High Performance Multigrid on Current Large Scale Parallel Computers

Tobias Gradl, Ulrich Rüde

Lehrstuhl für Systemsimulation
Universität Erlangen-Nürnberg

2008-02-26
Outline
Multigrid
The HHG Framework
Performance
Adaptive Mesh Refinement

Multigrid

The HHG Framework

Performance

Adaptive Mesh Refinement
What is Multigrid?

- Has nothing to do with Grid computing

- General methodology: respect the different scales of a problem

- Useful e.g. for solving elliptic PDEs
  - convergence rate independent of problem size
  - asymptotically optimal complexity $\Rightarrow$ algorithmic scalability
  - efficient parallelization — if one knows how to do it
Multigrid: V-Cycle

Goal: solve $A^h u^h = f^h$ using a hierarchy of grids

- **Relax** on $A^h u^h = f^h$
- **Residual** $r^h = f^h - A^h u^h$
- **Restrict** $f^H = I_H^h r^h$
- **Solve** $A^H u^H = f^H$ by recursion
- **Interpolate** $e^h = I_H^h u^H$
- **Correct** $u^h \leftarrow u^h + e^h$
Combining FE and MG: not straightforward

FE: mesh may be unstructured.
MG: what nodes to remove for coarsening?
Combining FE and MG: not straightforward

FE: mesh may be unstructured.
MG: what nodes to remove for coarsening?

*Hierarchical Hybrid Grids* start with the coarse grid!
Combining FE and MG: not straightforward

FE: mesh may be unstructured.
MG: what nodes to remove for coarsening?

*Hierarchical Hybrid Grids* start with the coarse grid!
Combining FE and MG: not straightforward

FE: mesh may be unstructured.
MG: what nodes to remove for coarsening?

*Hierarchical Hybrid Grids* start with the coarse grid!
Combining FE and MG: not straightforward

FE: mesh may be unstructured.
MG: what nodes to remove for coarsening?

*Hierarchical Hybrid Grids* start with the coarse grid!
Combining FE and MG: not straightforward

FE: mesh may be unstructured.
MG: what nodes to remove for coarsening?

*Hierarchical Hybrid Grids* start with the coarse grid!
Combining FE and MG: not straightforward

FE: mesh may be unstructured.
MG: what nodes to remove for coarsening?

*Hierarchical Hybrid Grids* start with the coarse grid!

⇒ same stencil for all points within a patch
Combining FE and MG: not straightforward

FE: mesh may be unstructured.
MG: what nodes to remove for coarsening?

Hierarchical Hybrid Grids start with the coarse grid!

⇒ same stencil for all points within a patch
HHG properties

Advantages

- Multigrid is straightforward
- Very memory efficient
  $10^{11}$ unknowns are possible
- Very fast

Limitation

- Coarse input mesh needed
HHG on parallel computers

Mesh is split up at **coarsest level**
→ Vertices, Edges, Faces, Volumes

Facilitates parallelization for **message passing** infrastructures
(distributed memory parallel computers)
The HHG Framework

Performance

Adaptive Mesh Refinement

Testing on HLRB II

HLRB II at Leibniz-Rechenzentrum München

- 9728 CPUs (1.6 GHz Intel Itanium 2)
- 56.5 Tflop/s peak performance (rank 15 on TOP500 list)
- 38 Tbytes of main memory
## Performance on HLRB II

| Processors | Unknowns ($\times 10^6$) | Avg. time per V-cycle (sec) | Time to solution ($|| r || < 10^{-6} \cdot || r_0 ||$) |
|------------|-------------------------|----------------------------|-----------------------------------------------|
| 64         | 2 147.5                 | 4.93                       | 59.2                                          |
| 504        | 16 911.4                | 5.44                       | 65.3                                          |
| 2040       | 68 451.0                | 5.60                       | 67.2                                          |
| 4080       | 136 902.1               | 5.68                       | 68.2                                          |
| 6120       | 205 353.1               | 6.33                       | 76.0                                          |
| 8152       | 273 535.7               | 7.43 *                     | 89.2                                          |
| 9170       | 307 694.1               | 7.75 *                     | 93.0                                          |

*: including high-density partitions

Exploring other computer architectures

DEISA*: Access to Europe’s largest computers, support with “enabling work”.

Project: Test HHG on a variety of architectures
- Summer 2008
- 100,000 CPU hours
- Combine with application

* Distributed European Infrastructure for Supercomputing Applications
Adaptive refinement: motivation

Varying mesh density required by

- geometry (e.g. walls of a room)
Adaptive refinement: motivation

Varying mesh density required by

- geometry (e.g. walls of a room)
- physics (e.g. singularities)
Two approaches

- Red-green → conforming grids
Two approaches

- Red-green → conforming grids

Tobias Gradl, Ulrich Rüde

High Performance Multigrid on Current Large Scale Parallel Computers
Two approaches

- Red-green → conforming grids
Two approaches

- Red-green → conforming grids

- Hanging nodes → non-conforming grids
Two approaches

- Red-green → conforming grids
  - Diagram showing conforming grids with red and green lines.

- Hanging nodes → non-conforming grids
  - Diagram showing non-conforming grids with hanging nodes.
Comparison: red-green vs. hanging nodes

Both have (dis-)advantages (some of them HHG specific):

- Red-green: impossible for purely quadrilateral/hexahedral grids
- Hanging nodes: too large refined areas
- Red-green: more elements on coarsest level

The ideal HHG grid has few coarse level elements.
⇒ Use refinement with hanging nodes whenever possible.
Mathematical foundation

Unknows:

\[(u^h)_i = (l^h_i u^h_i)_i \quad \text{for } h < h_i, \quad i = 1..n\]

Residual:

\[A^H u^H = f^H \quad \text{solved up to discretization error} \quad \iff \quad r^H = 0\]

\[r^h = f^h - A^h u^h = f^h - A^h (l^h_H u^H)\]

\[r^H = l^H r^h = 0\]
Refinement with hanging nodes in HHG

\[ Au = f, \quad r = f - Au, \quad \text{compact basis functions} \]
Refinement with hanging nodes in HHG

\[ Au = f, \ r = f - Au, \] compact basis functions

Uniform refinement: only one boundary layer
Refinement with hanging nodes in HHG

\[ Au = f, \quad r = f - Au, \] compact basis functions

Uniform refinement: only one boundary layer

Adaptive refinement: two boundary layers
Refinement with hanging nodes in HHG

\[ Au = f, \ r = f - Au, \] compact basis functions

Uniform refinement: only one boundary layer

Adaptive refinement: two boundary layers

1. Smooth \( u \)
Refinement with hanging nodes in HHG

\[ Au = f, \quad r = f - Au, \] compact basis functions

Uniform refinement: only one boundary layer

Adaptive refinement: two boundary layers

1. Smooth \( u \)
2. Compute & restrict \( r \)
Implementation

Become flexible, but stay fast.

- Preserve structured regions ...or at least...
- Treat unstructured regions efficiently.
- Avoid additional communication ...or at least...
- Preserve communication locality.

Work in progress...
Thank you for your attention!
Any questions?