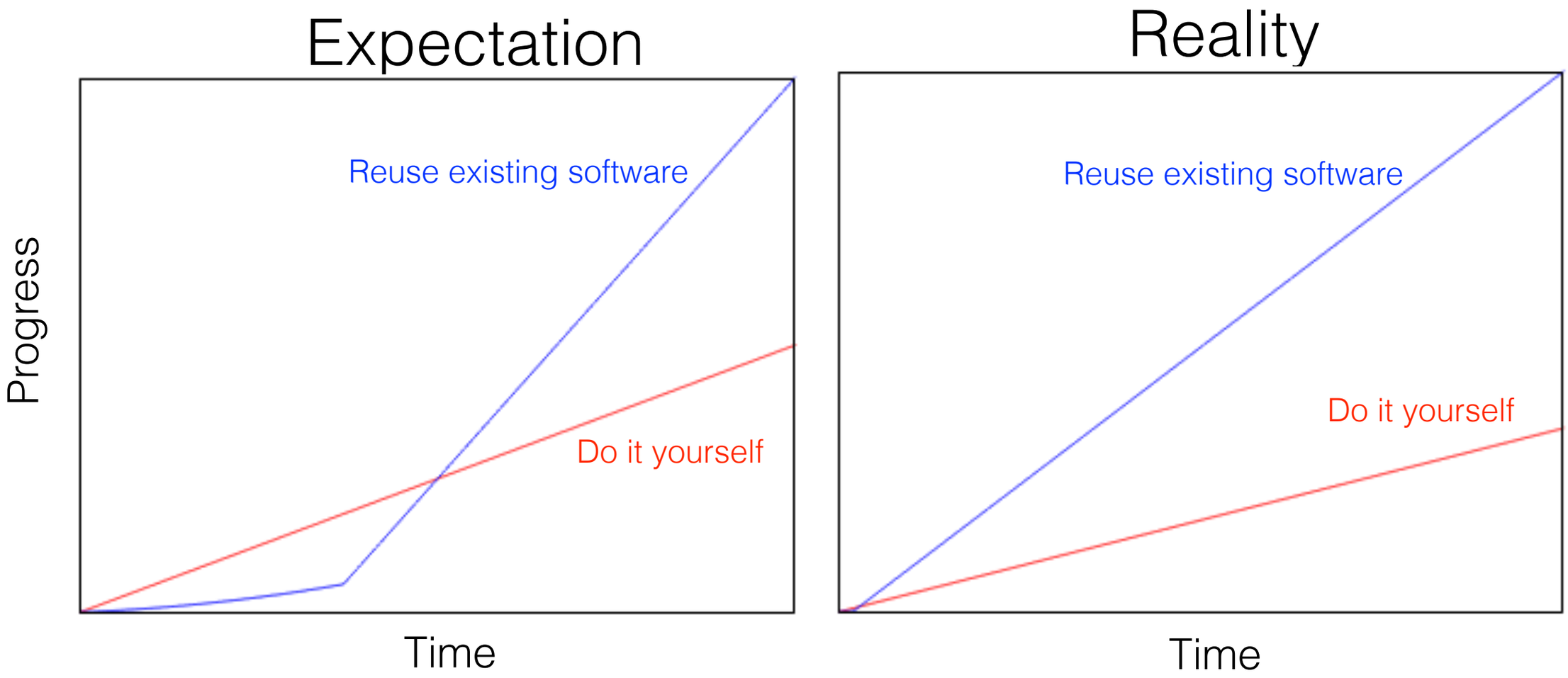
Introduction to deal.II

Jean-Paul Pelteret (jean-paul.pelteret@fau.de)
Luca Heltai (luca.heltai@sissa.it)

19 March 2018

Why use deal.II (or any other PDE toolbox)?

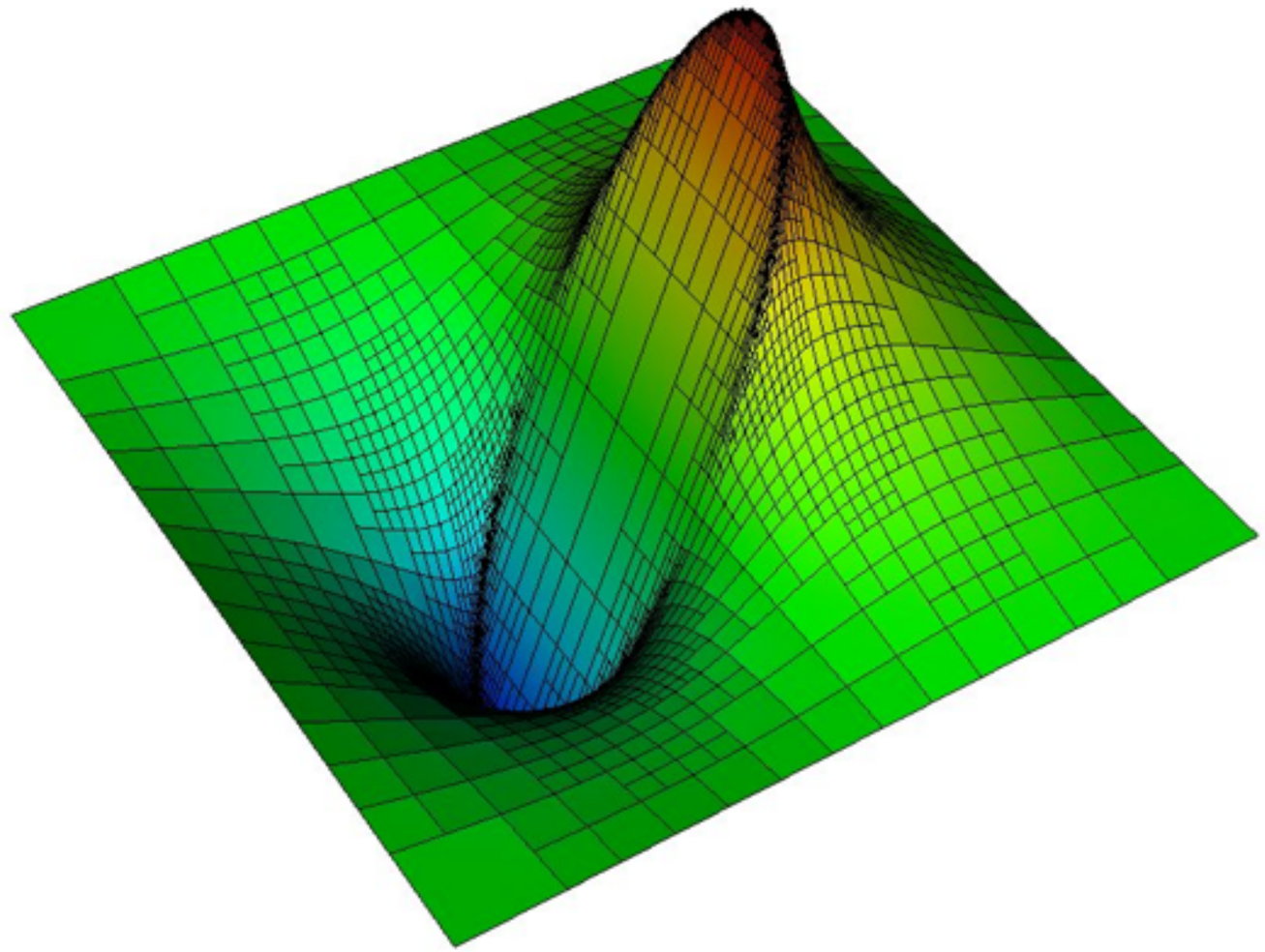


- Applies to:
  - Users
  - Developers
- “The secret to good scientific software is (re)using existing libraries”

What is deal.II?

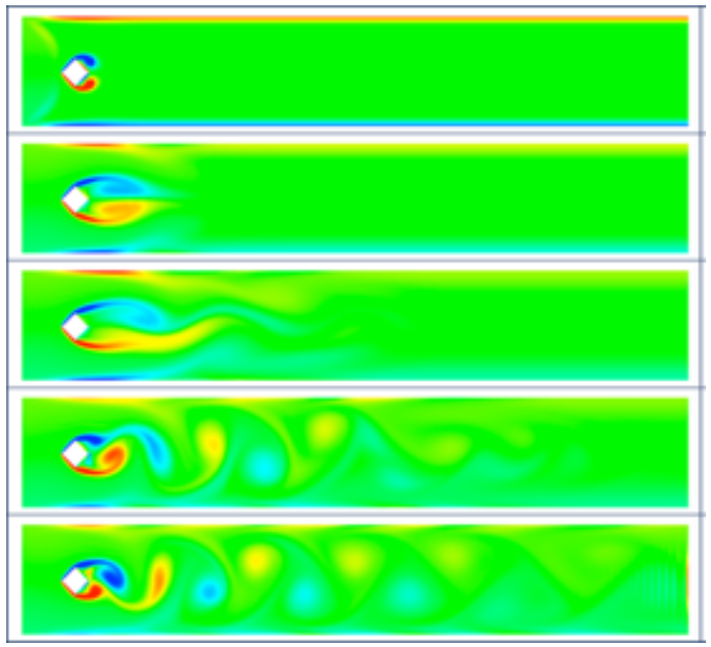
Differential Equation Analysis Library

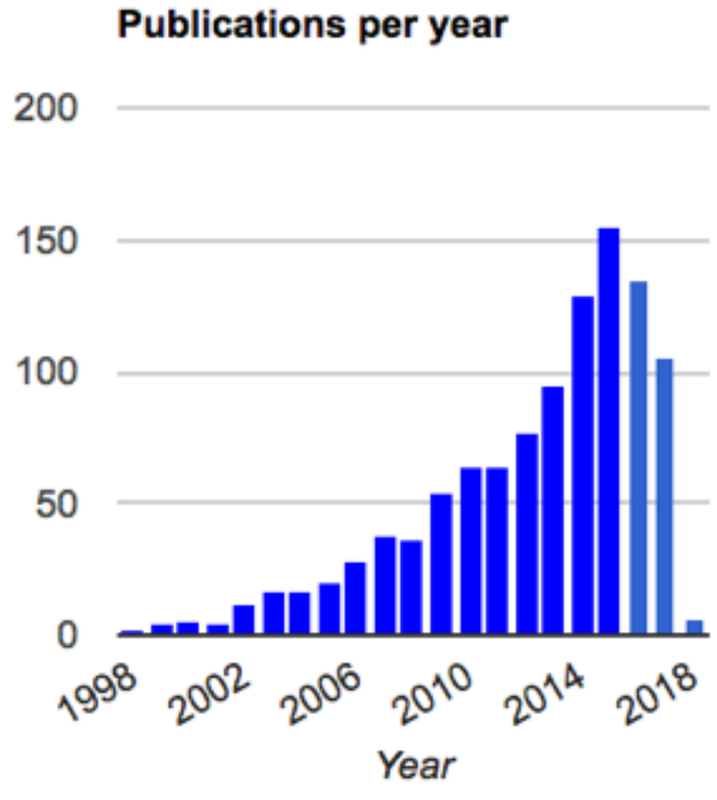
- Flexible open-source finite element toolkit
    - All the support functionality required to describe and solve a FE problem (PDEs)
    - Optimised for speed
    - Heavily tested
        - Many error checks (debug mode)
        - ~9500 regression tests run continuously
        - https://cdash.kyomu.43-1.org/index.php?project=deal.II
    - Part of SPEC CPU 2017 benchmark
- Templated C++ library (Object Orientated)
    - Dimension independent programming
- Portable
    - OS, architecture, compiler
- Origins
    - Study mesh adaptivity and error estimation
    - Now used in many other applications, frameworks



What is deal.II?

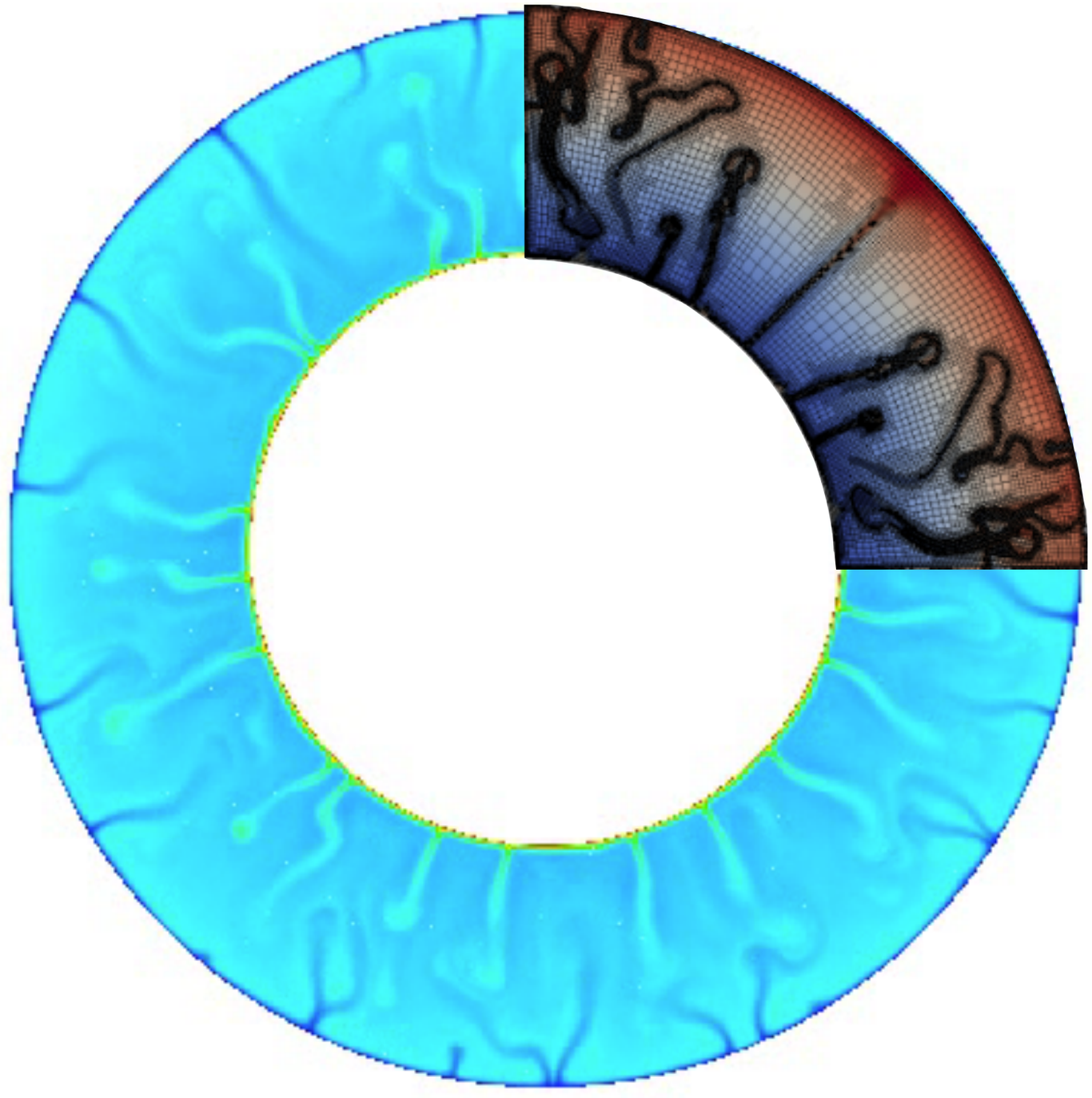
- Heavily documented
  - Over 10000 pages of interface documentation
  - Numerous tutorials
    - Illustrate functionality
    - Present methods to solve problems
- Quite widely used, and growing
- Active community
  - Approachable developers
  - Helpful online forum





Classes of problems solved using deal.II

- Geomechanics
- Fluid and gas dynamics
- Porous media
- Fluid-structure interaction
- Boundary element method
- Topology optimisation
- Medical image reconstruction
- Structural mechanics
- Biomechanics
- Crystal growth
- Gradient and crystal plasticity
- Contact mechanics
- Continuum-atomistic coupling
- Magneto- and electro-elasticity
- Thermo-plasticity

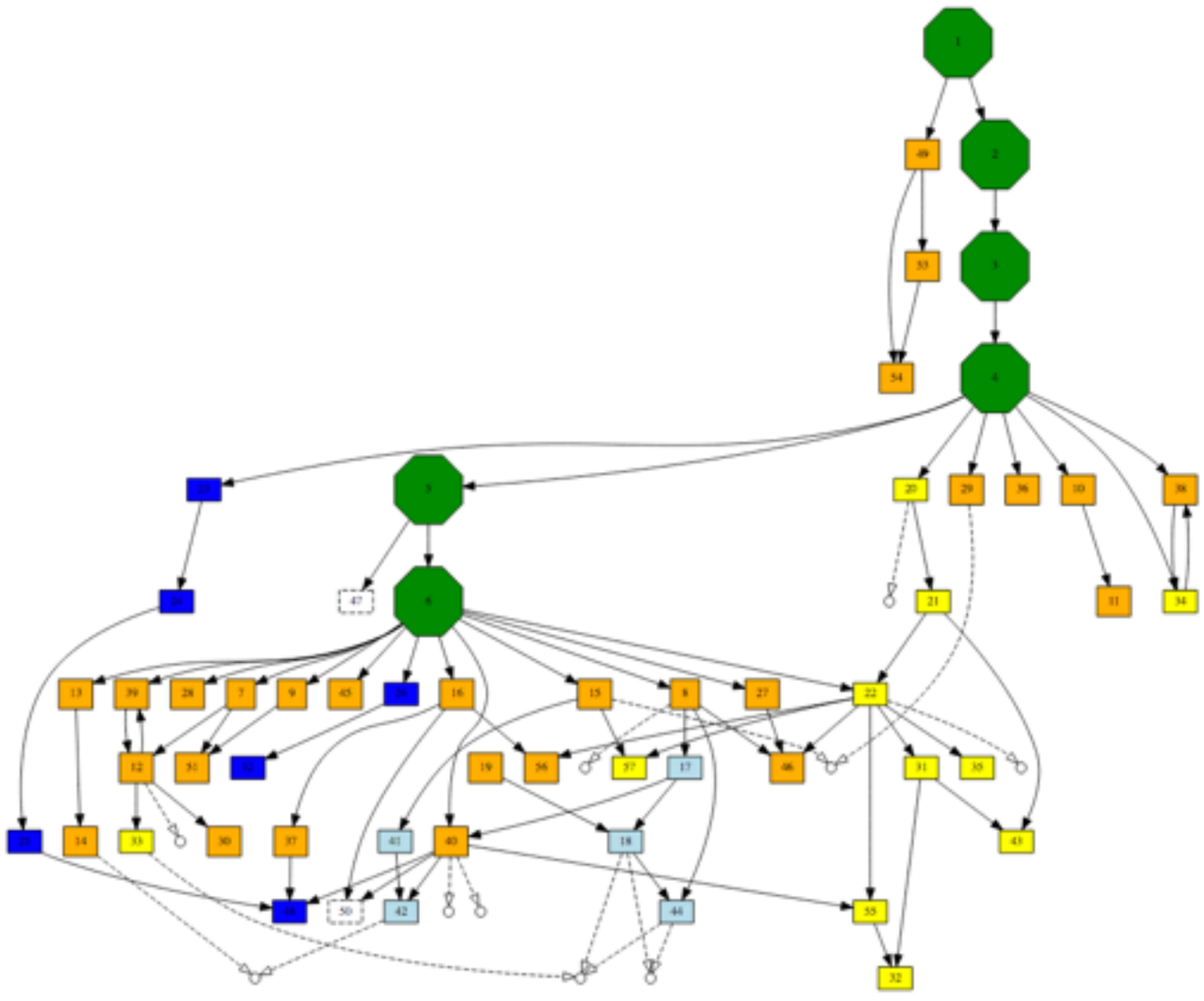


What deal.II is not

- A black box
  - Can’t throw any problem at it
  - Won’t do anything more than you ask it to
- Knows little* about
  - Numerical methods
  - Problem-specific implementation
    - Finite-element perspective
    - Physical conditions

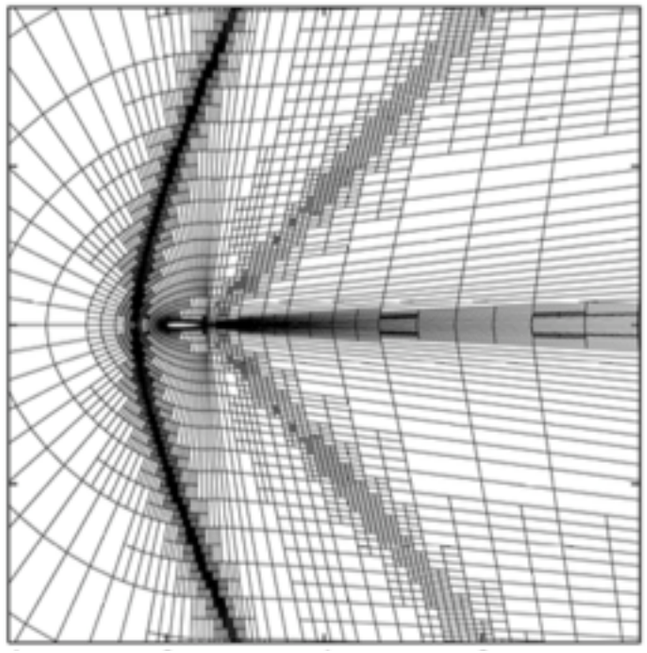
How deal.II will help you

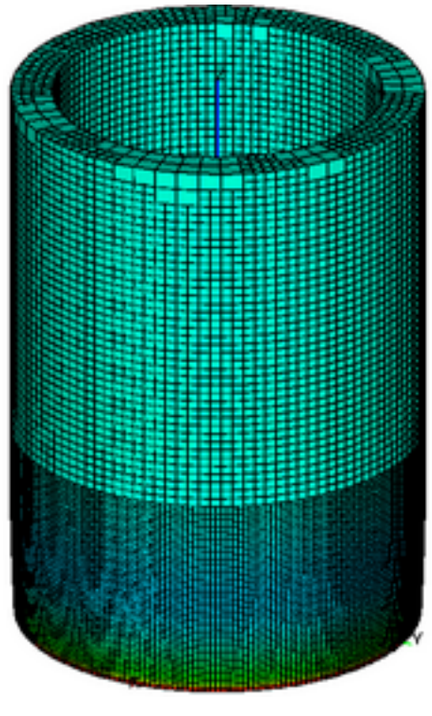
- Unified and well thought out data structure
  - Problem implementation
- Many tutorials
  - Baseline from which to build on
  - Demonstrate how to use features
- Comprehensive debugging support
  - Error messages everywhere!
- Some built in numerical tools
- Integration with advanced frameworks
  - Nonlinear solvers
  - Time integrators

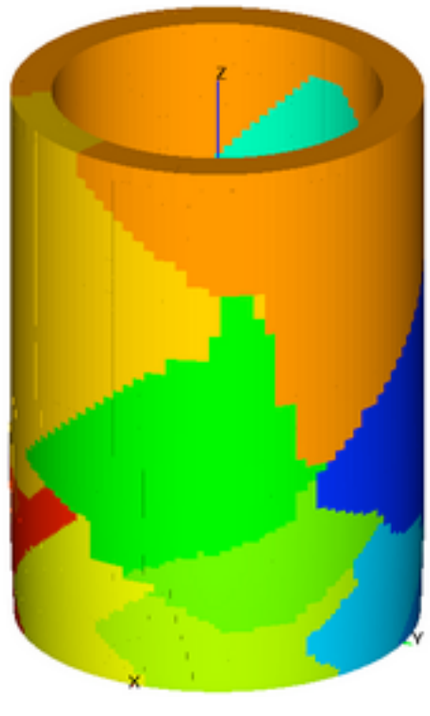


Fundamental capabilities and frameworks

- Mesh adaptivity
- Full and sparse linear algebra
  - Built in tensor, dense matrix/vector classes
  - BLAS and LAPACK integration; GSL
  - Built in linear solvers and preconditioners
  - Eigenvalue solvers
- Parallelisation
  - MPI
    - Linear algebra libraries (PETSc, Trilinos)
    - Distributed meshes ⟶ Billion DoF problems
  - Threading
  - Vectorised numbers (AVX extensions)
- Pre/post-processing

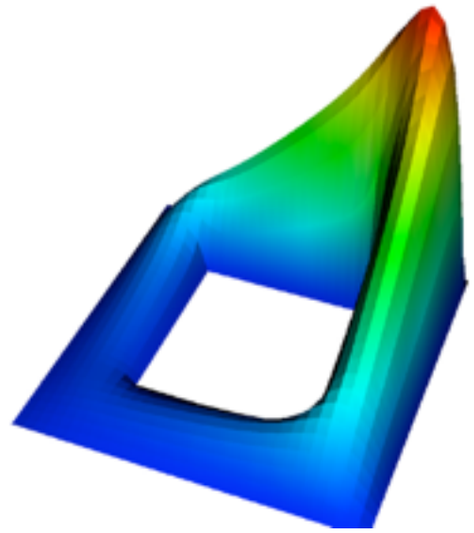


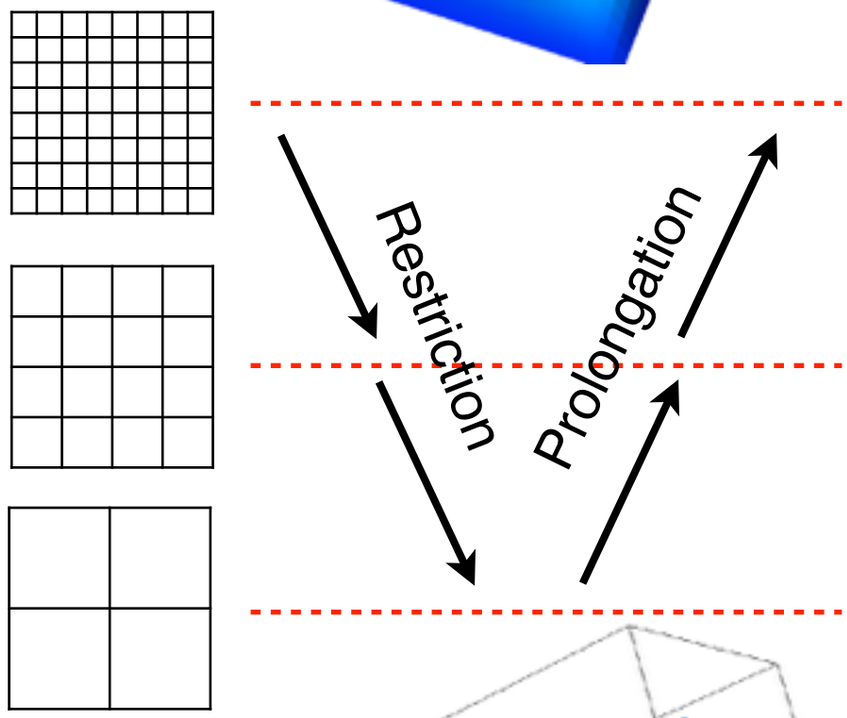


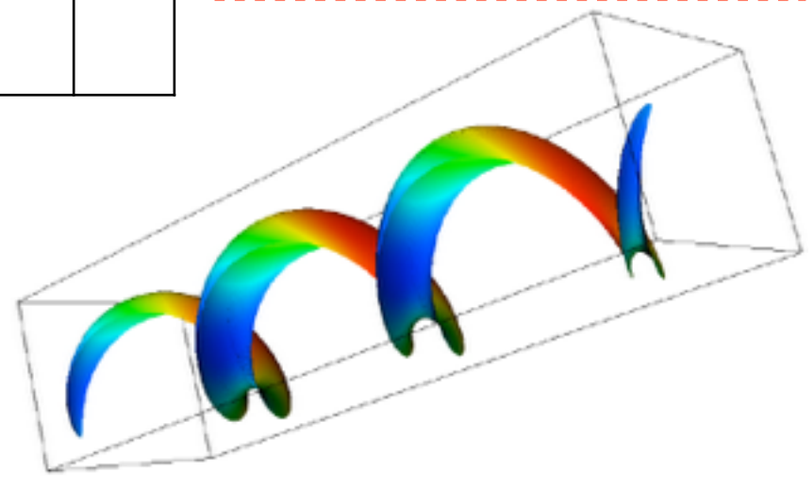


Advanced capabilities and frameworks

- hp-finite element support
- Meshworker
  - Assembly assistance
  - Functions to perform assembly for specific problem classes
- Geometric multi-grid
  - Using coarse grid as preconditioner to solution for finer grid
- Matrix-free
  - No explicit storing of matrix elements
  - Huge time-savings possible
- Charts and manifolds
  - Accurate description of topologically complex objects







How deal.II is developed

- Open repository on GitHub
  - https://github.com/dealii/dealii
- Anyone can contribute!
- We encourage all to participate