Coarsening and Interpolation in Algebraic Multigrid: a Balancing Act

Hans De Sterck
Department of Applied Mathematics
University of Waterloo, Ontario, Canada

Ulrike Meier Yang
Center for Applied Scientific Computing
Lawrence Livermore National Laboratory
California, USA
Outline

- introduction: AMG
- AMG coarsening: classical versus more aggressive
- convergence problems with more aggressively coarsened grids
- improved, long-range interpolation methods
- results
- conclusions and future work
Introduction

- solve $Au = f$
- $A$ from 3D PDE – sparse!
- large problems ($10^9$ dof) - parallel
- unstructured grid problems
Algebraic Multigrid (AMG)

- *multi-level*
- *iterative*
- *algebraic: suitable for unstructured*
AMG complexity - scalability

- **scalable algorithm:**
  
  \[ O(n) \text{ operations per V-cycle } (C_{op} \text{ bounded}) \]
  
  AND
  
  \[ \text{number of V-cycles independent of } n \]
  
  \[ (\rho_{AMG} \text{ independent of } n) \]

- **Operator complexity**
  
  \[ C_{op} = \frac{\sum \text{nonzeros}(A_i)}{\text{nonzeros}(A_0)} \]

- e.g., 3D:
  
  \[ C_{op} = 1 + 1/8 + 1/64 + \ldots < 8/7 \]

  measure of memory use, and work in solve phase
Classical AMG coarsening

- **(C1)** Maximal Independent Set:
  - Independent: no two \( C \)-points are connected
  - Maximal: if one more \( C \)-point is added, the independence is lost

- **(C2)** All \( F \)-\( F \) connections require connections to a common \( C \)-point (for good nearest-neighbor interpolation)
  - \( F \)-points have to be changed into \( C \)-points, to ensure (C2); (C1) is violated
  - more \( C \)-points, higher complexity
Classical coarsening: scalability results

- **example:** finite difference Laplacian, parallel CLJP coarsening algorithm

- **2D (5-point): near-optimal scalability** \((250^2 \text{ dof/proc})\)

<table>
<thead>
<tr>
<th>Procs</th>
<th>(C_{op})</th>
<th>(t_{tot})</th>
<th>Iter</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>4.48</td>
<td>2.89</td>
<td>9</td>
</tr>
<tr>
<td>64</td>
<td>4.50</td>
<td>3.85</td>
<td>9</td>
</tr>
<tr>
<td>256</td>
<td>4.50</td>
<td>5.01</td>
<td>9</td>
</tr>
</tbody>
</table>
Classical coarsening: complexity growth in some cases

- **3D (7-point): complexity growth**

<table>
<thead>
<tr>
<th>dof</th>
<th>$C_{op}$</th>
<th>Iter</th>
</tr>
</thead>
<tbody>
<tr>
<td>$32^3$</td>
<td>16.17</td>
<td>8</td>
</tr>
<tr>
<td>$64^3$</td>
<td>22.51</td>
<td>11</td>
</tr>
</tbody>
</table>

- **increased memory use, long solution times, long setup times, loss of scalability**
Classical coarsening: complexity growth in some cases

- 4D (9-point), 5D (11-point): complexity growth!!

<table>
<thead>
<tr>
<th></th>
<th>dof</th>
<th>(C_{op})</th>
<th>Iter</th>
</tr>
</thead>
<tbody>
<tr>
<td>4D</td>
<td>(20^4)</td>
<td>127.5</td>
<td>8</td>
</tr>
<tr>
<td>5D</td>
<td>(9^5)</td>
<td>256.9</td>
<td>5</td>
</tr>
</tbody>
</table>

- excessive memory use
our approach to reduce complexity: PMIS (parallel modified independent set)

- do not add C points for strong F-F connections that do not have a common C point

- less C points, reduced complexity, but worse convergence factors expected

- combine with GMRES acceleration

- in many cases (3D…), large gains
PMIS coarsening: reduce complexity

*finite difference Laplacian (CLJP-PMIS+GMRES)*

<table>
<thead>
<tr>
<th></th>
<th>dof</th>
<th>$C_{op}$</th>
<th>Iter</th>
<th>$t_{tot}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D</td>
<td>$120^2$</td>
<td>4.16</td>
<td>12</td>
<td>0.22</td>
</tr>
<tr>
<td></td>
<td>$120^2$</td>
<td>1.90</td>
<td>24</td>
<td>0.24</td>
</tr>
<tr>
<td>3D</td>
<td>$100^3$</td>
<td>25.94</td>
<td>12</td>
<td>129.42</td>
</tr>
<tr>
<td></td>
<td>$100^3$</td>
<td>2.36</td>
<td>20</td>
<td>27.68</td>
</tr>
<tr>
<td>4D</td>
<td>$20^4$</td>
<td>127.5</td>
<td>8</td>
<td>88.39</td>
</tr>
<tr>
<td></td>
<td>$20^4$</td>
<td>2.95</td>
<td>11</td>
<td>4.31</td>
</tr>
<tr>
<td>5D</td>
<td>$9^5$</td>
<td>256.9</td>
<td>5</td>
<td>73.92</td>
</tr>
<tr>
<td></td>
<td>$8^5$</td>
<td>3.14</td>
<td>8</td>
<td>0.91</td>
</tr>
<tr>
<td></td>
<td>$20^5$</td>
<td>4.02</td>
<td>12</td>
<td>181.93</td>
</tr>
</tbody>
</table>
Convergence problems on PMIS-coarsened grids

- PMIS coarsening works well for many problems
- for some problems, too many iterations are necessary because interpolation is not accurate enough ("not enough C-points")
- one solution: add C-points (CLJP…)
- other solution: use distance-two C-points for interpolation = long-range interpolation
  - Stuebe’s multipass interpolation
  - F-F interpolation
Convergence problems

- 3D elliptic PDE with jumps in coefficient $a$
  
  \[(au_x)_x + (au_y)_y + (au_z)_z = 1\]

- 1000 processors, $40^3$ dof/proc

<table>
<thead>
<tr>
<th></th>
<th>$t_{tot}$</th>
<th>$C_{op}$</th>
<th>Iter</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLJP</td>
<td>52.48</td>
<td>17.00</td>
<td>17</td>
</tr>
<tr>
<td>PMIS</td>
<td>211.79</td>
<td>2.40</td>
<td>686</td>
</tr>
</tbody>
</table>

- remedy: improve interpolation used with PMIS
classical AMG Interpolation

- after relaxation:

\[ A e \approx 0 \text{ (relative to } e) \]

- heuristic: error after interpolation should also satisfy this relation approximately

- derive interpolation from:

\[
a_{ii} e_i + \sum_{j \in C} a_{ij} e_j + \sum_{j \in F} a_{ij} e_j = 0 \quad \forall i \in F
\]
classical AMG interpolation

\[ a_{ii} e_i + \sum_{j \in C} a_{ij} e_j + \sum_{j \in F} a_{ij} e_j = 0 \quad \forall i \in F \]

- “large” \( a_{ij} \) should be taken into account accurately
- “strong connections”: \( i \) strongly depends on \( j \) (and \( j \) strongly influences \( i \)) if
  \[ -a_{ij} \geq \theta \max_{k \neq i} \{-a_{ik}\}, \quad 0 < \theta \leq 1 \]

with strong threshold \( \theta \)
classical AMG interpolation

- strong F-F connections interpolated from common C-point
- interpolation only from nearest-neighbor C-points
(1) Stueben’s multipass interpolation

1st pass:
Coarse points
Multipass interpolation

2\textsuperscript{nd} pass:

direct interpolation from coarse C-neighbor
Multipass interpolation

2\textsuperscript{nd} pass:

direct interpolation from coarse C-neighbor
Multipass interpolation

3rd pass:
direct interpolation from coarse F-neighbor
(indirectly from distance-2 C-point)
Multipass interpolation

3rd pass:

direct interpolation from coarse F-neighbor (indirectly from distance-2 C-point)
Multipass interpolation

Final pass
Multipass interpolation

Final pass
(2) F-F interpolation

- when strong F-F connection without a common C-point is detected, do not add C-point, but extend interpolation stencil to distance-two C-points
- no C-points added, but larger interpolation stencils
results using long-range interpolation

- **3D elliptic PDE with jumps in coefficient** $a$
  \[
  (a u_x)_x + (a u_y)_y + (a u_z)_z = 1
  \]

- **1 processor, AMG+GMRES, $80^3$ dof**

<table>
<thead>
<tr>
<th></th>
<th>$t_{tot}$</th>
<th>$C_{op}$</th>
<th>$s_{avg}$ (level)</th>
<th>Iter</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLJP</td>
<td>48.0</td>
<td>21.54</td>
<td>1007 (9)</td>
<td>7</td>
</tr>
<tr>
<td>PMIS</td>
<td>94.6</td>
<td>2.46</td>
<td>54 (3)</td>
<td>188</td>
</tr>
<tr>
<td>PMIS + mp</td>
<td>13.7</td>
<td>2.47</td>
<td>56 (3)</td>
<td>21</td>
</tr>
<tr>
<td>PMIS + F-F</td>
<td>21.4</td>
<td>4.90</td>
<td>204 (3)</td>
<td>9</td>
</tr>
</tbody>
</table>
results using long-range interpolation

- 3D elliptic PDE with jumps in coefficient \( a \)
- 1 processor, AMG+GMRES

<table>
<thead>
<tr>
<th>dof</th>
<th>( C_{op} )</th>
<th>( s_{avg} )</th>
<th>Iter</th>
<th>( t_{setup} )</th>
<th>( t_{solve} )</th>
<th>( t_{tot} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>PMIS + mp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>2.53</td>
<td>44</td>
<td>17</td>
<td>0.33</td>
<td>0.98</td>
<td>1.31</td>
</tr>
<tr>
<td>80</td>
<td>2.47</td>
<td>56</td>
<td>21</td>
<td>3.11</td>
<td>10.55</td>
<td>13.66</td>
</tr>
<tr>
<td>120</td>
<td>2.44</td>
<td>59</td>
<td>26</td>
<td>10.98</td>
<td>46.84</td>
<td>57.82</td>
</tr>
<tr>
<td>PMIS + F-F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>4.64</td>
<td>114</td>
<td>9</td>
<td>1.31</td>
<td>0.70</td>
<td>2.01</td>
</tr>
<tr>
<td>80</td>
<td>4.90</td>
<td>204</td>
<td>9</td>
<td>15.06</td>
<td>6.38</td>
<td>21.44</td>
</tr>
<tr>
<td>120</td>
<td>4.94</td>
<td>248</td>
<td>9</td>
<td>55.47</td>
<td>22.94</td>
<td>78.41</td>
</tr>
</tbody>
</table>

- \( mp \) uses less memory, is faster than F-F
Conclusions

- **PMIS leads to reduced, scalable complexities for large problems on parallel computers**

- **for difficult problems, nearest-neighbor interpolation is not sufficient on PMIS grids**

- **long-range interpolation improves convergence**

- **multipass appears superior to F-F**
Future work

- parallel implementation of multipass interpolation

- investigate scalability of parallel AMG algorithms on Blue Gene/L-class machines