Mantle Convection Simulation in ASPECT

Timo Heister, Clemson University, heister@clemson.edu

2014-05-05, Banff

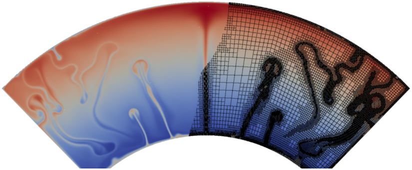
ASPECT Tutorial

- Tonight, 7:30pm
- Location: Black Bear
- Introduction, then hands-on
- Bring your laptop

Also: ASPECT office hours during poster sessions: Tuesday/Wednesday, starting 3:30pm

What?

ASPECT = **A**dvanced **S**olver for **P**roblems in **E**arth's **C**onvec**T**ion



- Mantle convection using modern numerical methods
- Open source, C++
- Available at: `http://aspect.dealii.org`
- Supported by NSF/CIG:

National Science Foundation WHERE DISCOVERIES BEGIN

CIG COMPUTATIONAL INFRASTRUCTURE for GEODYNAMICS

Kronbichler, Heister and Bangerth.
High Accuracy Mantle Convection Simulation through Modern Numerical Methods.
Geophysical Journal International, 2012, 191, 12-29.

Who?

- Wolfgang Bangerth (Texas A&M)
- Timo Heister (Clemson)
- Contributors:
  Markus Bürg, Juliane Dannberg,
  René Gaßmöller, Thomas Geenen,
  Ryan Grove, Eric Heien, Martin Kronbichler,
  Elvira Mulyukova, Ian Rose, Cedric Thieulot
  ⇝ **Thanks!**

Numerical Challenges

- Large range of scales in space and time
- High spacial resolution required
- Large problem sizes
- Non-linear coupling of equations
- Convection dominated
- Local vs. global models

$\rightsquigarrow$ need flexibility, accuracy, and scalability


www.shutterstock.com · 54480382

Modelling Challenges

- Complicated material models
- Uncertainties in parameters
- Benchmarking is hard
- Complex postprocessing
- Coupling with other tools

$\rightsquigarrow$ need usability, extensibility, easy experimentation

Social Challenges

- Expertise required in
  - numerical methods
  - scientific computing
  - large scale software development
  - geodynamics
  - mineral physics? seismology?
  - ...
- Large project: direction? continuity?
- need:
  - Documentation, tutorials
  - Training, workshops
  - Funding
  - Growing community is critical

$\rightsquigarrow$ CIG is a big help here!

Goals for ASPECT

- Modern numerical methods:
  adaptive mesh refinement, linear and nonlinear solvers,
  higher-order discretizations, stabilization schemes
- Usability and extensibility:
  manual: 170+ pages, cookbooks
  plugin architecture
- Parallel scalability
- Building on others' work:
  tested foundation, smaller codebase, automatic improvements
- Community:
  GPL, developed in the open,
  many contributors, we want to help

Timeline

- 2008-2011: deal.II based examples/experiments (Bangerth)
- Oct 2011: Aspect development started
- March 2012: release 0.1
- April 2013: release 0.2:
  - compositional fields, passive tracers, GPlates, mesh refinement criteria
- May 2013: release 0.3 (bugfixes)
- April 2014: release 1.0
  - a lot of new documentation
  - new examples (2d/3d shells, . . .)
  - dynamic topography
  - big performance improvements
  - compositional fields: reactions, boundary conditions, . . .
  - overhauled tracers
  - periodic meshes, nullspace removal, PETSc support

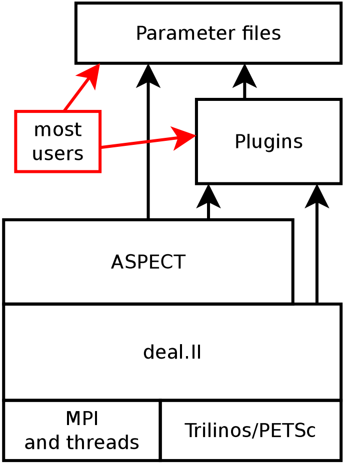
Status

Lifetime of scientific codes:

1. experimentation with new numerical methods ✓
2. working, useful for early adopters ✓
3. useful tool for science applications (with limitations) ✓
4. science driven development ⟵ ASPECT
5. maintenance
6. abandoned

⇝ need YOUR science problems and feedback for future directions!

Structure of ASPECT



Problem setup, configuration

Materials, Geometries/Boundaries, Adiabats, Postprocessing, Visualization, Interfacing to other tools

Equations, Numerical schemes, Framework

Finite Elements, AMR, Parallel abstraction, Postprocessing, Visualization

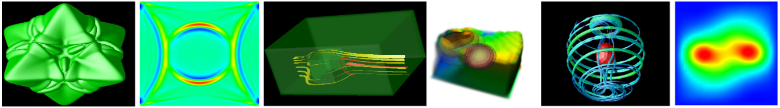
Parallelization, IO, linear algebra, linear solvers

Features of deal.II

- Open source project, C++
- Maintainers: W. Bangerth, T. Heister, G. Kanschat
- One of the most widely used finite element libraries:
  - ~ 400 papers using and citing deal.II, ~ 600 downloads/month
- Excellent documentation, examples
- Features:
  - Many finite element types (continuous, DG, RT, . . .)
  - Higher order elements, hp adaptivity
  - Adaptive mesh refinement in 2d, 3d (quads/hexas)
  - Linear algebra: interfaces to PETSc and Trilinos
  - Parallel computing (MPI and/or multi-threading)

  my area

Bangerth, Heister and Kanschat.
deal.II *Differential Equations Analysis Library, Technical Reference*, 2012.
http://www.dealii.org.



Equations

velocity $\mathbf{u}$, pressure p, temperature T, advected quantities c_i:

$$-\nabla\cdot[2\eta D(\mathbf{u})]+\nabla p=\rho\mathbf{g} \qquad \text{in } \Omega, \tag{1}$$

$$\nabla\cdot(\rho\mathbf{u})=0 \qquad \text{in } \Omega, \tag{2}$$

$$\rho C_p\left(\frac{\partial T}{\partial t}+\mathbf{u}\cdot\nabla T\right)-\nabla\cdot k\nabla T=F \qquad \text{in } \Omega, \tag{3}$$

$$\frac{\partial c_i}{\partial t}+\mathbf{u}\cdot\nabla c_i=0 \qquad \text{in } \Omega \tag{4}$$

- strain rate $D(\mathbf{u})=\frac{1}{2}\left(\nabla\mathbf{u}+\nabla\mathbf{u}^T\right)-\frac{1}{3}(\nabla\cdot\mathbf{u})\mathbf{1}$
- density $\rho(p,T,c,\mathbf{x})$, viscosity $\eta(\mathbf{u},p,T,c,\mathbf{x})$
- gravity $\mathbf{g}(\mathbf{x})$
- specific heat $C_p(p,T,c,\mathbf{x})$, thermal conductivity $k(p,T,c,\mathbf{x})$
- F: radioactive decay, friction heating, adiabatic compression, latent heat, ...

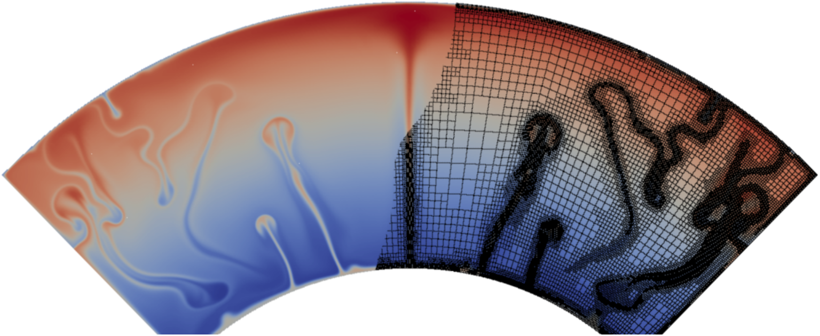
Misc

- Dimensionalized computations:
  - correct units and scalings
  - can still non-dimensionalize if desired
- Geometries (2d and 3d):
  shell, sphere, box, periodic domains, topography, ...
- Interface with: GPlates, PerpleX, (more in progress)
- Automated test suite
- We are moving from svn to git for version control

Discretization

- Time:
  - BDF2
  - 2nd order in time, unconditionally stable
- Space:
  - higher order finite elements
  - inf-sup stable element pair for velocity and pressure
  - typically: $u : Q_2$, $p : Q_1$, $T : Q_2$
  - convergence in space: 3rd order (for u, T)
- Temperature stabilization:
  - entropy viscosity method
  - add diffusion where:
    1. solution is non-smooth
    2. local Peclet number $= \frac{LU}{\kappa}$ is large

Adaptive Mesh Refinement (AMR)



- Put mesh resolution where needed
- Adapt mesh every couple of time steps

Why AMR?

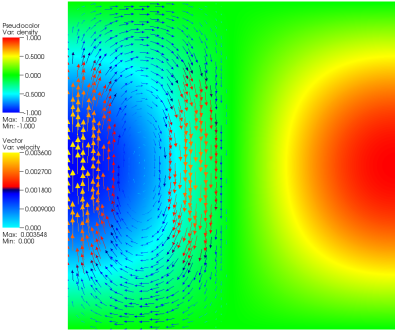
- Global fine resolution is too expensive:
  - 1km resolution requires ~ 4 trillion unknowns
  - 10km resolution requires ~ 4 billion unknowns
  - possible, but expensive (to compute, store, visualize, ...)
- Save 10x-100x over fixed fine resolution
- Way to cope with discontinuities in pressure/viscosity/density

Discontinuities: SolCx Benchmark

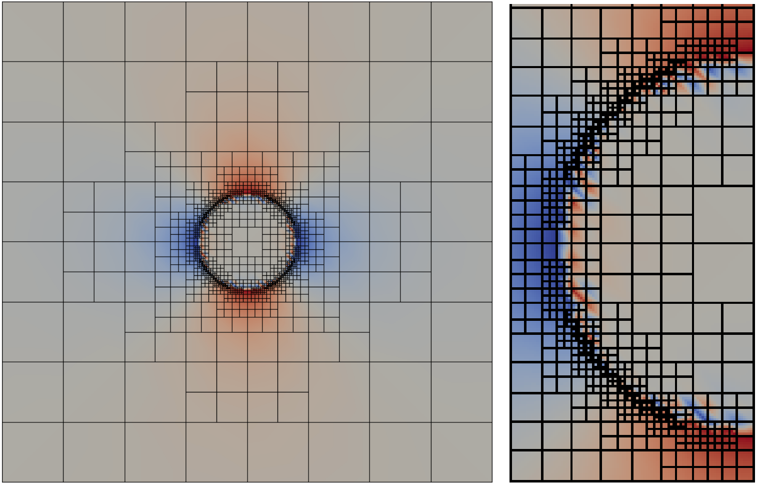
- 10^8 viscosity jump, gives boundary layer in pressure
- Continuous (Q_1) vs. discontinuous (P_{-1}) pressure element:

Element	Velocity	Pressure	Comment
$Q_2 \times Q_1$	h^3	$h^{1/2}$	
$Q_2 \times P_{-1}$	h^3	h^2	aligned
$Q_2 \times P_{-1}$	h^3	$h^{1/2}$	not aligned

- Discontinuous pressure space doesn't help in practice, refining into the layer does

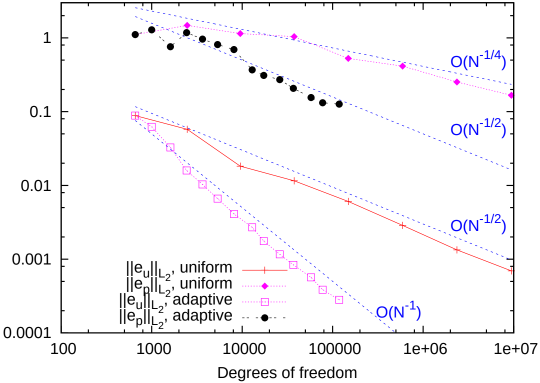


Discontinuities: Inclusion benchmark



- disk with 10^3 viscosity jump, never aligned with cells
- pressure oscillations

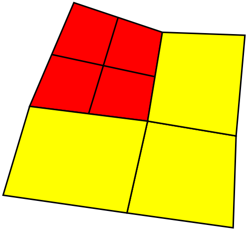
Discontinuities: Inclusion benchmark



- Adaptive refinement gives up to 100x smaller error

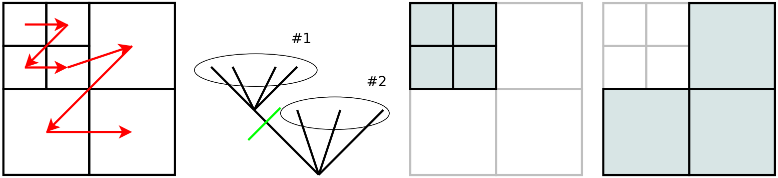
How?

- Start with coarse mesh
- Keep refining into 4 (8 in 3d) children where needed
- Need:
  - efficient, parallel datastructures
  - refinement criteria (temperature gradient-jumps, density jumps, . . .)
  - fast parallel partitioning every couple of timesteps



Space-Filling Curves

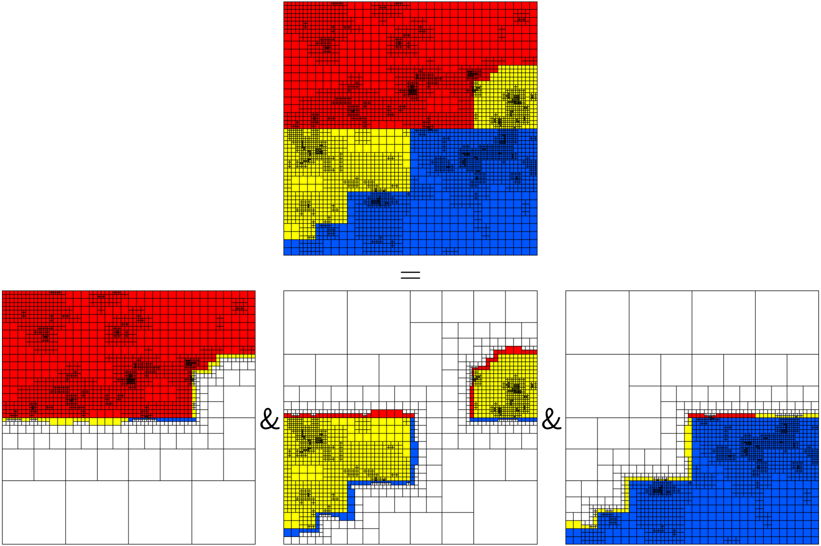
- Space-filling curves (using `p4est` library)
- Partitioning is cheap and simple:



- Only store local mesh on each core

Burstedde, Wilcox, and Ghattas.
p4est: Scalable algorithms for parallel adaptive mesh refinement on forests of octrees.
SIAM J. Sci. Comput., 33 no. 3 (2011), pages 1103-1133.

Example: Distributed Mesh Storage



Color: owned by CPU id

Solvers

- In each timestep: coupled $\mathbf{u}, p, T$ system
- IMPES scheme:
  - solve temperature and Stokes separately
  - extrapolate other quantities
  - unconditionally stable
  - in practice CFL due to mesh: $\triangle t = C \cdot \min \frac{h_k}{\|u\|_\infty}$
- Alternatively: iterate out non-linear coupling

$\rightsquigarrow$ need Stokes solver

Stokes Solver

- Incompressible Stokes:

$$
\begin{aligned}
-\triangle \mathbf{u} + \nabla p &= f, \\
\nabla \cdot \mathbf{u} &= g
\end{aligned}
$$

- Weak form:

$$
\begin{aligned}
(\nabla \mathbf{u}, \nabla \mathbf{v}) - (p, \nabla \cdot \mathbf{v}) &= (f, \mathbf{v}), \\
-(\nabla \cdot \mathbf{u}, q) &= -(g, q)
\end{aligned}
$$

- Linear system:

$$
\begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ p \end{pmatrix} = \begin{pmatrix} F \\ G \end{pmatrix}
$$

Stokes Solver

- Solve linear system with (flexible) GMRES:

$$\begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ p \end{pmatrix} = \begin{pmatrix} F \\ G \end{pmatrix} \qquad P = \begin{pmatrix} A & B^T \\ 0 & -S \end{pmatrix}$$

- Right preconditioning with operator P^{-1}.
- Schur complement $S = BA^{-1}B^T$.
- Applying P^{-1} requires application of A^{-1} and S^{-1}
  - Approximations are enough
  - A^{-1}: either one multigrid V-cycle, or CG preconditioned with V-Cycle. Algebraic multigrid (Trilinos ML)
  - S^{-1}: approximated using pressure mass matrix M_p: $M_p = (\eta^{-1}\phi_i, \phi_j)$ using CG with block ILU.

Compressible Case

- Instead of $\nabla \cdot \mathbf{u} = 0$ we have $\nabla \cdot (\rho \mathbf{u}) = 0$
- No longer symmetric to ∇p (makes preconditioning really difficult) and nonlinear
- Divide by ρ, then:

$$\frac{1}{\rho}\nabla \cdot (\rho \mathbf{u}) = \nabla \cdot \mathbf{u} + \boxed{\frac{1}{\rho}\nabla \rho \cdot \mathbf{u}}$$

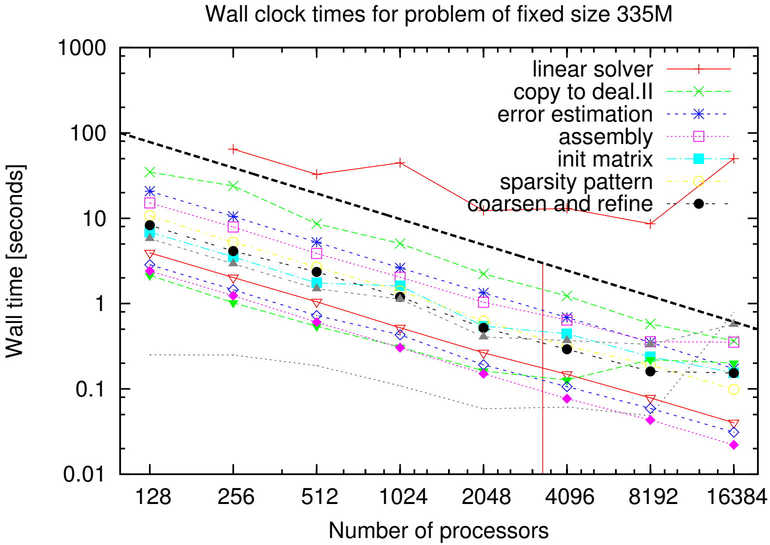
- simplify:

$$\frac{1}{\rho}\nabla \rho \cdot \mathbf{u} \approx \frac{1}{\rho}\frac{\partial \rho}{\partial p}\nabla p \cdot \mathbf{u} \approx \frac{1}{\rho}\frac{\partial \rho}{\partial p}\nabla p_s \cdot \mathbf{u} \approx \frac{1}{\rho}\frac{\partial \rho}{\partial p}\rho \mathbf{g} \cdot \mathbf{u}$$

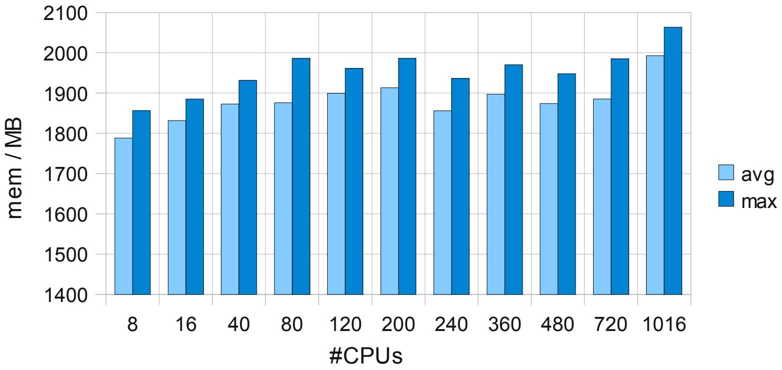
  compressibility $\frac{1}{\rho}\frac{\partial \rho}{\partial p}$; use static pressure to get $\nabla p \approx \nabla p_s \approx \rho \mathbf{g}$
- So get back to the incompressible case:

$$\nabla \cdot \mathbf{u} = -\frac{1}{\rho^*}\frac{\partial \rho^*}{\partial p}\rho^* \mathbf{g} \cdot \mathbf{u}^*$$

Strong Scaling: 2d Adaptive Poisson Problem



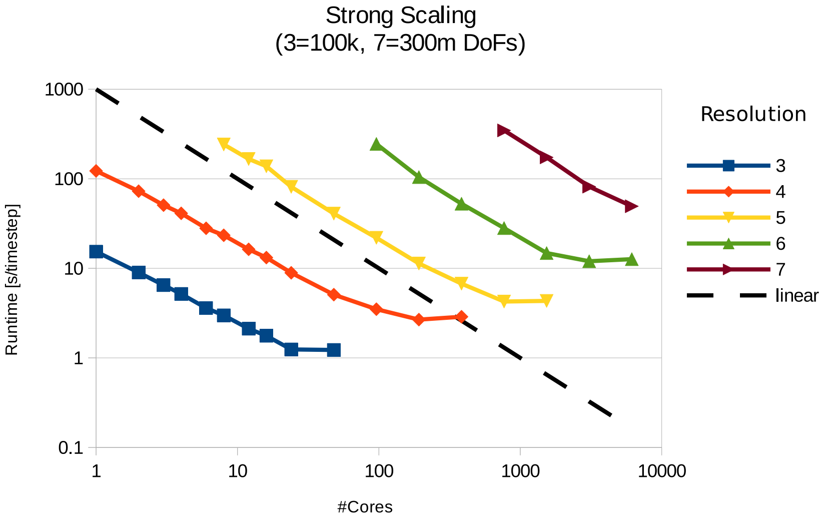
Test: Memory Consumption



average and maximum memory consumption (VmPeak)
3D, weak scalability from 8 to 1000 processors with about 500.000 DoFs per processor (4 million up to 500 million total)

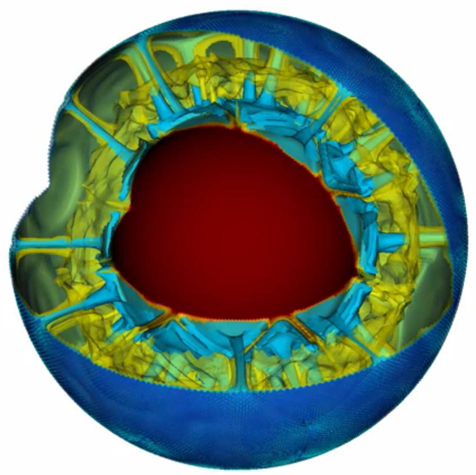
$\rightsquigarrow$ **Constant memory usage with increasing # CPUs & problem size**

ASPECT



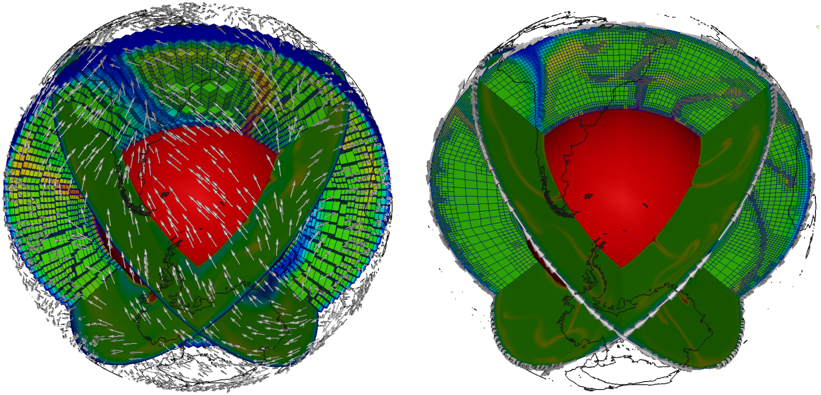
on Cray XC30, done by Rene Gassmoeller

Adaptive 3d Computations

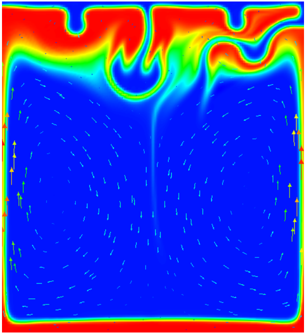


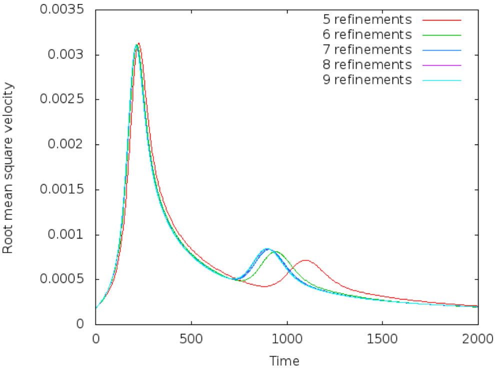
https://www.youtube.com/watch?v=j63MkEc0RRw

GPlates Coupling (Rene Gassmoeller)

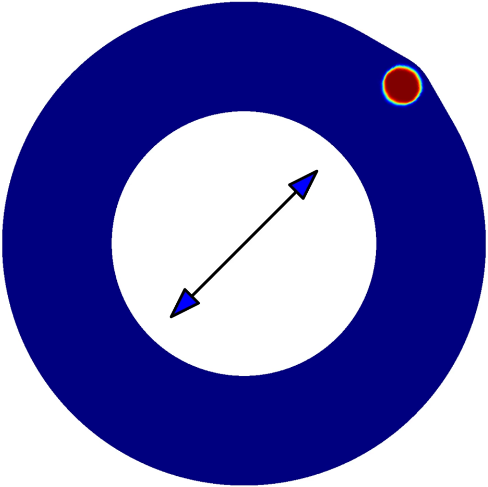


Benchmarks: van Keken

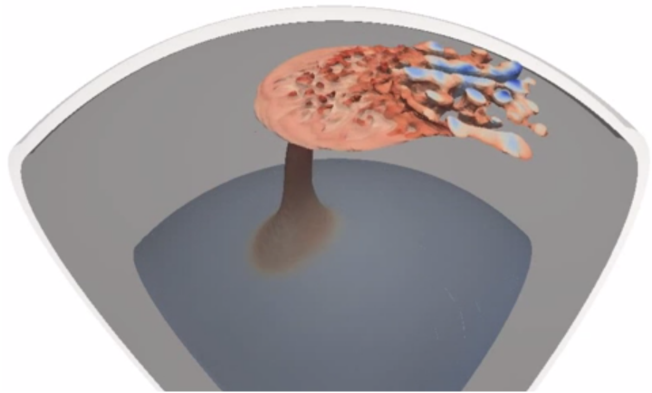




Free Surface Computations (Ian Rose)

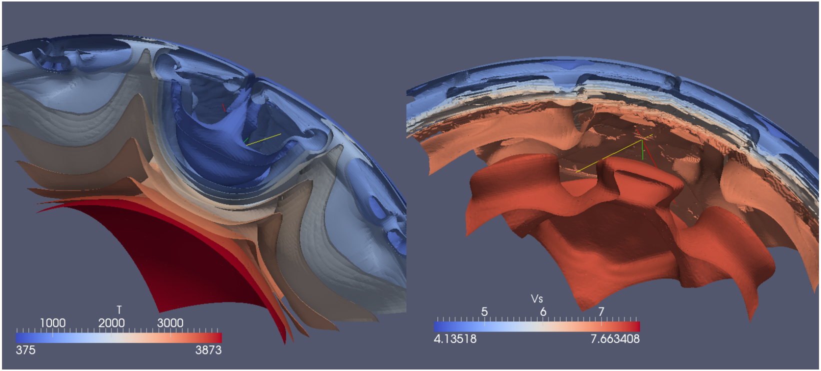


Thermochemical convection (Juliane Dannberg)



https://www.youtube.com/watch?v=dG-ULmcBr1E

Mineral phases, Vs (Thomas Geenen)



Future

- Free surface computations
- Benchmarking efforts
- Nonlinear solvers
- Improvements: tracers, stabilization, compositional fields/levelsets
- Science questions

Hackaton!

ASPECT Hackathon: Texas A&M, May 14-23, 2014

- development of ASPECT and work on your science problems
- travel support through CIG (if US based)

Conclusions

- Tutorials:
  - TONIGHT, 7:30pm, room: Black Bear
  - CIDER, July 2014, Santa Barbara, CA
  - SEDI, August 2014, Kanagawa, Japan
  - GeoMod2014, September 2014, Potsdam, Germany
- Hackathon: May 14-23, Texas A&M

Thanks for your attention!