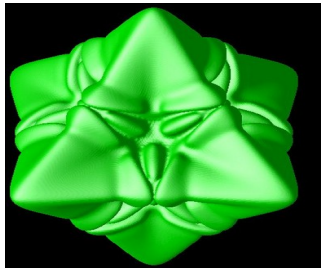
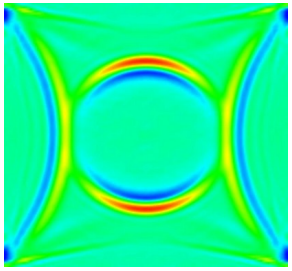
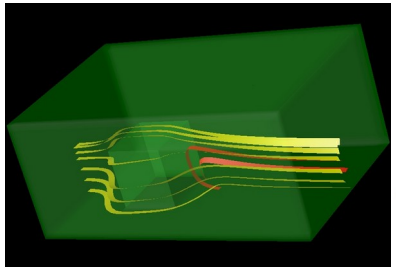
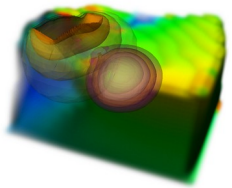
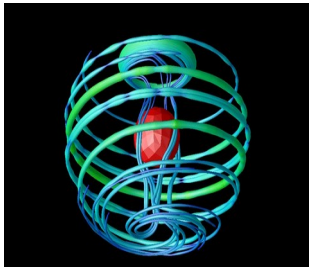
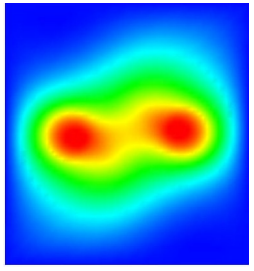
Complex simulations with the deal.II open source software, and how we use Trilinos for that

Wolfgang Bangerth
Colorado State University

With the help of many other deal.II contributors

A “typical” application in CS&E

What we generally need:

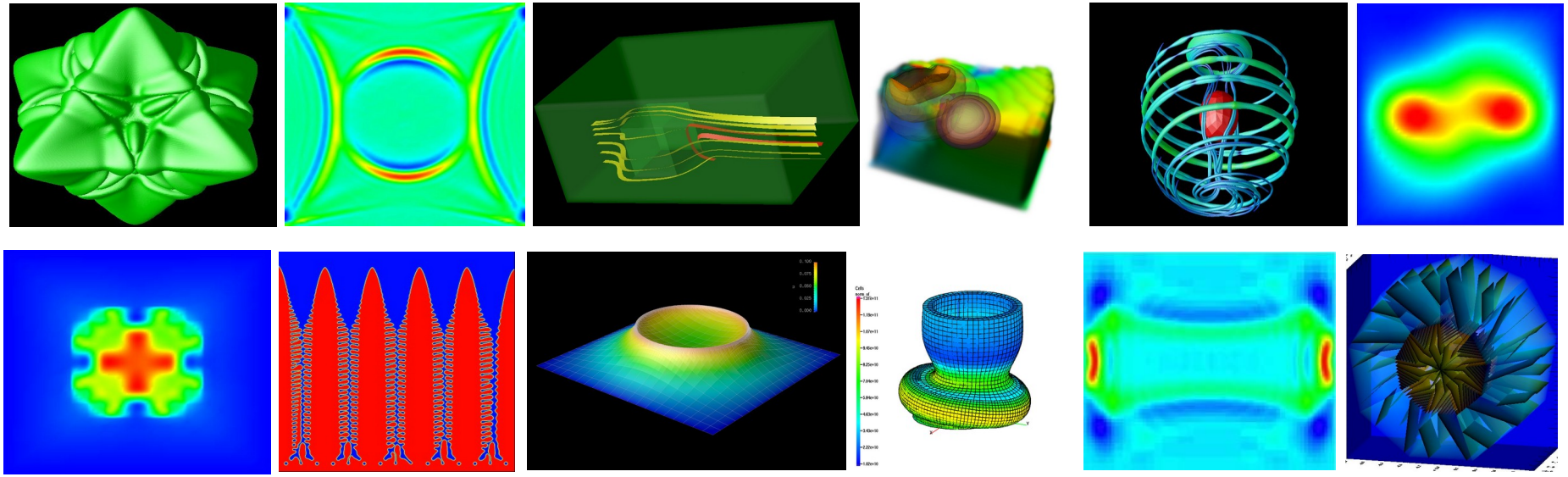
- Non-trivial 2d/3d geometry
- Interfaces to CAD and meshers
- Coupled systems of nonlinear PDEs
- Efficient non-linear iteration strategy
- Efficient linear solver
- Ways to visualize the solution

What we may need:

- Parallel execution on small and large systems
- Mixed or higher order finite elements
- …

deal.II

A library for finite element computations that supports...



...a large variety of PDE applications tailored to non-experts.

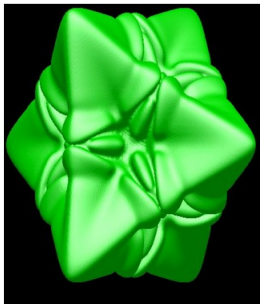
www.dealii.org

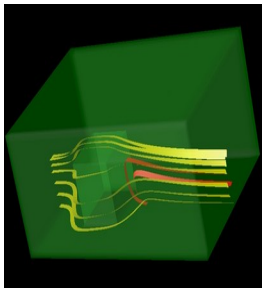
Fundamental premise:
Provide building blocks that can be used in many different ways, not a rigid framework.

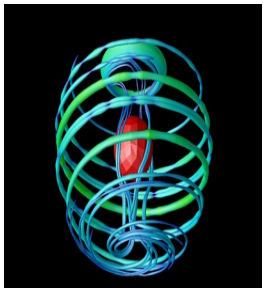
deal.II

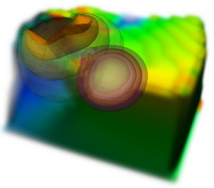
deal.II provides:

- Adaptive meshes in 1d, 2d, and 3d
- Tri/quad (2d) and tet/hex meshes (3d)
- Interfaces to all major graphics programs
- Standard refinement indicators
- Most standard finite element types (continuous, discontinuous, low and high order)
- Support for multi-component problems
- Its own sub-library for dense + sparse linear algebra
- Interfaces to PETSC, Trilinos, UMFPACK, ARPACK, OpenCASCADE, ...
- Supports multicore + cluster systems





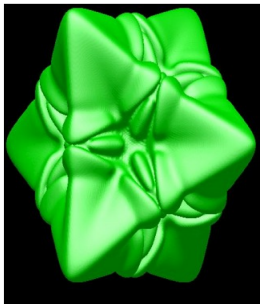


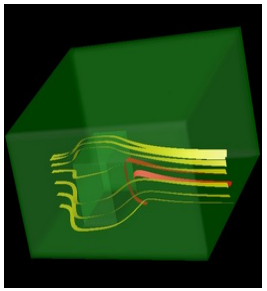


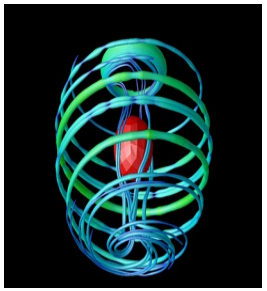
deal.II

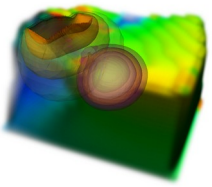
Status today:

- ~1000 downloads per month
- 1.4M lines of C++ code
- 10,000+ pages of documentation
- Portable build environment
- Used in teaching at many universities









- ~300 people have contributed to it
- ~40 people contribute to each release
- ~10 pull requests merged *each day*
- ~1 new publication *per day* that uses it

www.dealii.org

Examples

Examples of what can be done with deal.II (2013 only):

- Biomedical imaging
- Brain biomechanics
- E-M brain stimulation

- Microfluidics
- Oil reservoir flow
- Fuel cells
- Transonic aerodynamics
- Foam modeling
- Fluid-structure interactions
- Atmospheric sciences

- Quantum mechanics
- Neutron transport
- Nuclear reactor modeling

- Numerical methods research

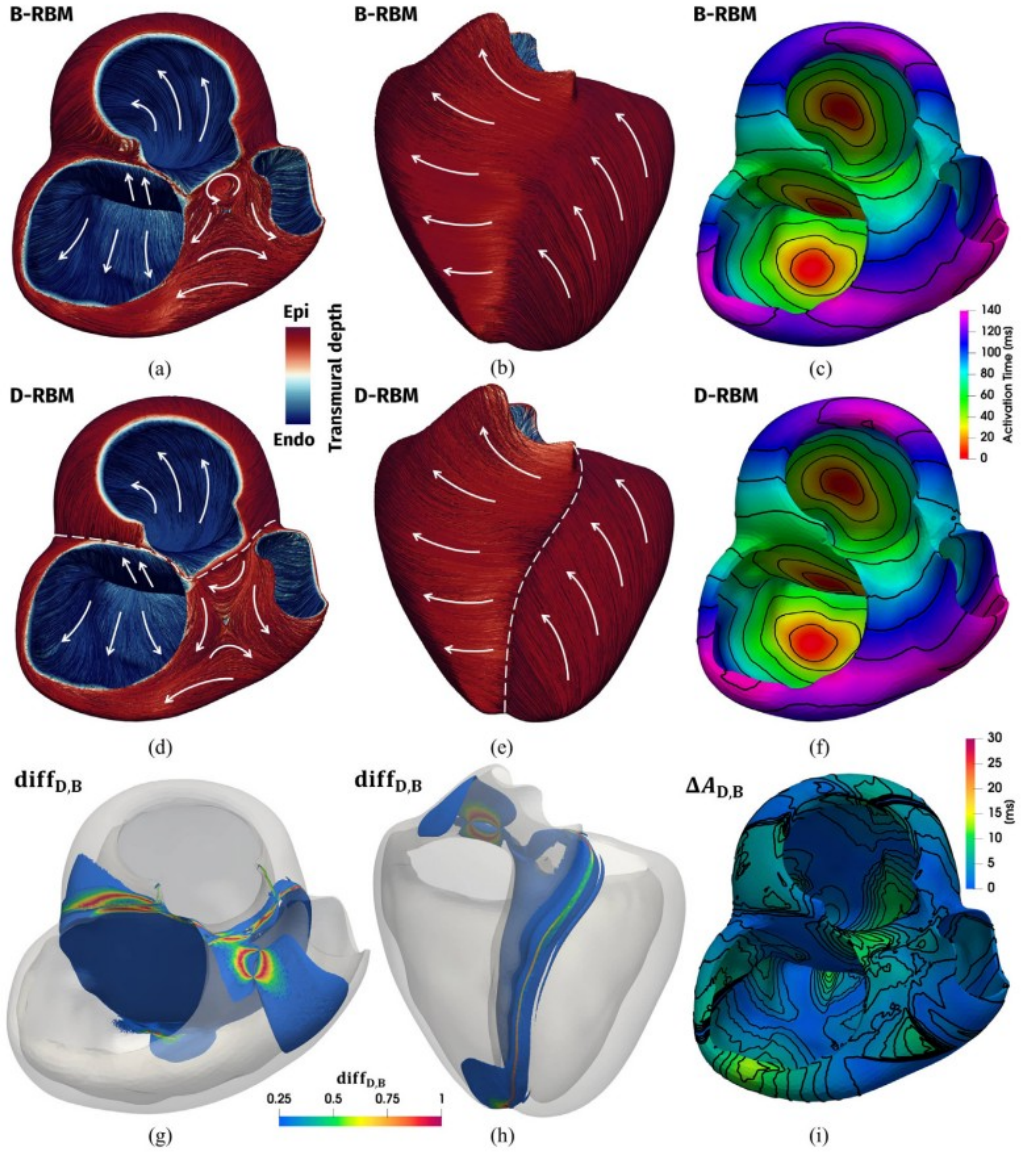
- Fracture mechanics
- Damage models
- Solidification of alloys
- Laser hardening of steel
- Glacier mechanics
- Plasticity
- Contact/lubrication models

- Electronic structure
- Photonic crystals

- Financial modeling

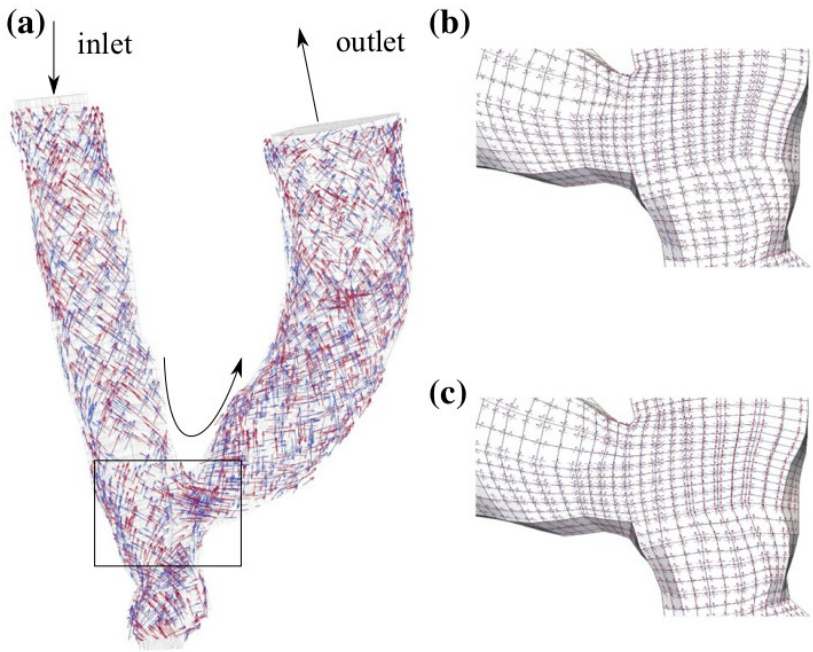
- Chemically reactive flow
- Flow in the Earth mantle

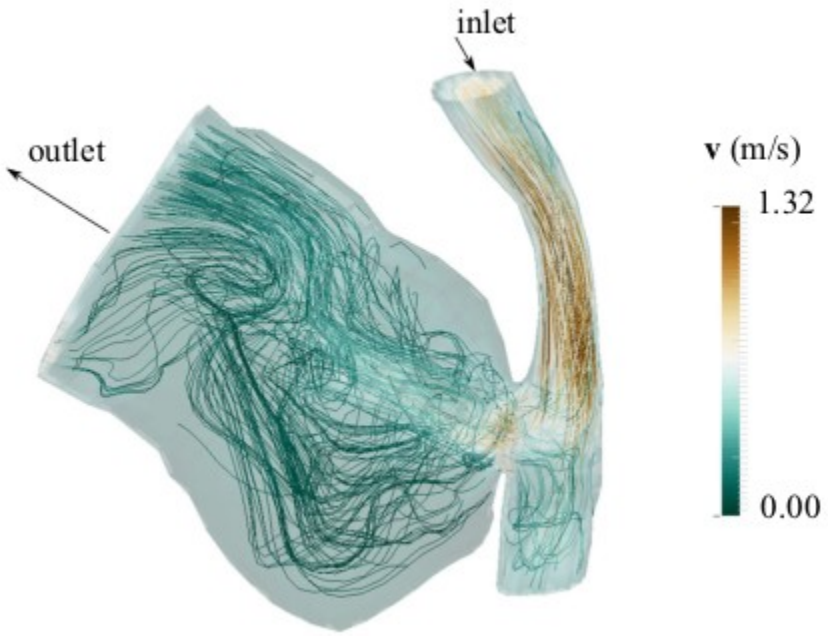
Example applications: Electromechanics



Electrophysiology of heart muscle fibers
(Piersanti et al., 2020)

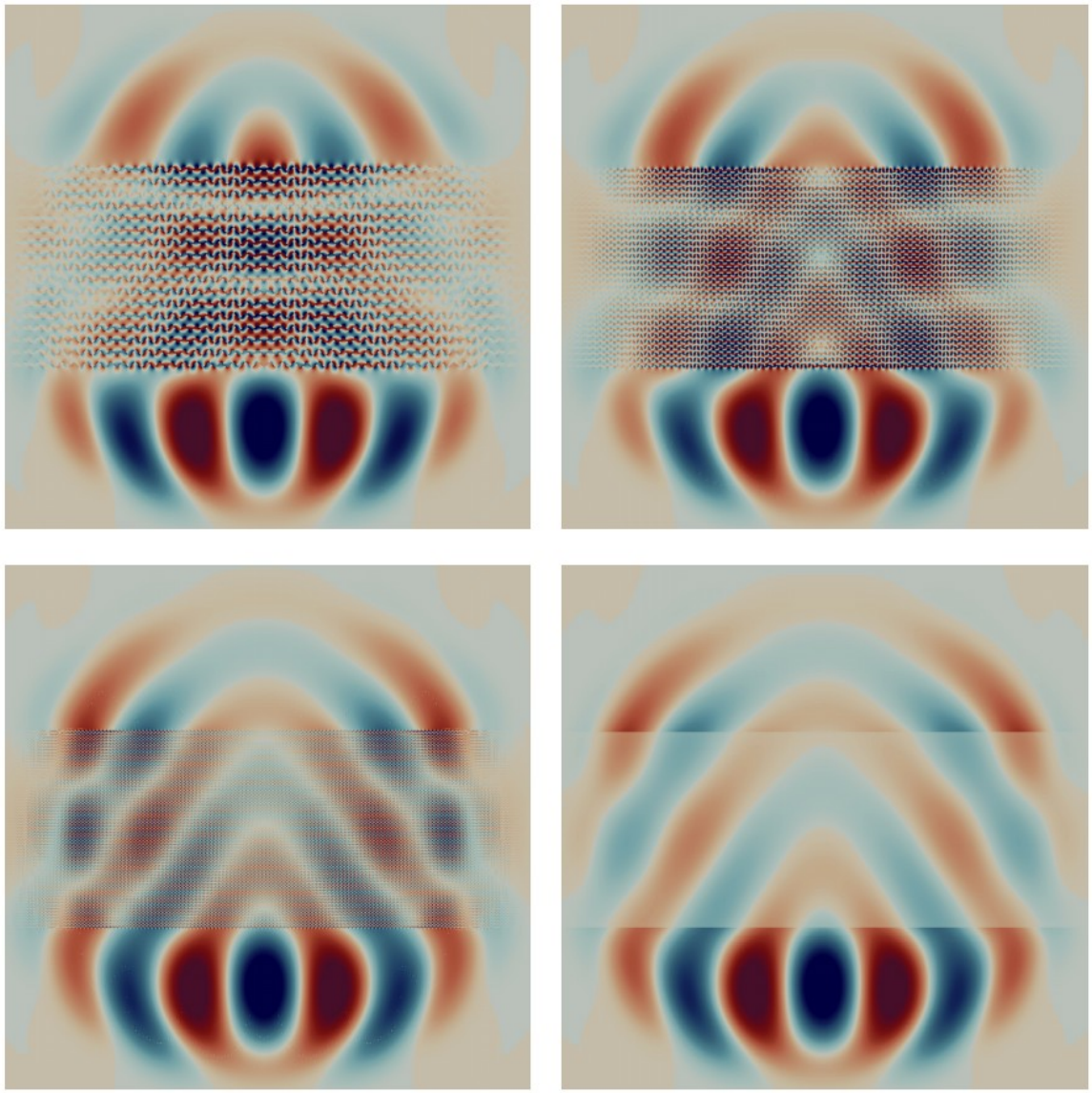
Example applications: Complex models





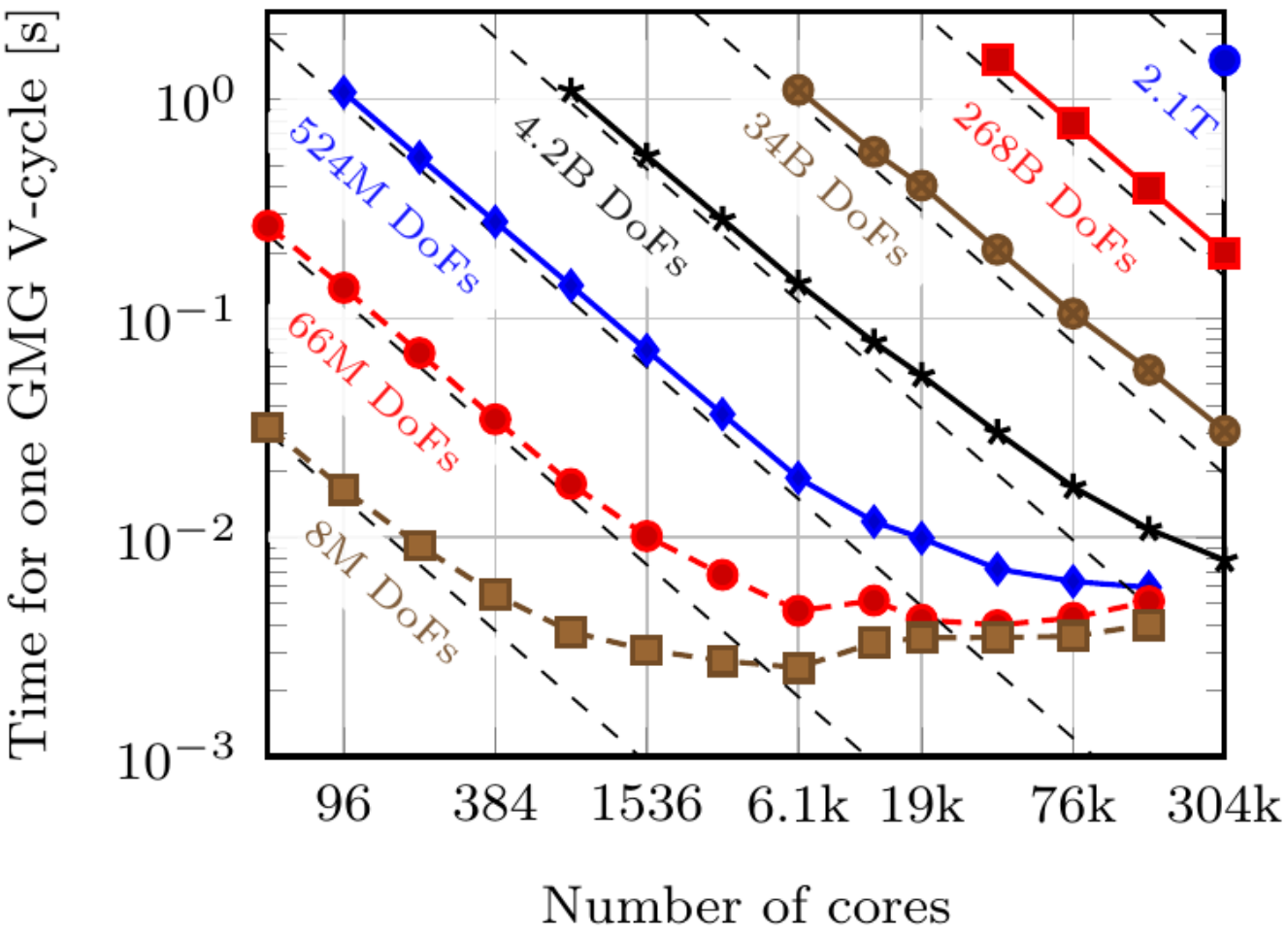
Patient-specific fluid-structure interaction with realistic material models for vascular modeling
(de Villiers, McBride, Reddy, Franz, Spottiswoode, 2018)

Example applications: Complex models



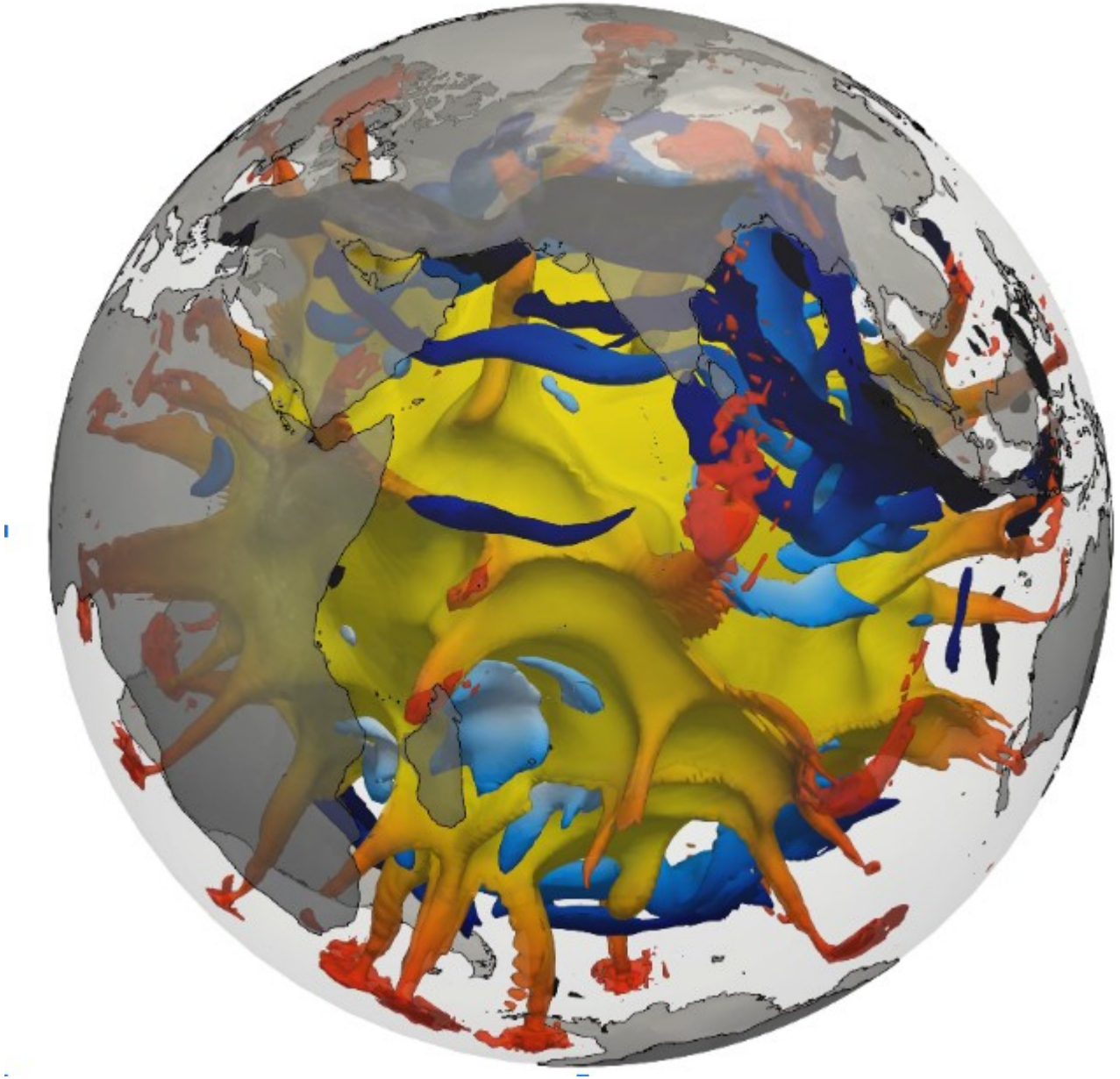
Homogenization of models for plasmonic crystals
(Maier et al., 2019)

Parallelization on parallel machines



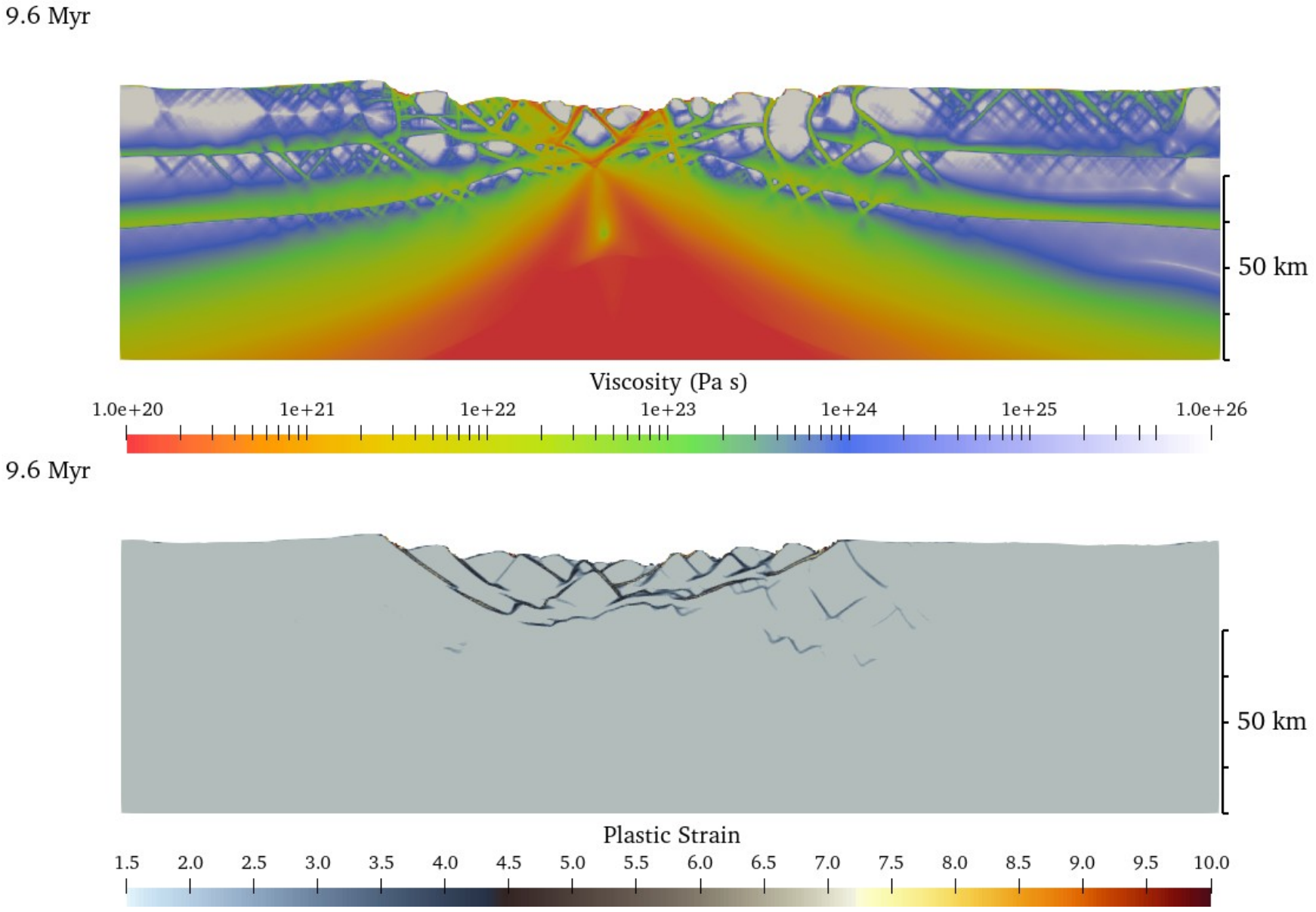
Scaling of a multigrid solver on up to $2*10^{12}$ unknowns on up to 300,000 processor cores.

Example applications: Complex models



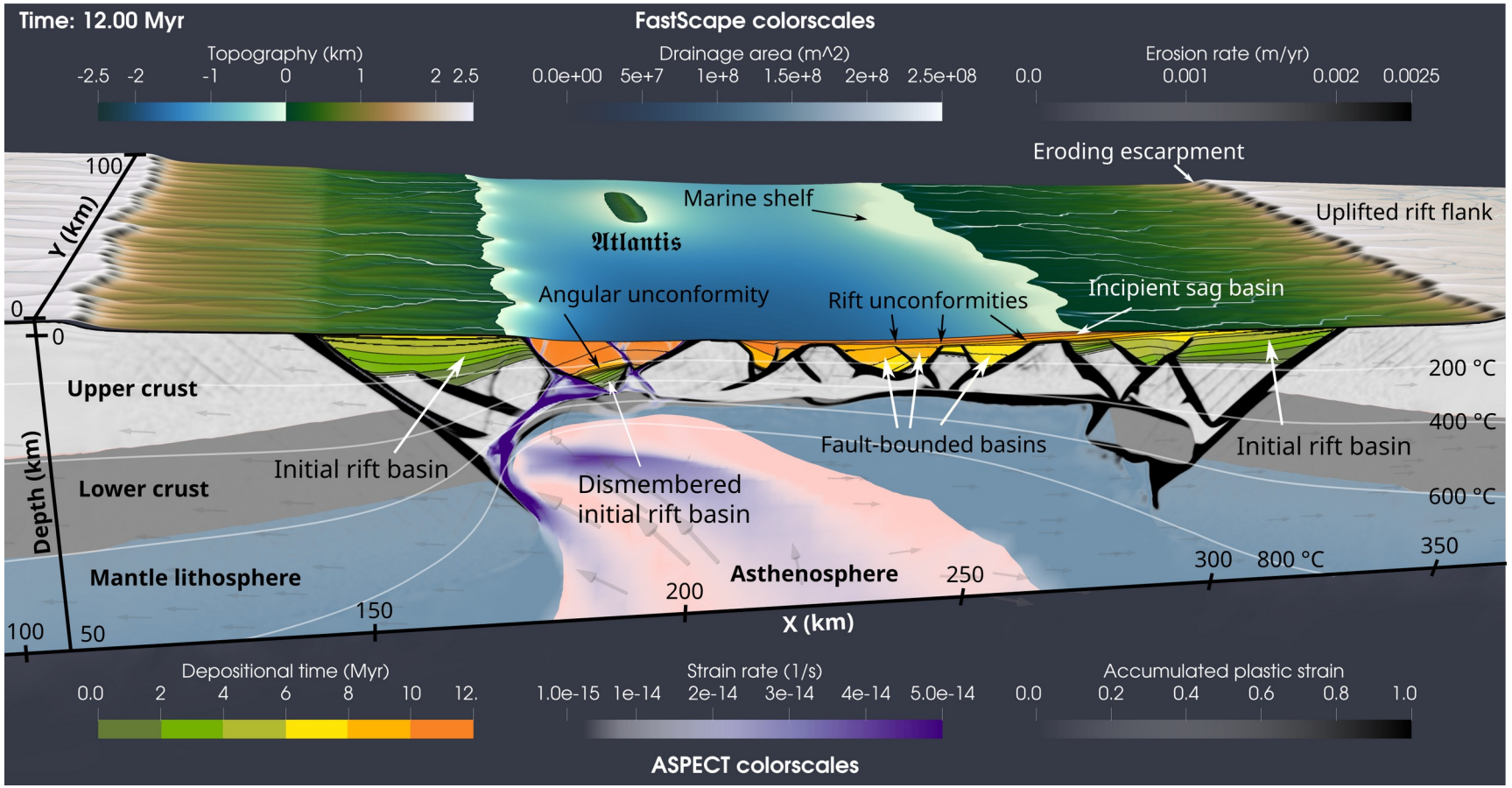
Flow in the earth interior
(Advanced Solver for Problems in Earth Convection, ASPECT)

Example applications: Complex models



Flow in the earth interior
(Advanced Solver for Problems in Earth Convection, ASPECT)

Example applications: Complex models



Flow in the earth interior
(Advanced Solver for Problems in Earth Convection, ASPECT)

Interfacing with other software

We have learned that it is best if deal.II focused on finite element methods.

Other functionality is obtained by **interfacing with other software**:

- Preprocessing:
  - Assimp
  - CGAL
  - Gmsh
  - OpenCASCADE

- Postprocessing:
  - HDF5
  - VTK-based visualization software

Interfacing with other software

We have learned that it is best if deal.II focused on finite element methods.

Other functionality is obtained by **interfacing with other software**:

- General algorithm support:
  - AdolC
  - ArborX
  - BOOST
  - GSL
  - HDF5
  - Metis
  - MPI
  - muparser
  - p4est
  - SUNDIALS
  - Symengine
  - TBB

Interfacing with other software

We have learned that it is best if deal.II focused on finite element methods.

Other functionality is obtained by **interfacing with other software**:

- Linear algebra:
  - ARPACK
  - BLAS/LAPACK
  - cuSolver/cuSparse
  - Gingko
  - MUMPS
  - PETSc (Vec, Mat, KSP functionality)
  - Scalapack
  - Trilinos (various packages)
  - UMFPACK

Trilinos Epetra interfaces

deal.II's Trilinos interfaces were originally written in the late 2000s.

At the time intended to complement our PETSc interfaces.

Goal: Scalable data structures, solvers, preconditioners for large linear systems.

→ **Built on Epetra**

A good decision: We are regularly running jobs efficiently on 1000s and occasionally on 10,000s of cores.

Trilinos Epetra interfaces

What we do with Epetra:

We don't actually need very much:

– Epetra_CrsGraph
– Epetra_FECrsMatrix
– Epetra_FEVector
+ some custom import/export functionality

– AztecOO solvers (CG, GMRES, …)
– Ifpack preconditioners

– ML
– MueLu via its Epetra interfaces

All of this is *wrapped* to provide a deal.II-style interface.

Trilinos Epetra interfaces

What we do with Epetra:

We combine these interfaces to build higher-order functionality for block systems.

For example, solve a Stokes system

$$\begin{pmatrix} A & B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} U \\ P \end{pmatrix} = \begin{pmatrix} F \\ G \end{pmatrix}$$

with a preconditioner of the form

$$\begin{pmatrix} A & B \\ 0 & S \end{pmatrix}^{-1} \approx \begin{pmatrix} \widetilde{A^{-1}} & B \\ 0 & M^{-1} \end{pmatrix} \quad \text{or} \quad \approx \begin{pmatrix} A^{-1} & B \\ 0 & M^{-1} \end{pmatrix}$$

- Implemented in deal.II's "Block*" classes
- Not dissimilar to PETSc's "field split" approach
- Just better

Trilinos Epetra interfaces

Problems we have with our Epetra interfaces:

- Epetra has been “end of line” for quite a while
- We have Tpetra wrappers, but they are not used

- We would love to always use Tpetra instead, but:
  - Switch-over has to be atomic
  - We don’t have the manpower
  - Interface stability was not clear for a while

Trilinos Epetra interfaces

Questions about Tpetra:

- We don’t know what functionality we would lose in a conversion.

  For example: Muelu only supports Epetra when Trilinos is not compiled with Tpetra

- Varying coding styles of Trilinos packages don’t make conversions easy
- Documentation is not always sufficient/helpful

But: We are working on switching. There is no alternative.

Interfaces to other packages

Sacado:

- Automatic differentiation package

- Used in a few tutorial programs to compute Jacobians from residuals
  – in fluid dynamics
  – in solid mechanics with complex energy functionals
- More tutorials coming

- Not originally wrapped by deal.II classes
- But there are now wrappers that make uniform whether you use Sacado, AdolC, or SymEngine

Interfaces to other packages

Zoltan:

- Mesh partitioning package
- As an alternative to METIS
- Not widely used because we typically partition meshes based on space-filling curves (via p4est)
- Has become more relevant because we now have mesh classes that do not rely on p4est
- Wrapped via a single deal.II function

Interfaces to other packages

NOX:

- Nonlinear solver package
- Patch is currently pending

- As an alternative to SUNDIAL's KINSOL package

- Wrapped with deal.II classes that closely mirror our KINSOL interfaces

Interfaces to other packages

ROL:

- Optimization package

- deal.II provides vector adaptors
- Usage unclear

Interfaces to other packages

SEACAS:

- Not actually quite sure what this package is
- We use it to read ExodusII files

- deal.II provides a single function as a wrapper

Kokkos

We have decided to use Kokkos!

- Trilinos requires Kokkos anyway
- deal.II will require it as well, regardless of whether deal.II is configured with Trilinos
- We think that many other applications will sooner or later use Kokkos

- It has functionality that we need but that is just too cumbersome to replicate – specifically how to deal with GPUs.

- **We are concerned about Trilinos bundling/using its own Kokkos version, rather than the system one**

Summary

deal.II is a toolbox to write finite element solvers with.

- Supports nearly every aspect of finite element methods
- Highly scalable to large systems/computations
- Widely used for applications in many engineering and science disciplines.

- Has interfaces to quite a large number of Trilinos packages
- These interfaces have provided us with great functionality for more than 15 years already – thank you!