Hierarchical Hybrid Grids

IDK Summer School 2012

Björn Gmeiner, Ulrich Rüde

July, 2012
Contents

Mantle convection

Hierarchical Hybrid Grids

Smoothers

Geometric approximation

Performance modeling
Mantle convection
Simulation of seismic velocities

Fig. by Schuberth and Moder - Simulation with TERRA (LMU)
Boussinesq model for mantle convection problems

derived from the equations for balance of forces, conservation of mass and energy:

\[-\nabla \cdot (2\eta \varepsilon(u)) + \nabla p = \rho(T)g,\]
\[\nabla \cdot u = 0,\]
\[\frac{\partial T}{\partial t} + u \cdot \nabla T - \nabla \cdot (\kappa \nabla T) = \gamma.\]

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>u</td>
<td>velocity</td>
</tr>
<tr>
<td>p</td>
<td>dynamic pressure</td>
</tr>
<tr>
<td>T</td>
<td>temperature</td>
</tr>
<tr>
<td>\nu</td>
<td>viscosity of the material</td>
</tr>
<tr>
<td>\varepsilon(u) = \frac{1}{2}(\nabla u + (\nabla u)^T)</td>
<td>strain rate tensor</td>
</tr>
<tr>
<td>\rho</td>
<td>density</td>
</tr>
<tr>
<td>\kappa, \gamma, g</td>
<td>thermal conductivity, heat sources, gravity vector</td>
</tr>
</tbody>
</table>
Discretization

Temporal for the temperature (explicit):
- Modified Euler
- BDF-2 scheme

Spatial with FE for the Stokes system:
- Add an artificial compressibility term
- Choose a pair of FE spaces that satisfy the LBB condition:
  \[ Q_{q+1}^d \times Q_q, \ (q \geq 1) \] - Taylor-Hood elements
  \[ Q_{q+1}^d \times P_{-q}, \ (q \geq 0) \] - discontinuous elements
Stokes equation

Weak form:

\[
(\epsilon(\varphi_i^u), 2\eta\epsilon(u_h^n)) - (\nabla \cdot \varphi_i^u, p^n_h) = (\varphi_i^u, \rho(T^n_h)g),
\]

\[
(\varphi_i^p, \nabla \cdot u_h^n) = 0.
\]

In matrix notation (saddle point structure with zero diagonal block):

\[
\begin{pmatrix}
A & B^T \\
B & 0
\end{pmatrix}
\begin{pmatrix}
U^n \\
P^n
\end{pmatrix}
= \begin{pmatrix}
F^n \\
0
\end{pmatrix}
\]

Pressure correction approach:

\[
\begin{pmatrix}
A & B^T \\
0 & -BA^{-1}B^T
\end{pmatrix}
\begin{pmatrix}
U^n \\
P^n
\end{pmatrix}
= \begin{pmatrix}
F^n \\
0
\end{pmatrix}
\]

\[A_{ij} = (\epsilon(\varphi_i^u), 2\eta\epsilon(\varphi_j^u))\]

\[B_{ij} = (\varphi_i^q, \nabla \cdot \varphi_j^u)\]

\[F_i^n = (\varphi_i^u, \rho(T^n)g)\]
Boussinesq model for mantle convection problems

\[-\nabla \cdot (2\eta \varepsilon (\mathbf{u})) + \nabla p = \rho (T) g,\]
\[\nabla \cdot \mathbf{u} = 0,\]
\[\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T - \nabla \cdot (\kappa \nabla T) = \gamma.\]

Additional challenges:

- Viscosity \(\eta\) variations by four orders of magnitude in radial direction
- High main memory consumption due to the operators
- Operator evaluations of the Stokes equation are the performance critical parts of a (forward) simulation
Viscosity profiles

(Fig. provided by M. Mohr)
Two-grid cycle (correction scheme)

1) Presmoothing
2) Residual
3) Restriction of the residual
4) Solving the error equation
5) Prolongation of the error
6) Correction
7) Postsmoothing

Level k

Level k-1
Viscosity $\eta$ variations by four orders of magnitude in radial direction:

- Adjust smoother
- Use matrix-dependent transfer operators
- Change coarse-grid operator

High main memory consumption due to the operators:

- Matrix-free operator evaluations

Operator evaluations of the Stokes equation are the performance critical parts of a (forward) simulation:

- Avoid indirect addressing by (semi-) structured meshes
Hierarchical Hybrid Grids
HHG - Combining finite element and multigrid methods

FE mesh may be unstructured.
What nodes to remove for coarsening? Not straightforward!
Why not start from the coarse grid?

The Hierarchical Hybrid Grids (HHG) concept

- Benjamin Bergen*: prototype
- Tobias Gradl: tuning, extensions and adaptivity

HHG mesh decomposition into primitives (2d-example)
Regular refinement of a tetrahedral element
Smoothers
Operator: 15-point stencil in a refined tetrahedral element
Different Tetrahedral Shapes

(a) Needle  
(b) Wedge  
(c) Spindle  
(d) Spade  
(e) Sliver  
(f) Cap

**Figure:** Different classes of degenerated elements
Comparison between local Fourier analysis (LFA) and HHG

<table>
<thead>
<tr>
<th></th>
<th>Regular</th>
<th>Needle</th>
<th>Wedge</th>
<th>Spindle</th>
<th>Spade</th>
<th>Sliver</th>
<th>Cap</th>
</tr>
</thead>
<tbody>
<tr>
<td>LFA</td>
<td>0.250</td>
<td>0.982</td>
<td>0.786</td>
<td>0.980</td>
<td>0.590</td>
<td>0.896</td>
<td>0.872</td>
</tr>
<tr>
<td>HHG</td>
<td>0.195</td>
<td>0.982</td>
<td>0.740</td>
<td>0.980</td>
<td>0.500</td>
<td>0.889</td>
<td>0.739</td>
</tr>
<tr>
<td>edge ratio</td>
<td>1.0</td>
<td>10.0</td>
<td>4.0</td>
<td>10.0</td>
<td>1.67</td>
<td>1.40</td>
<td>1.71</td>
</tr>
<tr>
<td>max angle</td>
<td>60°</td>
<td>87.1°</td>
<td>82.8°</td>
<td>87.1°</td>
<td>112.9°</td>
<td>88.9°</td>
<td>117.2°</td>
</tr>
<tr>
<td>min angle</td>
<td>60°</td>
<td>5.7°</td>
<td>14.3°</td>
<td>5.7°</td>
<td>33.6°</td>
<td>45.6°</td>
<td>31.4°</td>
</tr>
</tbody>
</table>

Table: LFA smoothing factors, $\mu^{\nu_1+\nu_2}$, and two-grid convergence factors, $\rho$, of the four-color smoother for different tetrahedra.

Results of the LFA are provided by Francisco Gaspar.
(a) Wedge

\( \nu(1, 1) \) four-color Gauss-Seidel: 0.786
\( \nu(1, 1) \) line-wise smoothing: 0.122

(b) Spade

\( \nu(1, 1) \) four-color Gauss-Seidel: 0.590
\( \nu(1, 1) \) line-wise smoothing: 0.230
\( \nu(1, 0) \) zebra-plane smoothing: 0.105
Mantle convection  
Hierarchical Hybrid Grids  
Smoothers  
Geometric approximation  
Performance modeling

(c) Needle

(d) Spindle

(c) $\nu(1, 1)$ four-color Gauss-Seidel: 0.982
$\nu(1, 0)$ lex. plane smoothing: 0.330
$\nu(1, 0)$ zebra-plane smoothing: 0.121

(d) $\nu(1, 1)$ four-color Gauss-Seidel: 0.980
$\nu(1, 0)$ zebra-plane smoothing: 0.124
(e) Sliver

(f) Cap

(e), (f) ???
**But:** Acc. to Shewchuk: "good" tetrahedra are of two types: those that are not flat, and those that can grow arbitrarily flat without having a large planar or dihedral angle. The "bad" tetrahedra have error bounds that explode, and a dihedral angle or a planar angle that approaches 180°, as they are flattened.

putting blocks together...

(coarsest mesh is generated by Gmsh)
Locally adaptive smoothing

We can predict the convergence rate for the single blocks of our domain by LFA.

How to improve the convergence of the whole domain?
We can predict the convergence rate for the single blocks of our domain by LFA.

How to improve the convergence of the whole domain?

Choose for each block:

- Different smoothers
- Optimal relaxation parameters
- Suitable number of pre-/post smoothing steps
Locally adaptive smoothing

We can predict the convergence rate for the single blocks of our domain by LFA.

How to improve the convergence of the whole domain?

Choose for each block part:

- Different smoothers
- Optimal relaxation parameters
- Suitable number of pre-/post smoothing steps
Chaotic smoothing (mixed smoothers)

**Figure:** Domain consisting of wedge, regular and needle tetrahedral types

<table>
<thead>
<tr>
<th>Shape</th>
<th>edge ratio</th>
<th>max angle</th>
<th>min angle</th>
<th>smoothing</th>
<th>$\nu_1$, $\nu_2$</th>
<th>$\rho$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Needle</td>
<td>10</td>
<td>87$^0$</td>
<td>5.7$^0$</td>
<td>plane-wise</td>
<td>1.0</td>
<td>0.12</td>
</tr>
<tr>
<td>Regular</td>
<td>1</td>
<td>60$^0$</td>
<td>60$^0$</td>
<td>4-color</td>
<td>2.1</td>
<td>0.09</td>
</tr>
<tr>
<td>Wedge</td>
<td>5</td>
<td>110$^0$</td>
<td>11$^0$</td>
<td>line-wise</td>
<td>2.1</td>
<td>0.09</td>
</tr>
</tbody>
</table>

**Table:** Convergence factor for the single blocks

**Global** convergence factor for the whole domain: 0.17
- using damping parameter at the interface: 0.14
Anisotropic smoothing

(a) Connection in isotropic direction

(b) Connection in anisotropic direction

Gauss-Seidel: 0.934
single region, line-wise smoother: 0.072

(a) line-wise smoother: 0.094
(b) line-wise smoother: 0.847
Locally adaptive smoothing

We can predict the convergence rate for the single blocks of our domain by LFA.

How to improve the convergence of the whole domain?

Choose for each block:

- Different smoothers
- Optimal relaxation parameters
- Suitable number of pre-/post smoothing steps
Redistributing the number of smoothing steps

- Elements colored by convergence rates
- $\nu$ (pre-smoothing steps, post-smoothing steps)
- Keeping the total number of smoothing steps constant
Model Domain: Box in a Half Sphere

- Coarsest mesh: 293 elements
- 8 times refined
- $1.02 \cdot 10^8$ unknowns
- lex. Gauss-Seidel
- $\beta_{min} = 0.241$, $\beta_{avg} = 0.537$

Please note that this is a serial run!
Model Domain: Box in a Half Sphere

We set up two conditions:

1. The required amount of communication is the same for all following methods. We are exchanging eight times all ghost boundary points at the interfaces during the smoothing procedure.

2. The total smoothing workload has to be comparable to the unoptimized version in the solving phase. The additional time for the setup phase strongly depends on the implementation of a LFA evaluation in some of the following smoothing strategies.
# Local Adaptive Smoothing

<table>
<thead>
<tr>
<th>Smoothing strategy</th>
<th>W-cycle</th>
<th>V-cycle</th>
<th>( \omega )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unoptimized</td>
<td>0.64</td>
<td>0.65</td>
<td>1.0</td>
</tr>
<tr>
<td>Exact block-wise solving</td>
<td>0.11</td>
<td>0.18</td>
<td>-</td>
</tr>
<tr>
<td>Adaptive smoothing steps</td>
<td>0.51</td>
<td>0.53</td>
<td>1.0</td>
</tr>
<tr>
<td>+ Additional interface smoothing</td>
<td>0.34</td>
<td>0.34</td>
<td>1.0</td>
</tr>
<tr>
<td>Full damping</td>
<td>0.59</td>
<td>0.56</td>
<td>1.15</td>
</tr>
<tr>
<td>Interior damping</td>
<td>0.44</td>
<td>0.49</td>
<td>1.55 / 1.0</td>
</tr>
<tr>
<td>+ Additional interface smoothing</td>
<td>0.30</td>
<td>0.30</td>
<td>1.55 / 1.0</td>
</tr>
<tr>
<td>Adaptive interior damping (simplex)</td>
<td>0.46</td>
<td>0.51</td>
<td>variable</td>
</tr>
<tr>
<td>+ Additional interface smoothing</td>
<td>0.31</td>
<td>0.34</td>
<td>variable</td>
</tr>
<tr>
<td>Combined Methods</td>
<td>0.15</td>
<td>0.19</td>
<td>1.55 / 1.0</td>
</tr>
</tbody>
</table>

**Table:** Measured convergence factors for different smoothing strategies
Geometric approximation
Spherical refinement of an icosahedron (0)
Discretization with prismatic elements
Decomposition of prisms into three tetrahedra
Non-conforming decomposition of prisms into tetrahedra
Spherical refinement of an icosahedron (0)
Spherical refinement of an icosahedron (1)
Spherical refinement of an icosahedron (2)
Spherical refinement of an icosahedron (3)
Spherical refinement of an icosahedron (4)
Intersection through the spherical refinement

- Each tetrahedral element (≈ 132k) was assigned to one Jugene compute core and further refined without curved boundaries.
Performance modeling
Storage of the variable viscosity field

Since we store the viscosity like a variable, there are different ways to store them for linear finite elements:

**Element-wise**

- More storage necessary for tetrahedral elements
- Clear location of a jump when dealing with jumping coefficients

**Node-wise**

- Requires averaging
- No additional data layout necessary for structured tetrahedral elements
Viscosity averaging for a 15-point stencil

Coefficient averaging for each element

- Naive: $24 \text{ elements} \cdot 3 = 72 \text{ adds}$
- By partial sums: $16 \text{ edges} + 24 \text{ elements} = 40 \text{ adds}$
Parallel generation of the coarsest mesh

1. Build up a local mesh on each process
   - Loop over all elements in the mesh file: Read own macro-volume
   - Loop over all elements in the mesh file: Read neighbouring macro-volumes

2. Map all other macro-primitives to processes
   - Assign these primitives to the same process as the adjacent element with the largest ID

3. Assign IDs to edge and face primitives
   - Build tuples for all macro-edges and macro-faces (with one tuple entry per vertex ID)
   - Sort the lists of tuples
   - Assign IDs according to the index of the tuple in the sorted list
A triangle has a resolution of $6km^2$; $h_r = 3km$ in radial direction.
Strong scaling on Jugene

![Graph showing strong scaling on Jugene](image)

- **Memory - constant viscosity**
- **Memory - variable viscosity**
- **Strong scaling - constant viscosity**
- **Strong scaling - variable viscosity**
Performance modeling and analysis

V-cycle

1. Computation: stencil evaluations
2. Network effects: bandwidth, latency, reductions

including

- all multigrid levels and
- conjugate gradient on the coarsest mesh

Assumptions

- one inner point on the coarsest mesh per processor
- hardware information exclusively extracted from
  *IBM System Blue Gene Solution: Blue Gene/P Application Development*
BlueGene/P performance model (different setup)

![Graph showing performance model](image-url)
Summary

- Introduction into challenges of mantle convection simulations
- Smoothing strategies for anisotropic elements and varying coefficients
- Presentation of a conforming tetrahedral decomposition of an icosahedral mesh
- Performance evaluation and modeling

Current work

- Multigrid implementation & evaluation of (jumping) variable coefficients for HHG
- Preparations for SuperMUC
Thank you for your attention!
Benchmarks on Blue Gene/P in Jülich (Jugene)

- Compute node: 4-way SMP processor
- Processortype: 32-bit PowerPC 450 core 850 MHz
- Cores: 294 912
- Overall peak performance: 1 Petaflops
- Main memory: 2 Gbytes per node (aggregate 144 TB)