Simulating Complex Flows in the Earth Mantle: Time stepping in the ASPECT code

Wolfgang Bangerth

Colorado State University

Joint work with Juliane Dannberg, Menno Fraters, Rene Gassmoeller, Anne Glerum, Timo Heister, Bob Myhill, John Naliboff, and many many other contributors

With funding by:

National Science Foundation
WHERE DISCOVERIES BEGIN

CIG COMPUTATIONAL INFRASTRUCTURE for GEODYNAMICS

There are no “Conclusions” to this talk.

Take-away message (charitable version):

The world is messy:

- **Models are messy**
- **Development histories are messy**

Take-away message (not-so-charitable version):

The developers of this code are amateurs:

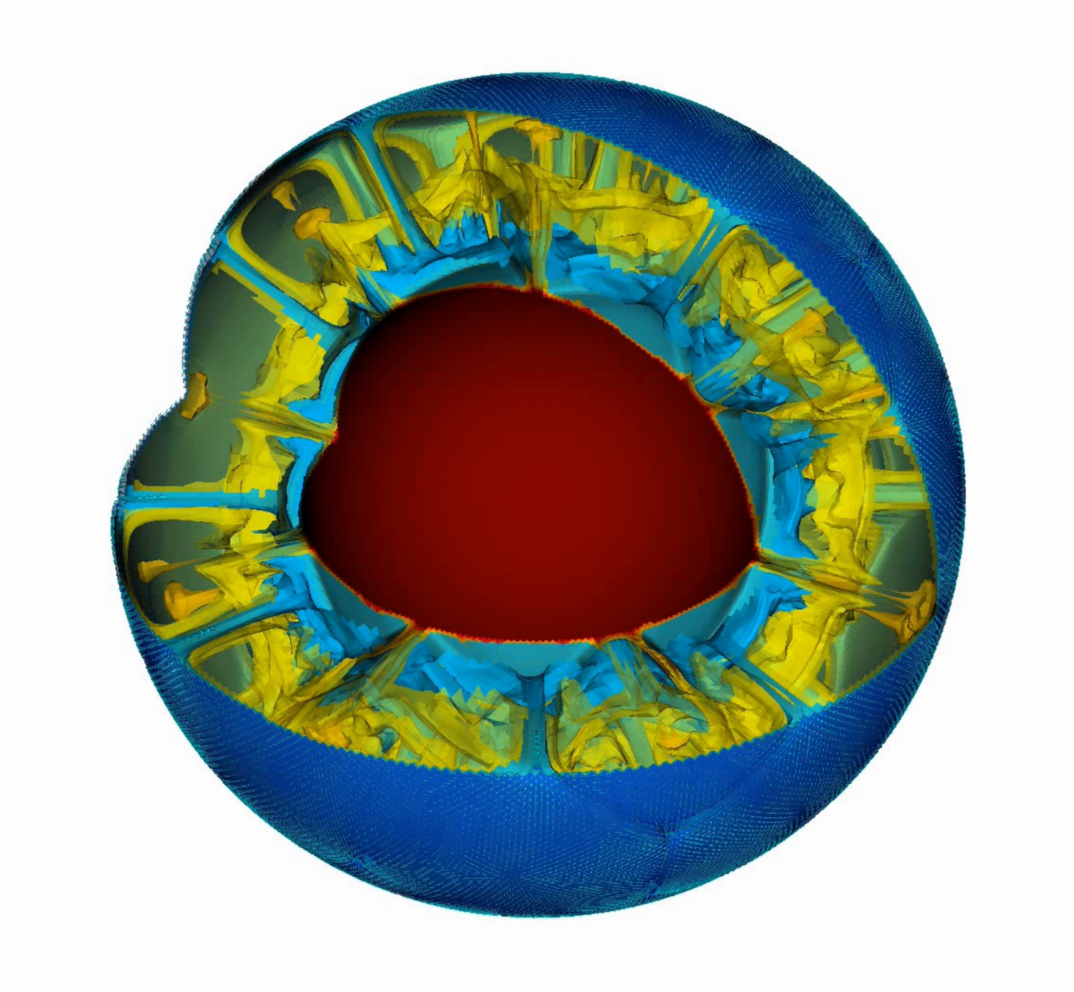
- **We were (and are) not experts in time stepping**
- **We failed to anticipate future directions when coming up with the original design**

In particular: There is no “holistic design”.

Introduction:

What and why?

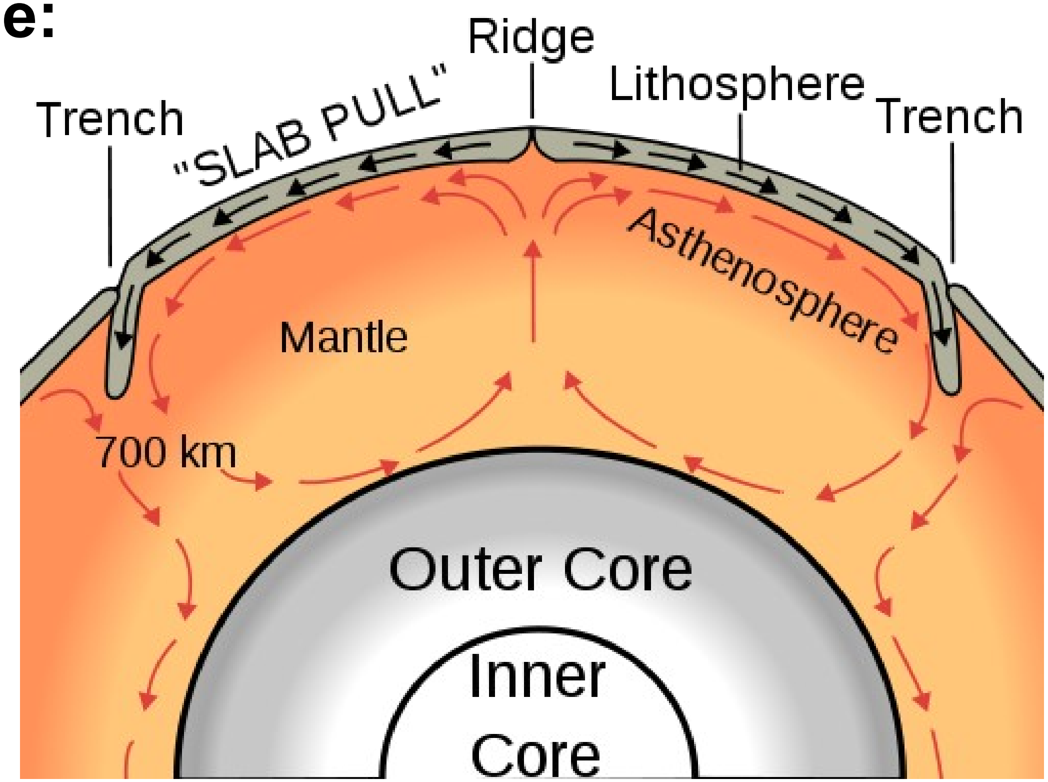
The what and the why



The what and the why

Brief recap on the Earth mantle:

- It makes up the largest component of Earth (~80%)
- It is solid
- It flows on long time scales
- Thermal gradients drive convection



- It is probably Earth's component we understand the least
- Yet, it has a large impact on basically everything else
- It's relevance lies in the interaction with the rest of Earth

The what and the why

“Big” questions one may ask about the mantle in relation to other systems:

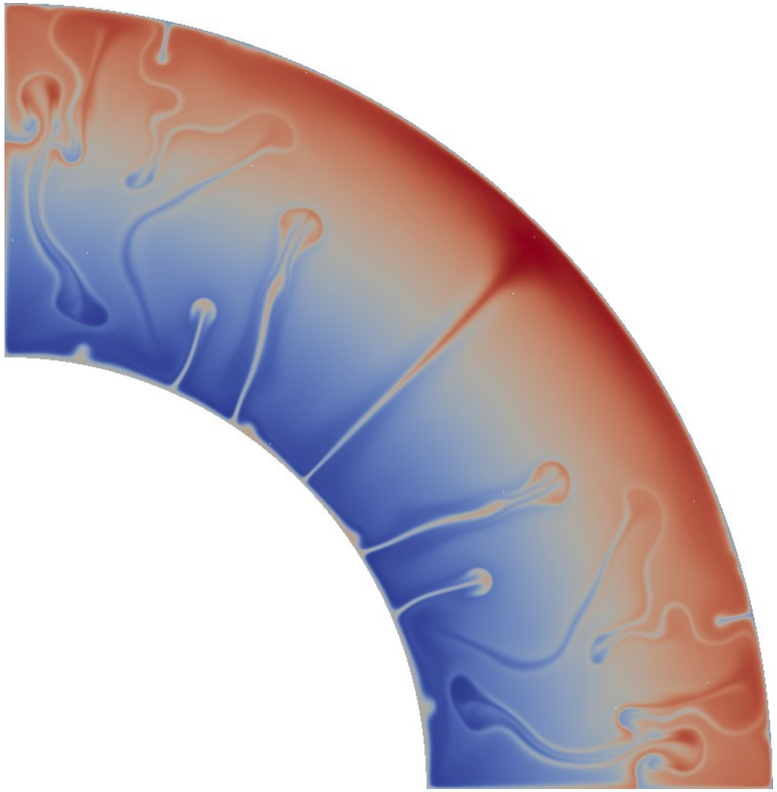
- *Mantle vs. Lithosphere (the “crust”):*
  How does mantle convection interact with plate tectonics?
  How does mantle convection affect natural resources?

- *Mantle vs. Atmosphere:* How does carbon cycle in the Earth?
  Mantle vs. Oceans: How does water cycle in the Earth?
  → Does mantle convection affect the habitability of planets?

- *Mantle vs. Core:* How is heat transported from core to surface?
  → impact on the magnetic field
  → thermal history of Earth
  → history of the inner core

Part I:

“Classical” mantle convection



The model

Thermal convection is described by the relatively “simple” Boussinesq approximation (or variations):

$$-\nabla\cdot[2\eta\epsilon(u)]+\nabla p=g\rho(T)$$

$$\nabla\cdot u=0$$

$$\frac{\partial T}{\partial t}+u\cdot\nabla T-\kappa\Delta T=\gamma+\alpha\left(\frac{\partial p}{\partial t}+u\cdot\nabla p\right)+\eta(\nabla u)^2$$

This is not dissimilar from a typical “model problem”.

Challenges: Problem size

For (global) convection in the earth mantle:

- Depth: ~35 – 2890 km
- Volume: $\sim 10^{12}$ km^3
- Resolution required: <10 km
- Uniform mesh: $\sim 10^9$ cells
- Using Taylor-Hood (Q_2/Q_1) elements: 33B unknowns
- At 100k-1M DoFs/processor: 30k-300k processors!

Consequence: We need adaptive mesh refinement.

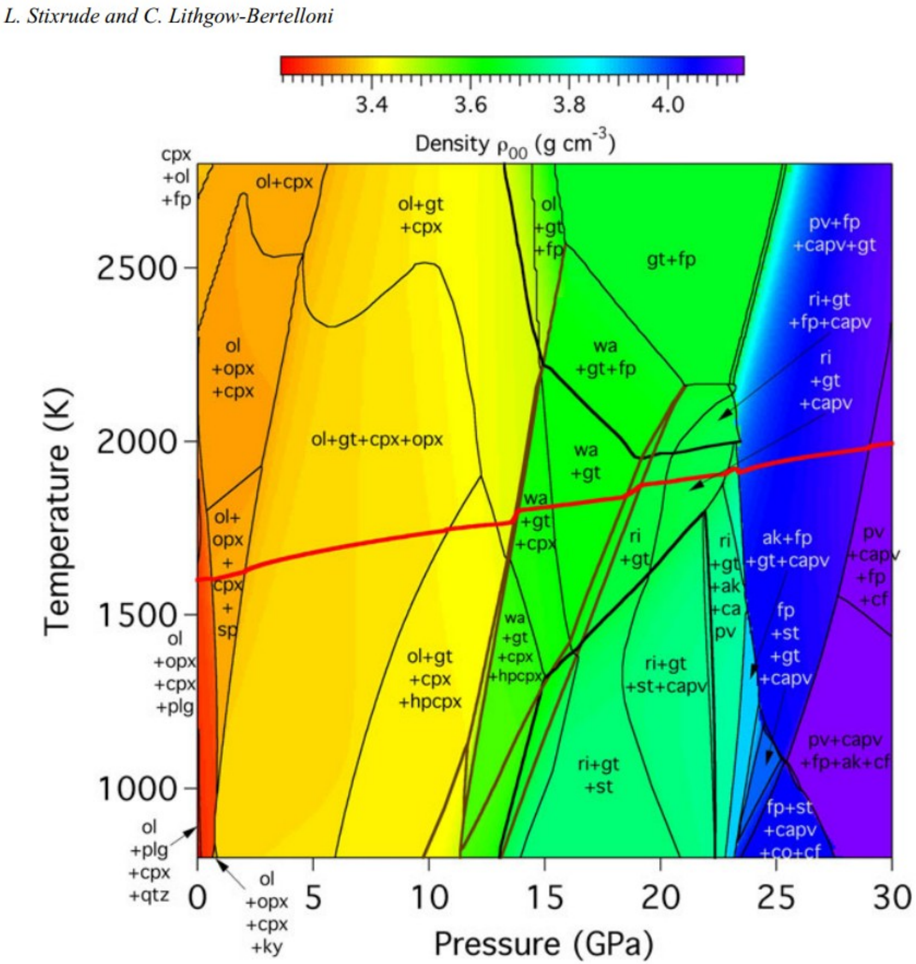
Challenges: Model complexity

However, in reality:

- All coefficients depend nonlinearly on
  - pressure
  - temperature
  - strain rate
  - chemical composition
- Dependency is not continuous

Moreover:

- Viscosity varies by at least 10^6
- Material is compressible
- Geometry depends on solution



Solutions

Among the mathematical techniques we use are:

- *"Higher" order time stepping schemes*
- Higher order finite elements
- Fully adaptive, dynamically changing 3d meshes
- Iterate out the nonlinearity via fixed-point and Newton methods
- Silvester/Wathen-style block preconditioners with F-GMRES
- Algebraic or geometric multigrid for the elliptic part
- Parallelization using MPI, threads, and tasks

To make the code usable by the community:

- Use object-oriented programming, build on external tools
- Make it modular, separate concerns
- Extensive documentation
- Extensive and frequent testing

Time discretization

Recall the model:

$$-\nabla\cdot[2\eta\epsilon(u)]+\nabla p=g\rho(T)$$

$$\nabla\cdot u=0$$

$$\frac{\partial T}{\partial t}+u\cdot\nabla T-\kappa\Delta T=\gamma+\alpha\left(\frac{\partial p}{\partial t}+u\cdot\nabla p\right)+\eta(\nabla u)^2$$

Time discretization

Overall algorithm:

While ($T<T_{end}$):

- Solve Stokes equation
- From velocity, compute time step
- Solve for temperature field
- Advance time

Time discretization for the temperature

Time equation:

$$\frac{\partial T}{\partial t} + u \cdot \nabla T - \kappa \Delta T = \gamma + \alpha \left(\frac{\partial p}{\partial t} + u \cdot \nabla p \right) + \eta (\nabla u)^2$$

Considerations:

- Adaptive mesh refinement → no high-order multistep methods
- Variable time step size → no high-order multistep methods
- Segregated solver → velocity not available at intermediate times → what to do about RK methods?
- Spatial error dominant (?) → high order not necessary (?)

Our choice: BDF2

Time discretization for the temperature

BDF2 applied to

$$\frac{\partial T}{\partial t} + u \cdot \nabla T - \kappa \Delta T = \gamma + \alpha \left(\frac{\partial p}{\partial t} + u \cdot \nabla p \right) + \eta (\nabla u)^2$$

results in

$$\alpha_n T^n + u^n \cdot \nabla T^n - \kappa \Delta T^n = F(u^{n-1}, T^{n-1}, u^{n-2}, T^{n-2})$$

Considerations:

- We need an efficient linear solver for the discretized system
- The matrix is non-symmetric
- Treat advection as explicit instead:

$$\alpha_n T^n - \kappa \Delta T^n = -u^n \cdot \nabla T^* + F(u^{n-1}, T^{n-1}, u^{n-2}, T^{n-2})$$

Time discretization for the temperature

Semi-implicit BDF2:

$$\alpha_n T^n - \kappa \Delta T^n = -u^n \cdot \nabla T^* + F\left(u^{n-1}, T^{n-1}, u^{n-2}, T^{n-2}\right)$$

Consequences:

- The matrix is now symmetric
- Efficient linear solvers are easy to construct
- **But:** We now have a time step restriction

$$k_n \;\leq\; C_{\mathrm{BDF2}} \min_{K \in T} \frac{h_K}{\|u\|_{L_\infty(K)}}$$

Time discretization for the temperature

CFL condition – the struggle is real:

$$k_n \leq C_{\mathrm{BDF2}} \min_{K \in T} \frac{h_K}{\|u\|_{L_\infty(K)}}$$

Questions:

- What is C_{BDF2}?
- What is h_K on unstructured 3d meshes with curved edges?
- How does all of this relate to the eigenvalues of the matrix?

After much experimentation:

- Choose h_K as the diameter of K
- Choose C_{BDF2}=0.085 → quite small actually

Time discretization for the temperature

After much agony, change of mind – go back to fully implicit:

$$\alpha_n T^n + u^n \cdot \nabla T^n - \kappa \Delta T^n = F\left(u^{n-1}, T^{n-1}, u^{n-2}, T^{n-2}\right)$$

Considerations:

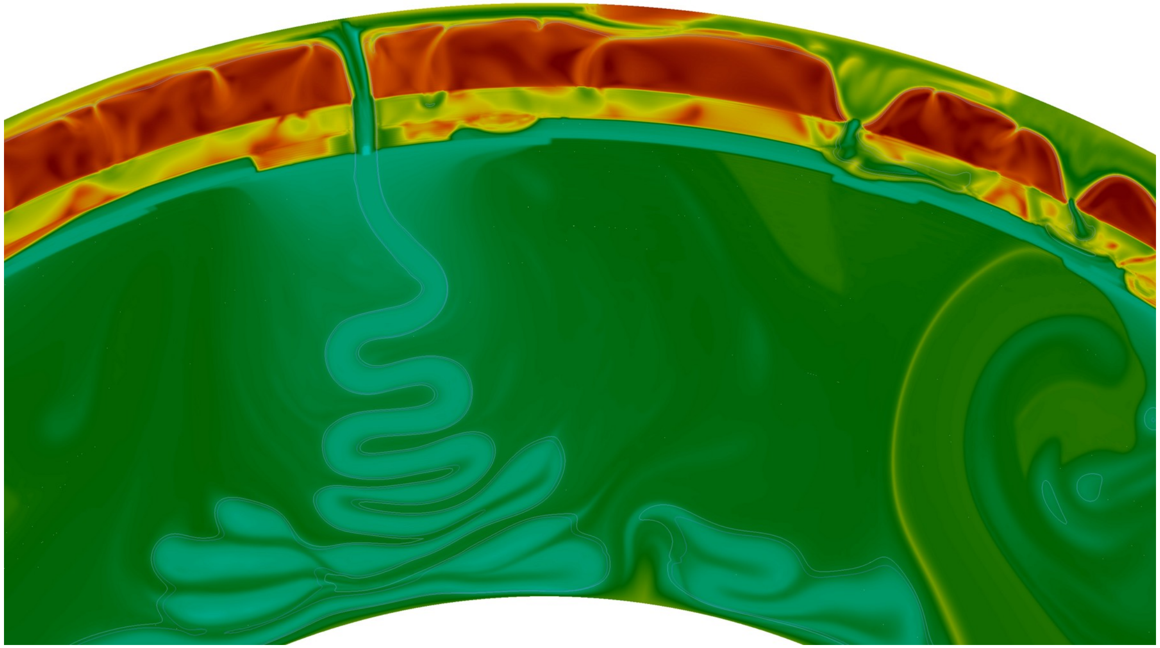
- Need to work harder to solve linear system
- But no longer time-step restricted; choose

$$k_n = 1 \cdot \min_{K \in T} \frac{h_K}{\|u\|_{L_\infty(K)}}$$

Because the Stokes solve is so expensive, the larger time step easily balances the more expensive temperature solve.

Part II:

What then? – Compositional fields



Compositional fields

Juliane Dannberg comes along (2012):

We also want to track chemical compositions:

$$-\nabla\cdot[2\eta\epsilon(u)]+\nabla p=g\rho(T)$$

$$\nabla\cdot u=0$$

$$\frac{\partial T}{\partial t}+u\cdot\nabla T-\kappa\Delta T=\gamma+\alpha\left(\frac{\partial p}{\partial t}+u\cdot\nabla p\right)+\eta(\nabla u)^2$$

$$\boxed{\begin{gathered}\frac{\partial c_1}{\partial t}+u\cdot\nabla c_1-\kappa\Delta c_1=q_1(u,p,T,\vec{c})\\ \ldots\\ \frac{\partial c_N}{\partial t}+u\cdot\nabla c_N-\kappa\Delta c_N=q_N(u,p,T,\vec{c})\end{gathered}}$$

Chemical compositions

Considerations:

- Originally meant to track *compositions* → zero right hand sides
- Then *mineral compositions* → chemical reactions
- Then also other quantities (melting, accumulated strains, level sets, …) → many many such fields

- Would like to solve in a coupled fashion, but too memory expensive
- Solving advection equations suddenly becomes expensive

- Solve in segregated fashion, treat rhs explicitly

Time discretization

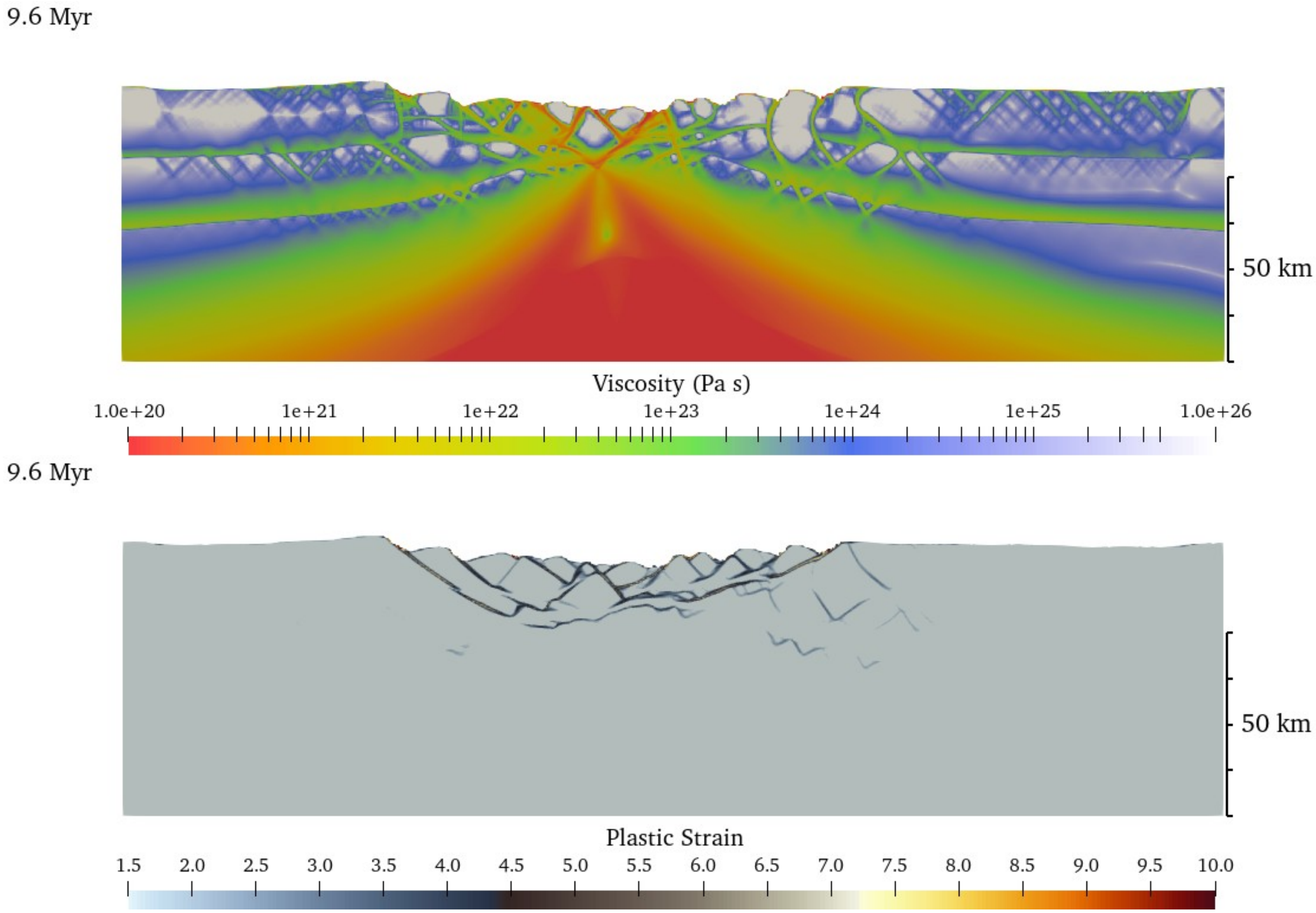
Overall algorithm:

While ($T<T_{end}$):

- Solve Stokes equation
- From velocity, compute time step
- Solve for temperature field with implicit BDF2
- Solve for compositional field 1 with implicit BDF2, explicit rhs
- …
- Solve for compositional field N with implicit BDF2, explicit rhs
- Advance time

Part III:

What then? – Stiff source terms



Stiff source terms

John Naliboff, Juliane Dannberg, et al. come along (2017):

"compositional field" is elastic stress, which decays quickly in time:

$$-\nabla \cdot [2\eta\epsilon(u)] + \nabla p = g\rho(T)$$

$$\nabla \cdot u = 0$$

$$\frac{\partial T}{\partial t} + u \cdot \nabla T - \kappa \Delta T = \gamma + \alpha\left(\frac{\partial p}{\partial t} + u \cdot \nabla p\right) + \eta(\nabla u)^2$$

$$\frac{\partial c_1}{\partial t} + u \cdot \nabla c_1 - \kappa \Delta c_1 = q_1(u, p, T, \vec{c})$$

$$\ldots$$

$$\frac{\partial c_N}{\partial t} + u \cdot \nabla c_N - \kappa \Delta c_N = q_N(u, p, T, \vec{c})$$

Stiff source terms

Considerations:

- Elastic stress relaxes on a time scale faster than the flow → "multirate" system
- Impossible to resolve this time scale in a coupled scheme
- Treat rhs via operator splitting:
  – currently uses Lie splitting
  – currently integrates pointwise ODE with a fixed micro timestep

Time discretization

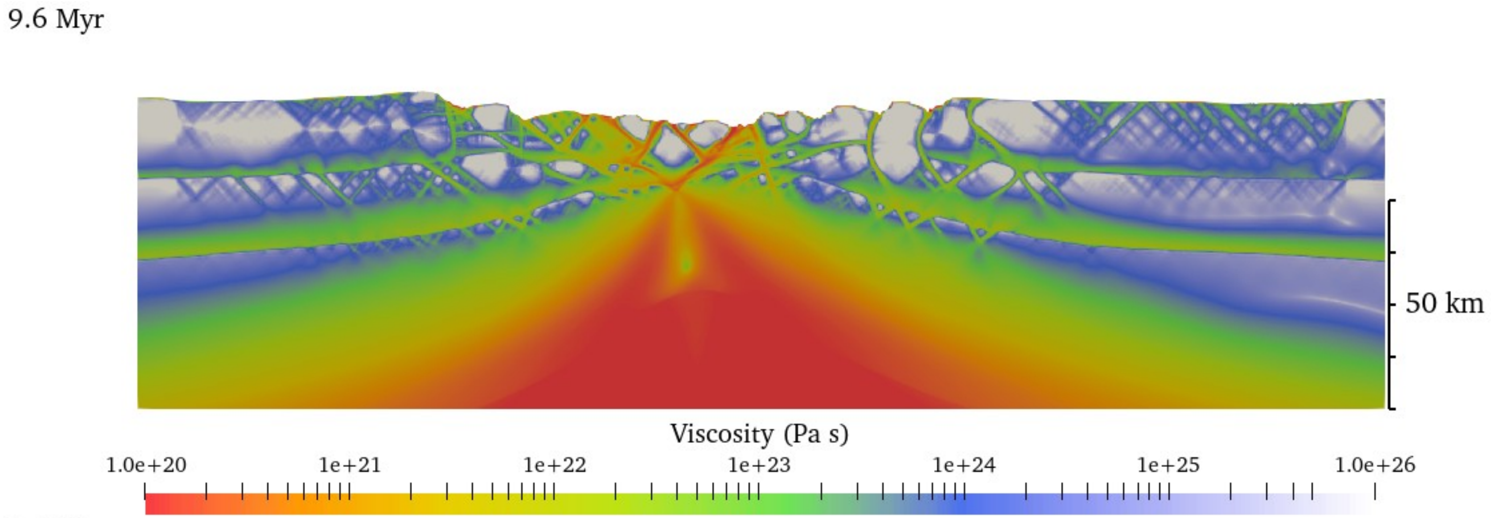
Overall algorithm:

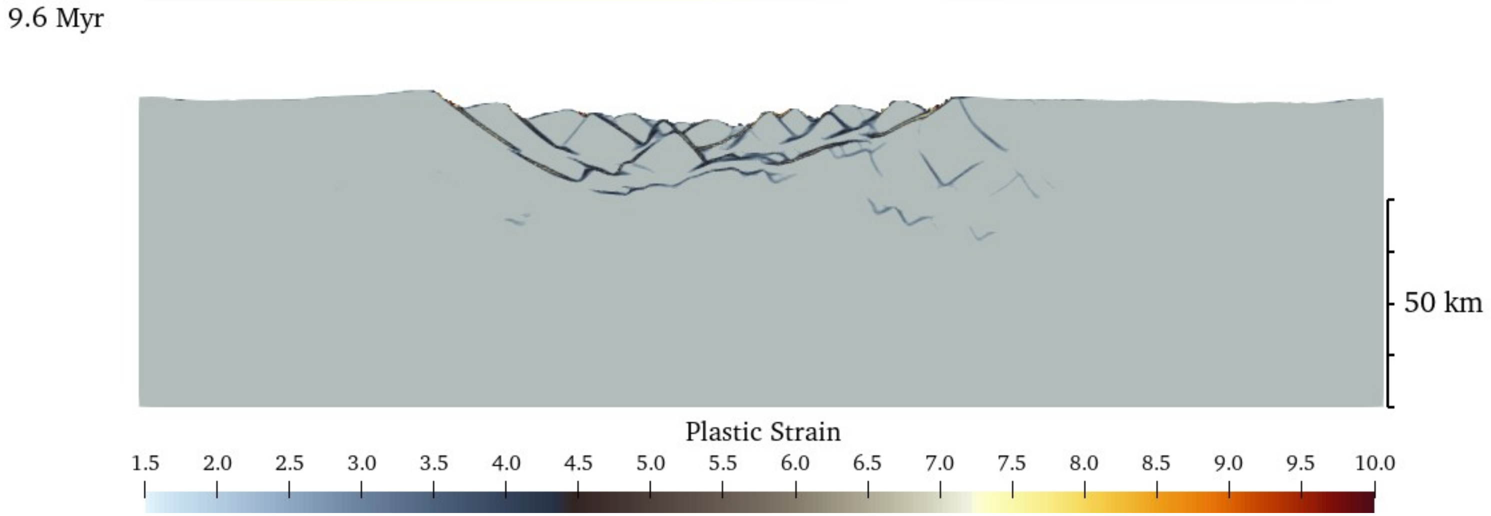
While ($T<T_{end}$):

- Solve Stokes equation
- From velocity, compute time step
- Solve for temperature field with implicit BDF2
- Solve for compositional field 1, implicit BDF2, operator splitting
- …
- Solve for compositional field 1, implicit BDF2, operator splitting
- Advance time

Part IV:

What then? – Free surfaces





Free surfaces

Then Ian Rose and Timo Heister come along (2014):

- We also want to deform the surface of the domain
- Evaluate residual stresses at the surface, move nodes at boundary and in domain
- Requires one Laplace solve

Then Anne Glerum comes along (2020):

- Diffuse the surface to mimic erosion

Free surfaces

Equations now:

$$-\nabla\cdot[2\eta\epsilon(u)]+\nabla p=g\rho(T)$$

$$\nabla\cdot u=0$$

$$\frac{\partial T}{\partial t}+u\cdot\nabla T-\kappa\Delta T=\gamma+\alpha\left(\frac{\partial p}{\partial t}+u\cdot\nabla p\right)+\eta(\nabla u)^2$$

$$\frac{\partial c_1}{\partial t}+u\cdot\nabla c_1-\kappa\Delta c_1=q_1(u,p,T,\vec{c})$$

$$\ldots$$

$$\frac{\partial c_N}{\partial t}+u\cdot\nabla c_N-\kappa\Delta c_N=q_N(u,p,T,\vec{c})$$

$$\boxed{\frac{\partial h}{\partial t}-A\Delta h=r(u,p)}$$

Time discretization

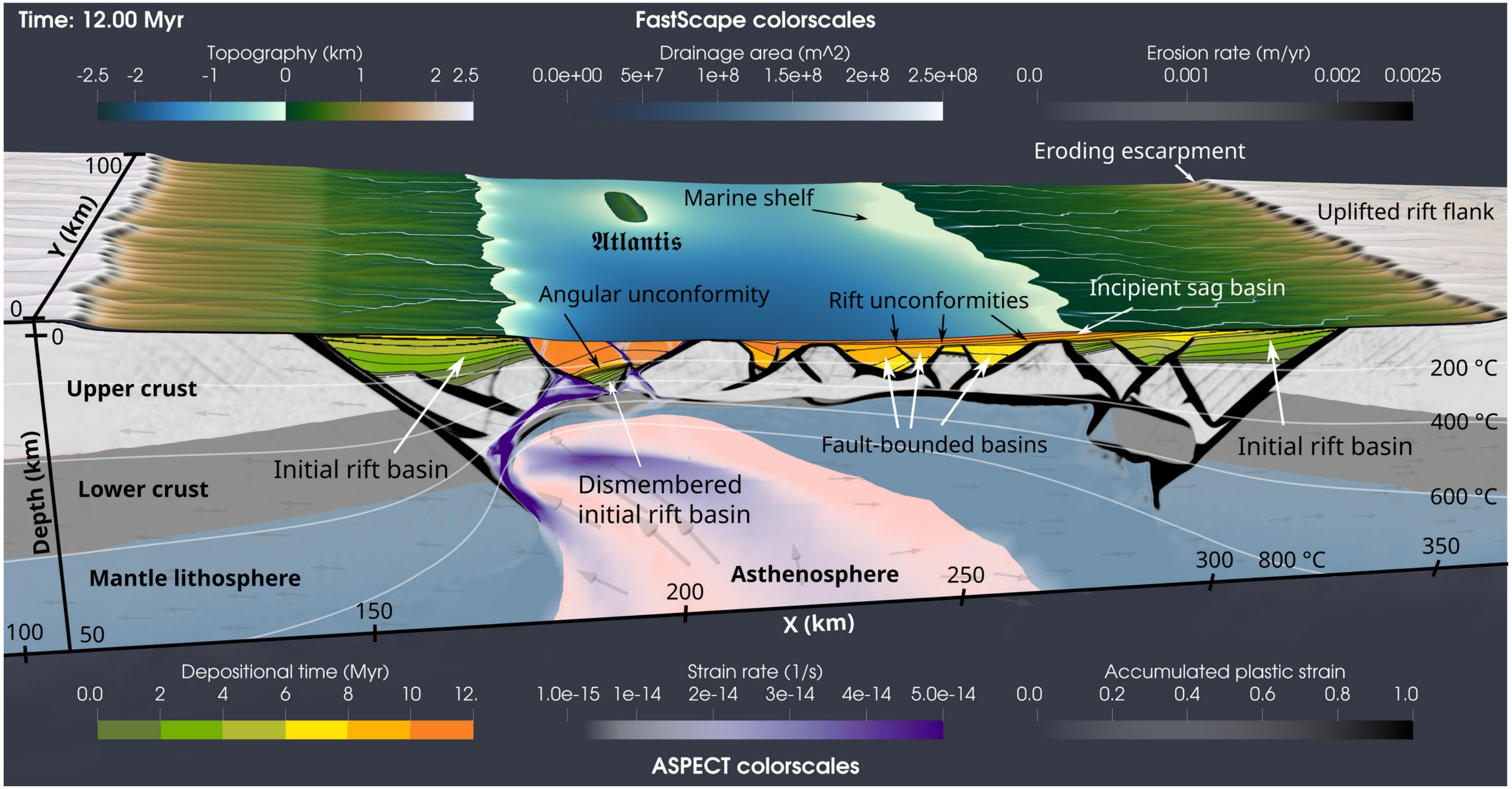
Overall algorithm:

While ($T<T_{end}$):

- Solve Stokes equation
- From velocity, compute time step
- Solve for temperature field with implicit BDF2
- Solve for compositional field 1, implicit BDF2, operator splitting
- …
- Solve for compositional field 1, implicit BDF2, operator splitting
- Solve for surface deformation
- Advance time

Part V:

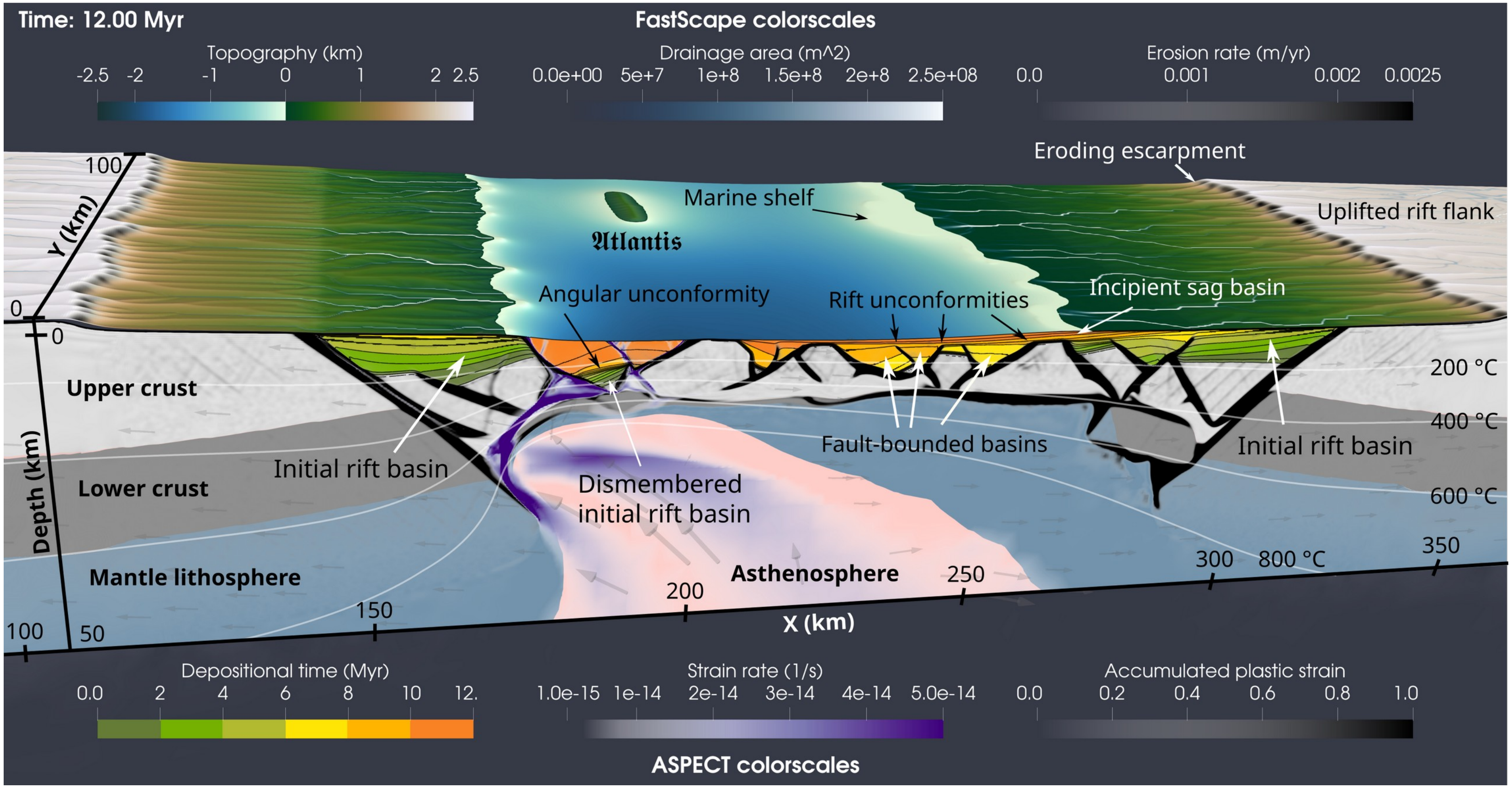
What then? – Surface evolution



Realistic surfaces

Derek Neuharth et al. come along (2021):

“Real” surface models are too complicated to re-implement in ASPECT. Couple with an external code: FastScape



Time discretization

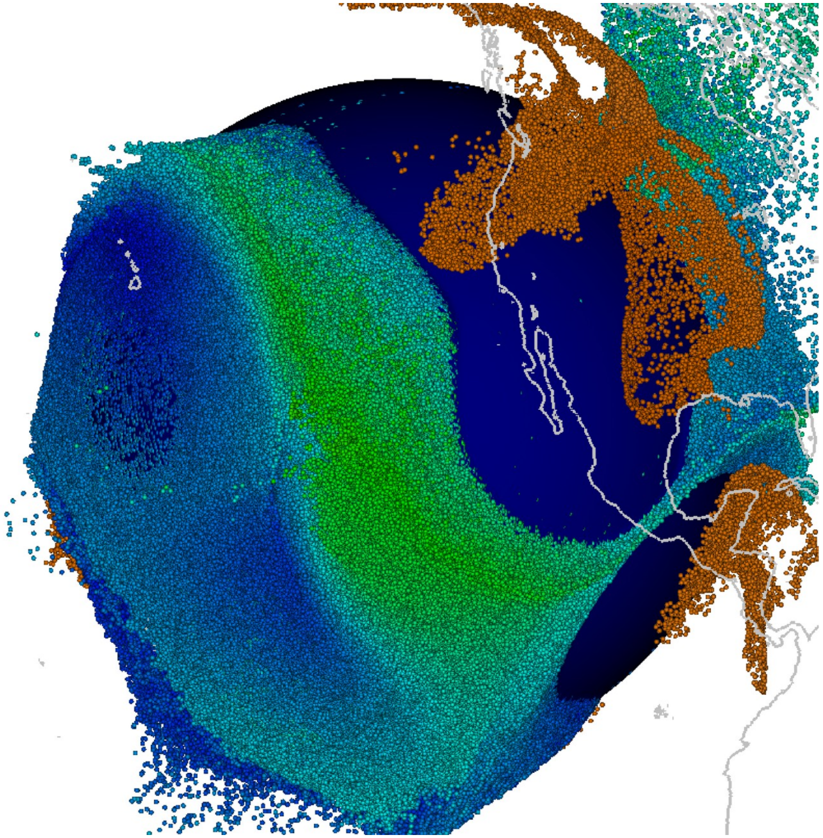
Overall algorithm:

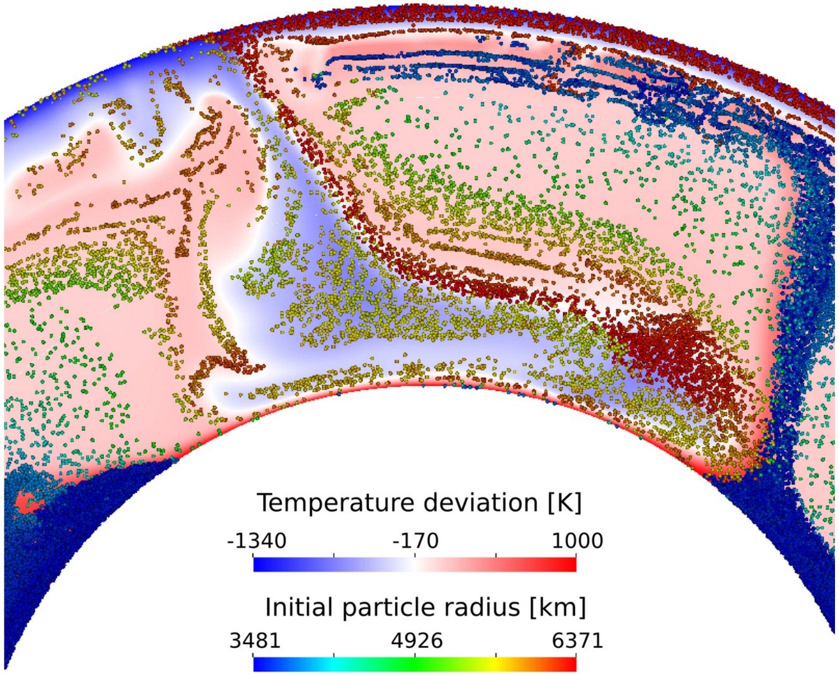
While ($T<T_{end}$):

- Solve Stokes equation
- From velocity, compute time step
- Solve for temperature field with implicit BDF2
- Solve for compositional field 1, implicit BDF2, operator splitting
- …
- Solve for compositional field 1, implicit BDF2, operator splitting
- Solve for surface deformation, couple with FastScape
- Advance time

Part VI:

What then? – Particles

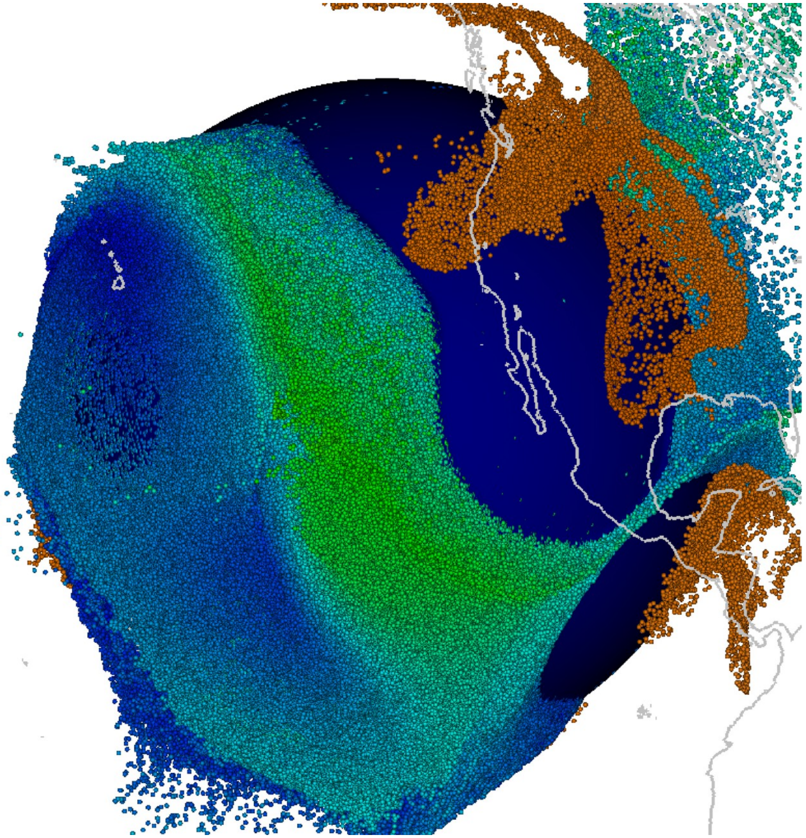


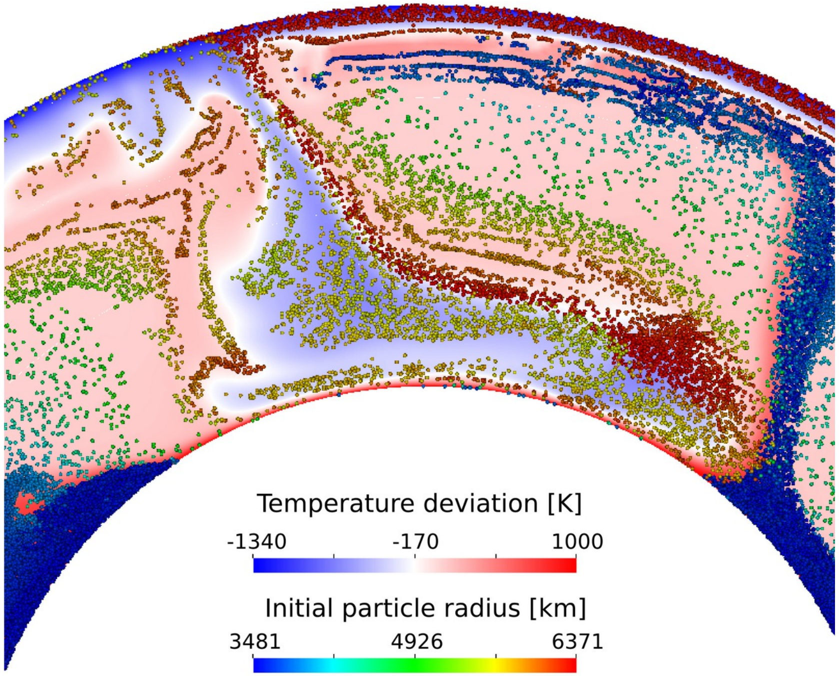


Compositional fields

Rene Gassmoeller comes along (2013?):

Fields are expensive. We want to track particles move along with the flow.





Free surfaces

Equations now:

$$-\nabla\cdot[2\eta\epsilon(u)]+\nabla p=g\rho(T)$$

$$\nabla\cdot u=0$$

$$\frac{\partial T}{\partial t}+u\cdot\nabla T-\kappa\Delta T=\gamma+\alpha\left(\frac{\partial p}{\partial t}+u\cdot\nabla p\right)+\eta(\nabla u)^2$$

$$\frac{\partial c_1}{\partial t}+u\cdot\nabla c_1-\kappa\Delta c_1=q_1(u,p,T,\vec{c})$$

$$\dots$$

$$\frac{\partial c_N}{\partial t}+u\cdot\nabla c_N-\kappa\Delta c_N=q_N(u,p,T,\vec{c})$$

$$\frac{\partial h}{\partial t}-A\Delta h=r(u,p)$$

$$\boxed{\frac{dx_i(t)}{dt}=u(x_i(t)) \qquad \frac{dp_{i,j}(t)}{dt}=s(u,p,T,\vec{c},\vec{p}_i)}$$

Compositional fields

Considerations:

- Velocity affects particle locations
- Sometimes particle properties affect flow equations

- Computationally quite different from field-based methods
- Efficiency requires CFL<=1 → Particles transported at most one cell per time step
- Evaluation of rhs is *very* expensive

- Do one explicit Euler/RK4 step per (macro) step

Time discretization

Overall algorithm:

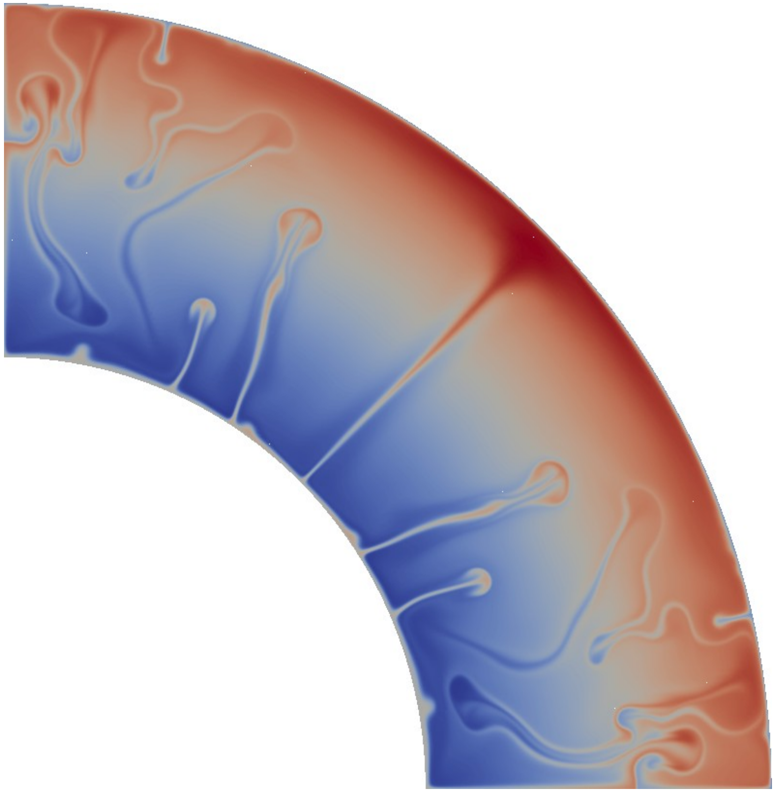
While ($T<T_{end}$):

- Solve Stokes equation
- From velocity, compute time step
- Solve for temperature field with implicit BDF2
- Solve for compositional field 1, implicit BDF2, operator splitting
- …
- Solve for compositional field 1, implicit BDF2, operator splitting
- Solve for surface deformation, couple with FastScape
- Advance cell positions and properties
- Advance time

Conclusions

ASPECT has turned out to be a very good tool to do interesting science!

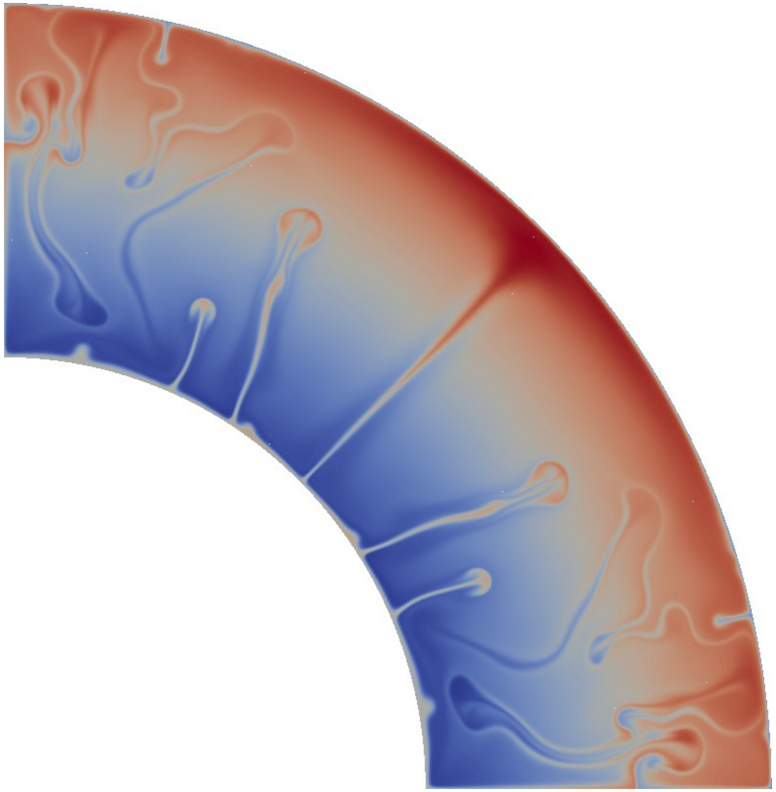
- We can produce lots of colorful pictures → something must be right!
- ***But is this the right approach?***



Questions I don't know the answer to

Question 1: What would be the costs of a better approach?

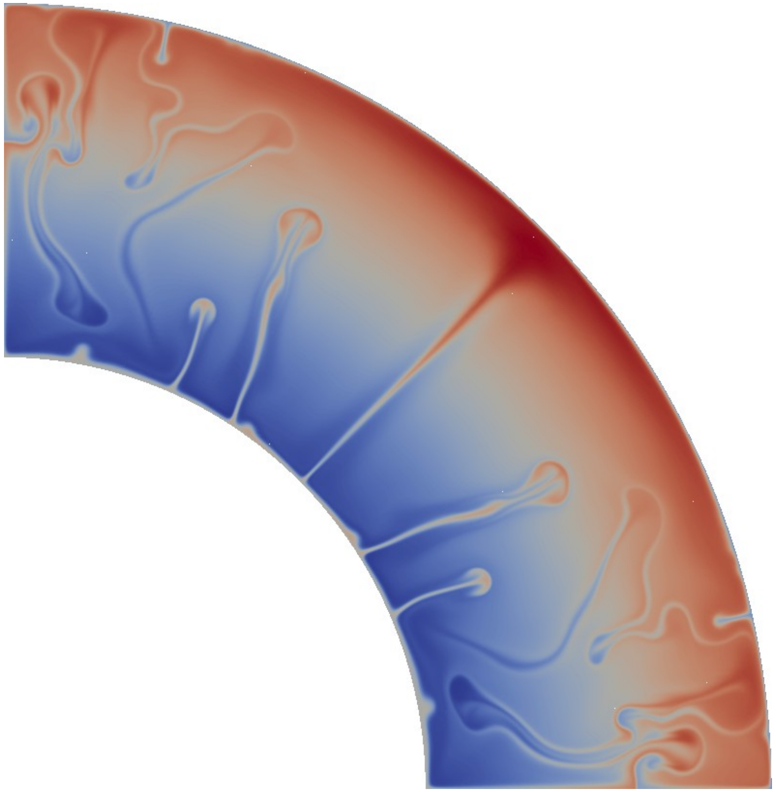
- Coupled solvers will require more memory, likely more computations
- How would one even approach integrating multiple modalities (solving in the bulk, on the surface, particles, external codes)?



Questions I don't know the answer to

Question 2: What would be the benefits of a better approach?

- How big is the time discretization error?
- What is the dominant contribution to the overall error?
- What is the *error* anyway?



Conclusions

Aspect – Advanced Solver for Problems in Earth's ConvecTion:

http://aspect.geodynamics.org/

References:
M. Kronbichler, T. Heister, W. Bangerth:
High accuracy mantle convection simulation through modern numerical methods.
Geophysics Journal International, 2012.

T. Heister, J. Dannberg, R. Gassmoeller, W. Bangerth:
High accuracy mantle convection simulation through modern numerical methods. II: Realistic models and problems
Geophysics Journal International, 2017.