsening, 1 processor, $128^3$ dof
- AMG

<table>
<thead>
<tr>
<th></th>
<th>$C_{op}$</th>
<th># iterations</th>
<th>$t_{setup}$ (s)</th>
<th>$t_{solve}$ (s)</th>
<th>$t_{total}$ (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular</td>
<td>2.36</td>
<td>77</td>
<td>16.63</td>
<td>85.93</td>
<td>102.56</td>
</tr>
<tr>
<td>FF</td>
<td>4.80</td>
<td>13</td>
<td>83.81</td>
<td>22.86</td>
<td>106.67</td>
</tr>
<tr>
<td>FF1</td>
<td>3.68</td>
<td>15</td>
<td>44.22</td>
<td>22.07</td>
<td>66.29</td>
</tr>
</tbody>
</table>

- AMG-GMRES(5)

<table>
<thead>
<tr>
<th></th>
<th>$C_{op}$</th>
<th># iterations</th>
<th>$t_{setup}$ (s)</th>
<th>$t_{solve}$ (s)</th>
<th>$t_{total}$ (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular</td>
<td>2.36</td>
<td>20</td>
<td>16.67</td>
<td>35.26</td>
<td>51.93</td>
</tr>
<tr>
<td>FF</td>
<td>4.80</td>
<td>9</td>
<td>83.87</td>
<td>22.97</td>
<td>106.84</td>
</tr>
<tr>
<td>FF1</td>
<td>3.68</td>
<td>9</td>
<td>43.85</td>
<td>19.85</td>
<td>63.70</td>
</tr>
</tbody>
</table>
Results: 3D elliptic PDE with jumps in $a$

- $(au_x)_x + (au_y)_y + (au_z)_z = 1$
- **AMG, 1 processor, 120^3 dof**

<table>
<thead>
<tr>
<th></th>
<th>$C_{op}$</th>
<th># iterations</th>
<th>$t_{setup}$ (s)</th>
<th>$t_{solve}$ (s)</th>
<th>$t_{total}$ (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular</td>
<td>2.44</td>
<td>$&gt;&gt;$ 200</td>
<td>Slow to converge</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FF</td>
<td>4.94</td>
<td>14</td>
<td>62.95</td>
<td>20.54</td>
<td>83.49</td>
</tr>
<tr>
<td>FF1</td>
<td>3.84</td>
<td>18</td>
<td>35.36</td>
<td>22.47</td>
<td>57.83</td>
</tr>
</tbody>
</table>

- **AMG-GMRES(5), 1 processor, 80^3 dof**

<table>
<thead>
<tr>
<th></th>
<th>$C_{op}$</th>
<th># iterations</th>
<th>$t_{setup}$ (s)</th>
<th>$t_{solve}$ (s)</th>
<th>$t_{total}$ (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular</td>
<td>2.46</td>
<td>188</td>
<td>4.06</td>
<td>77.34</td>
<td>81.40</td>
</tr>
<tr>
<td>FF</td>
<td>4.90</td>
<td>9</td>
<td>16.46</td>
<td>5.34</td>
<td>21.80</td>
</tr>
<tr>
<td>FF1</td>
<td>3.85</td>
<td>10</td>
<td>9.67</td>
<td>4.99</td>
<td>14.66</td>
</tr>
</tbody>
</table>
Modified FF Interpolation (FF1) Summary

• PMIS with FF or FF1 is scalable! (comparable to RS with classical interpolation, but lower complexity; no need for GMRES)
• FF1 reduces setup time compared to FF
• FF1 iteration number kept low as with FF
• FF1 operator complexity reduced compared to FF
• PMIS with FF1 seems a good parallel alternative for RS with a second pass and classical interpolation
Thanks…

- Ulrike Meier Yang and the Center for Applied Scientific Computing, Lawrence Livermore National Laboratory

- travel support University of Waterloo Graduate Studies
Algebraic Multigrid (AMG)

- Multiple levels (coarsen to some minimum)
- Iterative
- AMG suitable for unstructured grids
PMIS Coarsening

- weighted independent set algorithm:
  - points, i, that influence many equations ($\lambda_i$ large) are good candidates for C-points
- add random number between 0 and 1 to $\lambda_i$ to break ties
- pick C-points with maximal measures (like in CLJP), then make all their neighbors fine points (like in RS)
- proceed until all points have been assigned as fine or coarse
Conclusions

• PMIS vs. PMIS Greedy coarsening
  – PMIS Greedy does not provide a considerable advantage
  – any advantage will be degraded in a parallel implementation

• Restricting PMIS to finer grid levels
  – considerable improvement over uniform RS and PMIS coarsening possible
  – scalability possible
  – trade-off between memory use and convergence speed

• FF1 Interpolation (with PMIS coarsening)
  – comparable performance to FF (both better than regular interpolation)
  – scalability possible
  – better suited to parallel implementation than FF