Numerical solution of PDEs using the Finite Element Method

Luca Heltai (luca.heltai@sissa.it)

16 March - 20 March 2020

MHPC
Master in High Performance Computing

Course schedule

Time	Duration	Content	Content
		MONDAY 16.03.2020	**TUESDAY 17.03.2020**
09:30	1.25 hours	Introduction First steps	Local refinement Hanging nodes - Part 1
		COFFEE / TEA	
11:15	1.25 hours	Introduction to FEM	Exercises, Q&A
		LUNCH	
14:00	1.25 hours	Solving Poisson's equation	Local refinement Hanging nodes - Part 2
		COFFEE / TEA	
15:45	1.25 hours	Exercises, Q&A	Exercises, Q&A
		WEDNESDAY 18.03.2020	**THURSDAY 19.03.2020**
09:30	1.25 hours	Shared memory parallelisation: Part 1	MPI parallelisation: Part 1
		COFFEE / TEA	
11:15	1.25 hours	Exercises, Q&A	Exercises, Q&A
		LUNCH	
14:00	1 hour	Shared memory parallelisation: Part 2	MPI parallelisation: Part 2
		COFFEE / TEA	
15:45	1.25 hours		Exercises, Q&A
		FRIDAY 20.03.2020	Project

Course goals

- Learn the fundamentals of deal.II
  - Commonly used data structures, their interface
  - Structure of finite element problems
  - Good implementation practices
  - Navigate the documentation
- Implement PDE solver (prototypical problem)
  - On complex Geometries
  - Using local refinement
  - Running in parallel (on both shared and distributed memory systems)

How the course will be run

- Each module will have a lecture
  - Present salient information
  - Put what we'll learn into context
- Then we'll walk through aspects of the tutorials/examples together
  - Discuss important functionality
    - What it does
    - How it works
    - Caveats and tips
- Remainder of the lecture will be spent doing some exercises/assignments
  - Suggestion: Work in groups of two/three
  - Continued at in the last session of the day

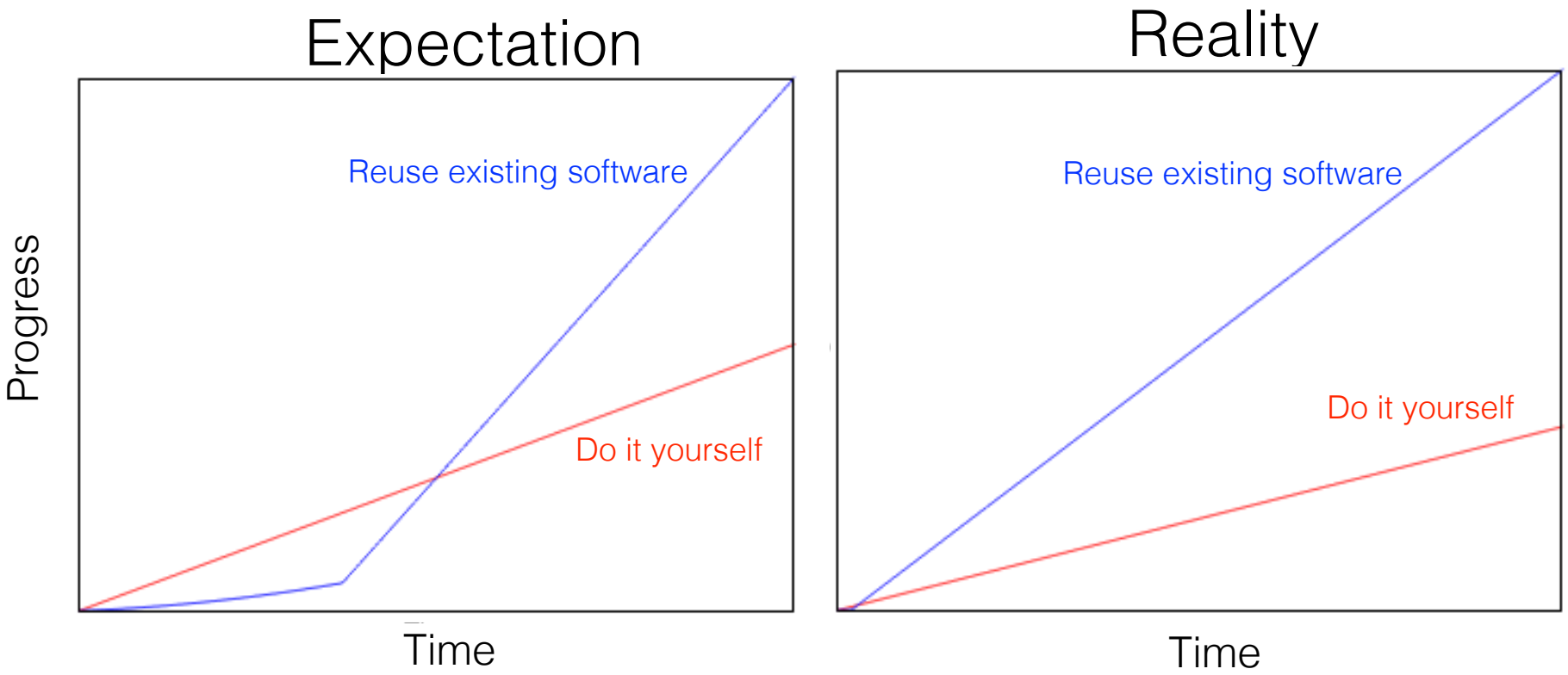
Resources

- deal.II user manual
  - https://www.dealii.org/current/doxygen/deal.II/index.html
  - https://www.dealii.org/current/doxygen/deal.II/modules.html
  - https://www.dealii.org/current/doxygen/deal.II/DEALGlossary.html
- deal.II tutorials and code gallery
  - https://www.dealii.org/current/doxygen/deal.II/Tutorial.html
  - https://www.dealii.org/current/doxygen/deal.II/CodeGallery.html
- Me :-)
  - Don’t hesitate to ask questions

Lecture 1: Introduction to deal.II (tools configuration)

Luca Heltai (luca.heltai@sissa.it)

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

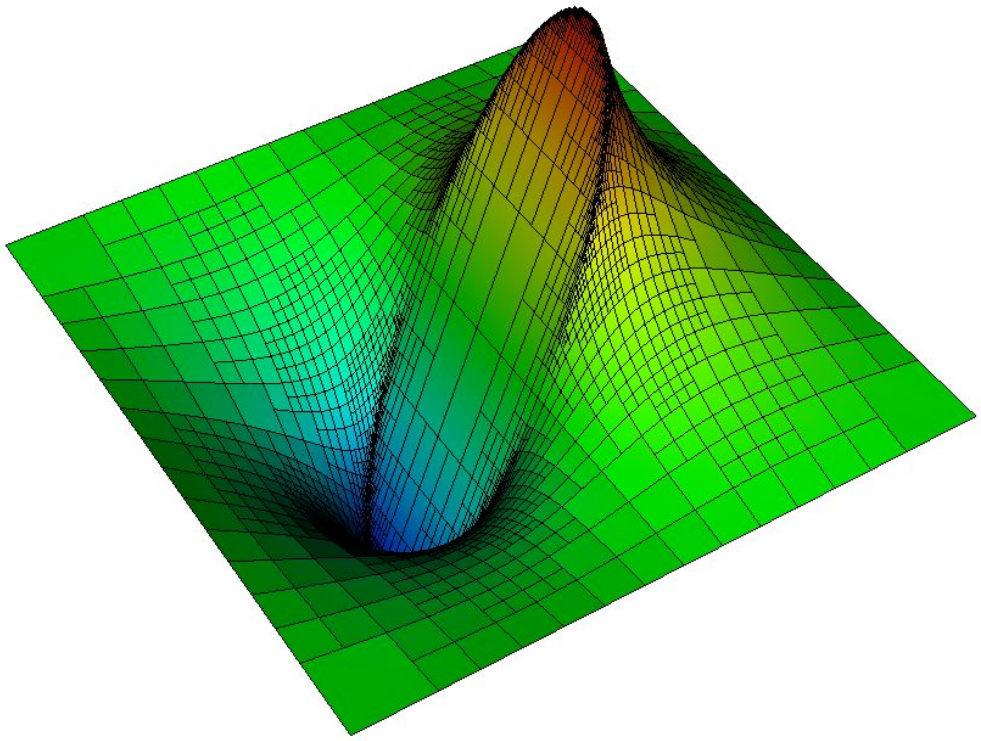


- Applies to:
  - Users
  - Developers
- “The secret to good scientific software is (re)using existing libraries”

Why deal.II?

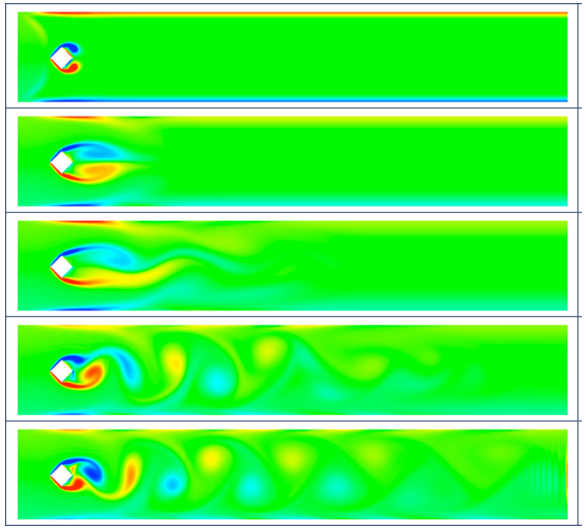
Differential Equation Analysis Library

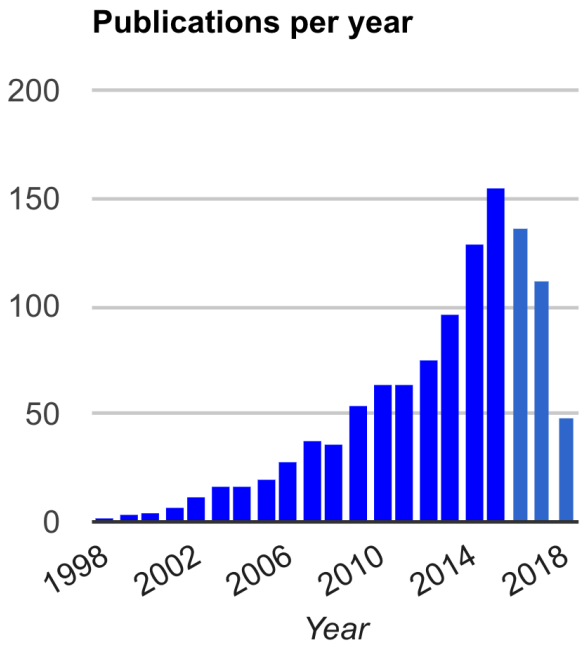
- Flexible open-source finite element toolkit
  - All the support functionality required to describe and solve a FE problem (PDEs)
  - Optimized for speed
  - Heavily tested
    - Many error checks (debug mode)
    - +10,000 regression tests run continuously
  - Part of SPEC CPU 2017 benchmark
- Templated 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



Why deal.II?

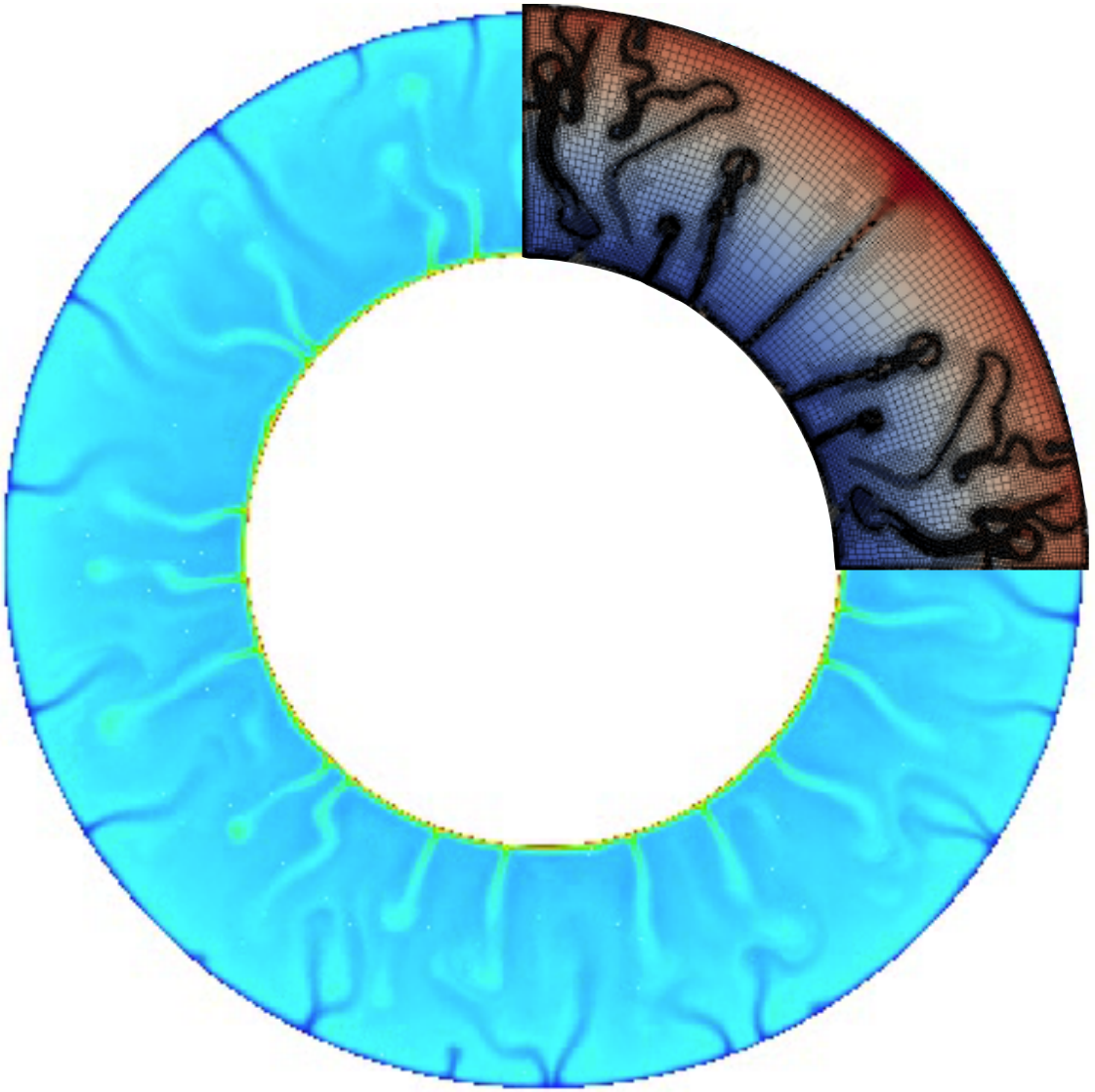
- Heavily documented
  - Over 10000 pages of interface documentation
  - Numerous tutorials
    - Illustrate functionality
    - Present methods to solve problems
- Quite widely used, and growing
- **Active** community
  - Approachable developers
  - Helpful online forum





Classes of problems solved using deal.II

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

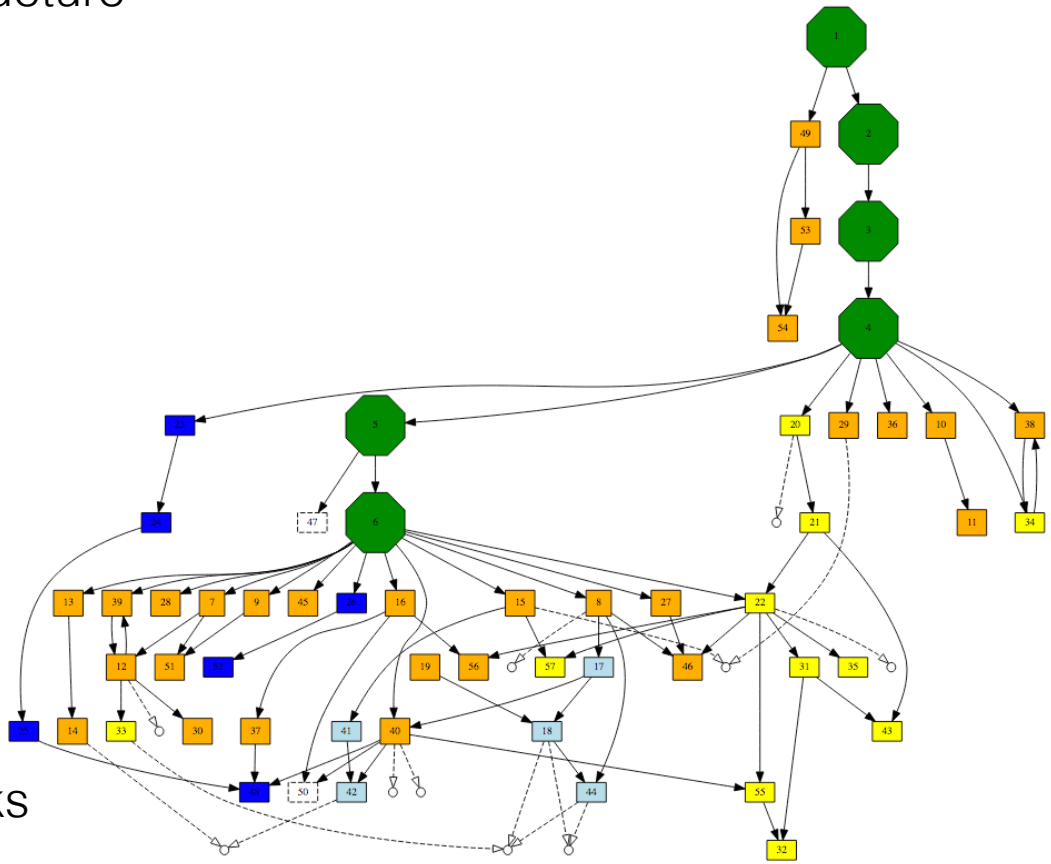


What deal.II is not

- A black box
  - You can't throw a problem at it, and hope it will solve it (like, say, comsol)
  - Won't do anything more than you ask it to
- deal.II knows little about
  - Numerical methods
  - Problem-specific details, i.e.
    - Preconditioners
    - Constitutive equations

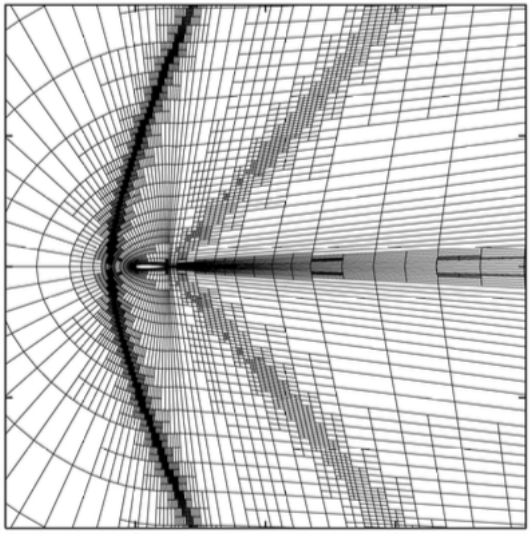
How deal.II will help you

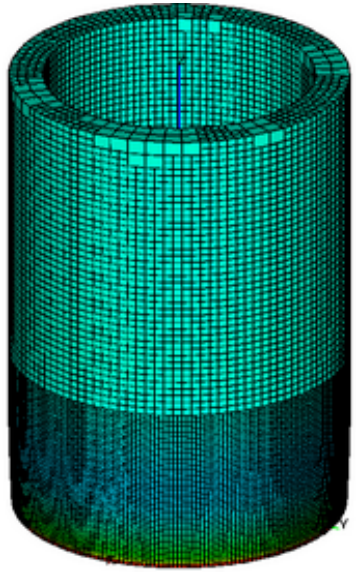
- Unified and well thought out data structure
  - Problem implementation
- Many tutorials
  - Baseline from which to build on
  - Demonstrate how to use features
- Comprehensive debugging support
  - Error messages everywhere!
- Some built in numerical tools
- Integration with advanced frameworks
  -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

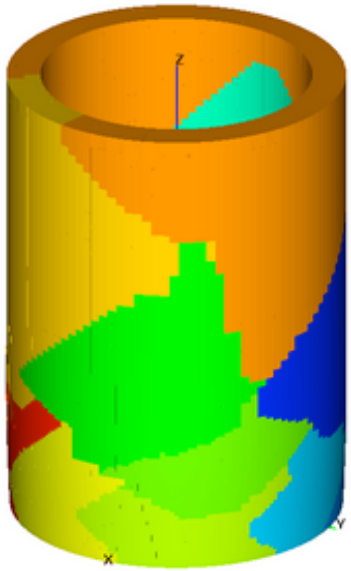


Fundamental capabilities and frameworks

- Mesh adaptivity
- Dense and sparse linear algebra
  - Built in tensor, dense matrix/vector classes
  - BLAS and LAPACK integration; GSL
  - Built in linear solvers and preconditioners
  - Eigenvalue solvers
- Parallelization
  - MPI
    - Linear algebra libraries (PETSc, Trilinos)
    - Distributed meshes ⟶ Billion DoFs
  - Threading (Intel TBB)
  - Vectorized numbers (AVX extensions)
- Pre/post-processing

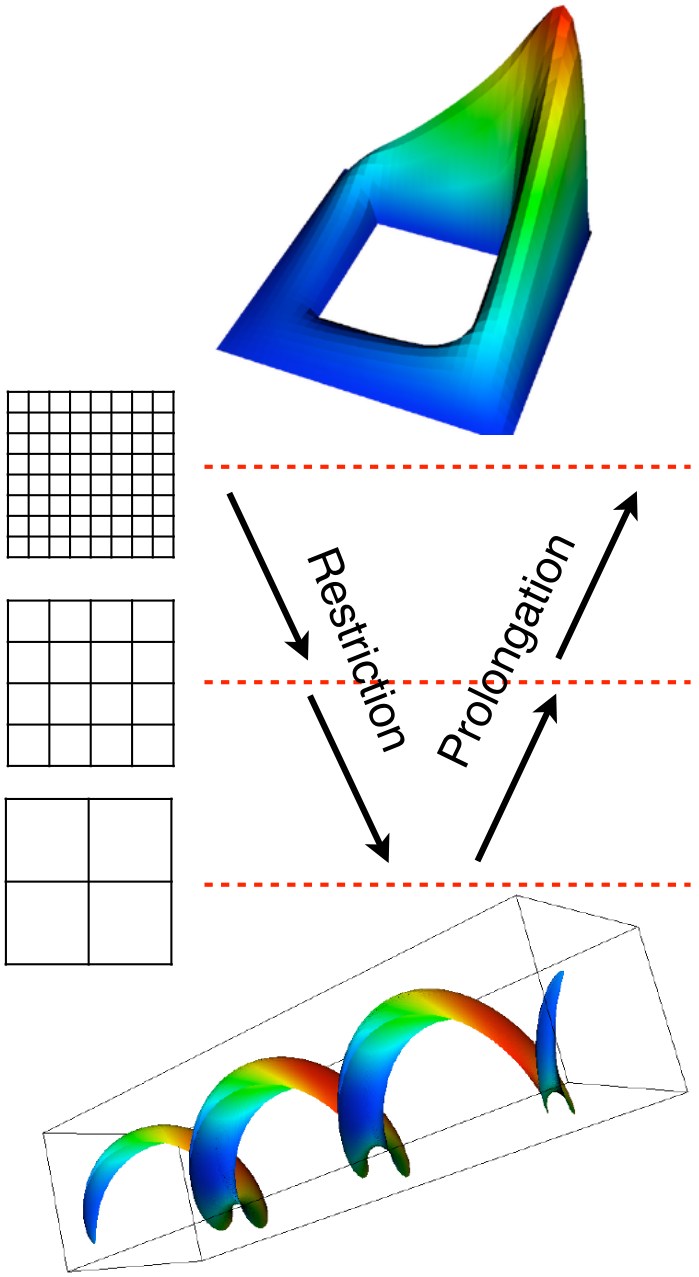






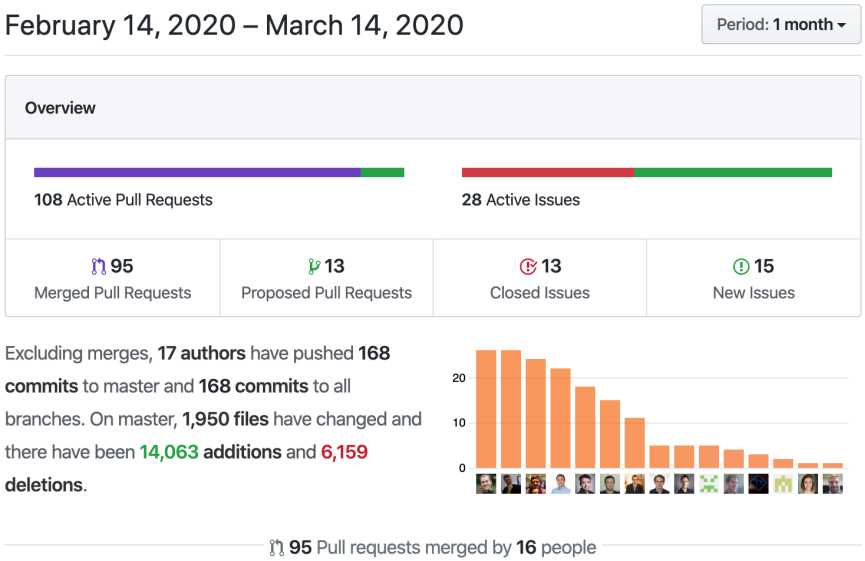
Advanced capabilities and frameworks

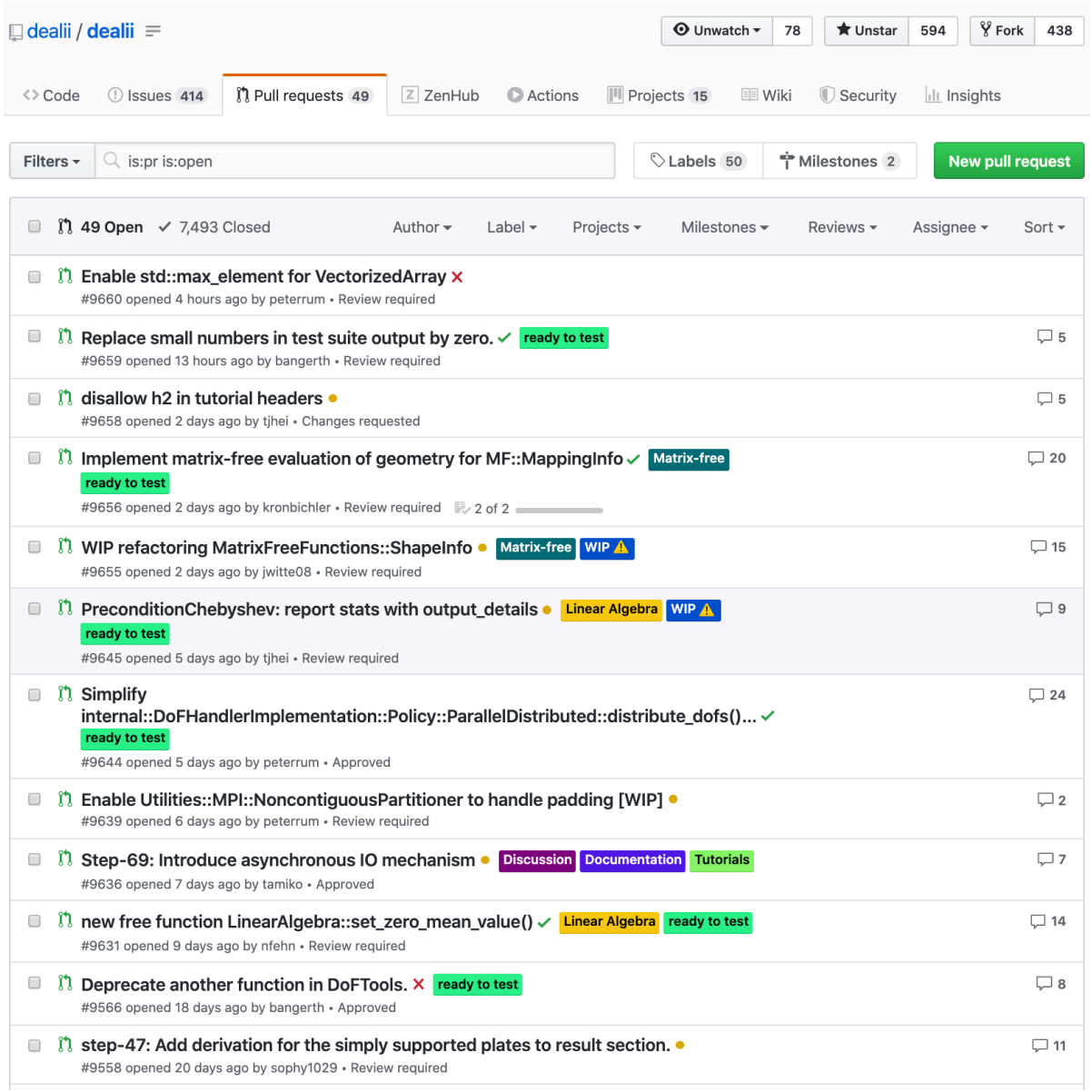
- hp-finite element support
- Meshworker
  - Assembly assistance
  - Functions to perform assembly for specific problem classes
- Geometric multi-grid
  - Using coarse grid as preconditioner to solution for finer grid
- Matrix-free
  - No explicit storing of matrix elements
  - Exchange memory transfer for computations
- 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
- On average: 3/4 pull requests/day





Configuration and Setup

- Course repository: **https://github.com/sissa/p2.3_seed**
- Docker image: **https://hub.docker.com/r/dealii/dealii**
- .dmg file (for mac users): **https://github.com/dealii/dealii/releases**
- Manual installation with spack: **spack install dealii**
- Other means of installation: **https://dealii.org/download.html**
- Notice: build from scratch, with all dependencies ~ 6hours…
- We'll use an IDE (**Qt Creator** for me, but you can choose the one you prefer)
- We'll use a VTK viewer (**Paraview** for me, but **Visit** is also possible)