Multigrid for Finite Elements

Finite Elements
- Numerical discretization of PDEs
- Advantage: support of unstructured meshes

Multigrid
- Solve a PDE using a hierarchy of discretizations with varying mesh width.
  - Advantage: time to solution linear in number of unknowns. Multigrid is an asymptotic optimal solver!
  - Problem when combining with Finite Elements: hierarchy of coarser meshes not easy to construct from unstructured fine mesh.

HHG: Hierarchical Hybrid Grids

Strategy
- Use unstructured coarse mesh (3D) as input and refine it in a structured way.
  - Multigrid is straightforward.
  - Structured regions allow for vectorized implementation and, thus, very fast execution on current hardware.
  - Structured regions are memory efficient, so that very large problems can be solved (10^11 unknowns).

Parallel Efficiency

<table>
<thead>
<tr>
<th>Cores</th>
<th>Unknowns</th>
<th>Time per V-cycle (s)</th>
<th>Phase 1</th>
<th>Phase 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>133 432 830</td>
<td>3.16</td>
<td>6.38*</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>267 126 780</td>
<td>3.27</td>
<td>6.67*</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>534 514 680</td>
<td>3.35</td>
<td>6.75*</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>1 069 290 480</td>
<td>3.38</td>
<td>6.80*</td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>2 140 673 490</td>
<td>3.53</td>
<td>4.90</td>
<td></td>
</tr>
<tr>
<td>128</td>
<td>4 283 439 510</td>
<td>3.60</td>
<td>7.06*</td>
<td></td>
</tr>
<tr>
<td>256</td>
<td>8 443 205 295</td>
<td>3.87</td>
<td>7.39*</td>
<td></td>
</tr>
<tr>
<td>504</td>
<td>16 890 993 705</td>
<td>3.96</td>
<td>5.44</td>
<td></td>
</tr>
<tr>
<td>2040</td>
<td>68 400 754 095</td>
<td>4.92</td>
<td>5.60</td>
<td></td>
</tr>
<tr>
<td>3825</td>
<td>128 268 413 349</td>
<td>6.90</td>
<td>7.75*</td>
<td></td>
</tr>
<tr>
<td>4080</td>
<td>136 819 385 295</td>
<td>5.68</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6120</td>
<td>205 245 783 375</td>
<td>6.33</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8152</td>
<td>273 000 685 365</td>
<td>7.43*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9170</td>
<td>307 476 954 685</td>
<td>7.75*</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Timings marked with * are from quad-core nodes, all other timings are from dual-core nodes.

\( \tau \)-Extrapolation

Motivation
- How to raise the order of consistency of linear Finite Elements to \( O(h^4) \)?
  - PDE with its exact solution restricted to a grid:
    \[ R(N(u)) = f \]
    \[ \Rightarrow N_k(\hat{R}(u)) = f_k + N_k(\hat{R}(u)) - R(N(u)) \]
    \[ \text{local truncation error } \tau_k \]

  - \( N \) ... Differential operator
  - \( k \) ... Finest grid
  - \( u \) ... Solution
  - \( k-1 \) ... Coarser grid
  - \( \tilde{u} \) ... Approximation
  - \( \tilde{R}, \hat{R} \) ... Linear, injective restriction

1) Estimating the local truncation error at the coarse grid:
   \[ t_{k-1} \approx N_{k-1}(\hat{R}(\tilde{u}_k)) - R(N_i(\tilde{u}_k)) \]

2) Calculate the defect with the estimated local truncation error on the right hand side.

3) Correct the solution on the finest grid by this defect.
- Main difference: Using a Full Approximation Scheme (FAS) instead of a Correction Scheme (CS).

Numerical Experiment

Poisson's problem:

\[ -\Delta u = f \text{ on } \Omega \]

Boundary conditions (Dirichlet) and solution of a model problem:

\[ u = e^{-3x(2x^2+6.5x^2(2y^2)+(3y-1.05\cos(2x^2))^2)} \]

\[ + e^{-3x(2x^2+6.5x^2(2z^2)+(3z-1.05\cos(2z^2))^2)} \]

Error reduction with decreasing mesh size:

Semi-structured domain: