NSF CSSI PI Meeting, Alexandria, VA, July 25-26, 2022

CSSI Frameworks: Future Proofing the Finite Element Library deal.II – Development and Community Building

Award #:
OAC-1835673
OAC-2015848

PI: Wolfgang Bangerth, Co-Pi: Timo Heister, Representative: Marc Fehling

Institutions: Colorado State University, Clemson University

Summary

Finite element computations in science and engineering are becoming more complex:

- More coupled physical processes
- More advanced numerical methods (adaptive meshes, multigrid preconditioners, coupled nonlinear solvers)
- More parallelism, from laptops to supercomputers

There is a clear need for libraries supporting such applications at all levels of complexity and parallelism.

deal.II is a widely used, open source library providing finite element and linear algebra support for individual research applications. This project aims at broadening support for parallel computations in these libraries:

- Expand the user and developer community, and support them in their goals
- Extend support for massively parallel and GPU computations
- Expand the range of documentation, tutorials, video lectures, summer schools, and other sources of training
- Implement foundational functionality too complex to obtain from the user-developer community
- Better support a wide range of platforms.

Background

There is large demand for generic libraries supporting modern finite element applications. This demand stems from two sources:

- Applications scientists wanting to solve complex problems for which no software exists yet
- Methods developers wanting to try new discretizations, solver schemes, parallelization approaches.

Both communities require a large, well documented, extensively tested, and portable foundation for their work that abstracts the basis building blocks for finite element codes.

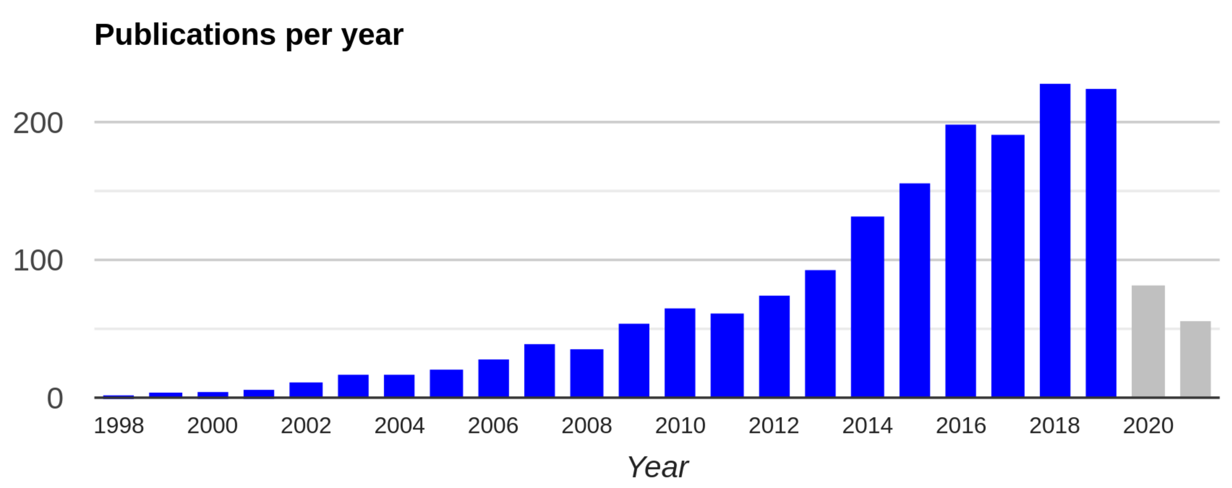
deal.II provides this basis to hundreds or thousands of users and has supported ~1,800 research projects. Providing such a foundation also supports more user groups in using large-scale computing – democratizing supercomputing and lowering the threshold for entry for new groups.

Building a user community

Software can only be sustainable in the long run if it has both a user and developer community.

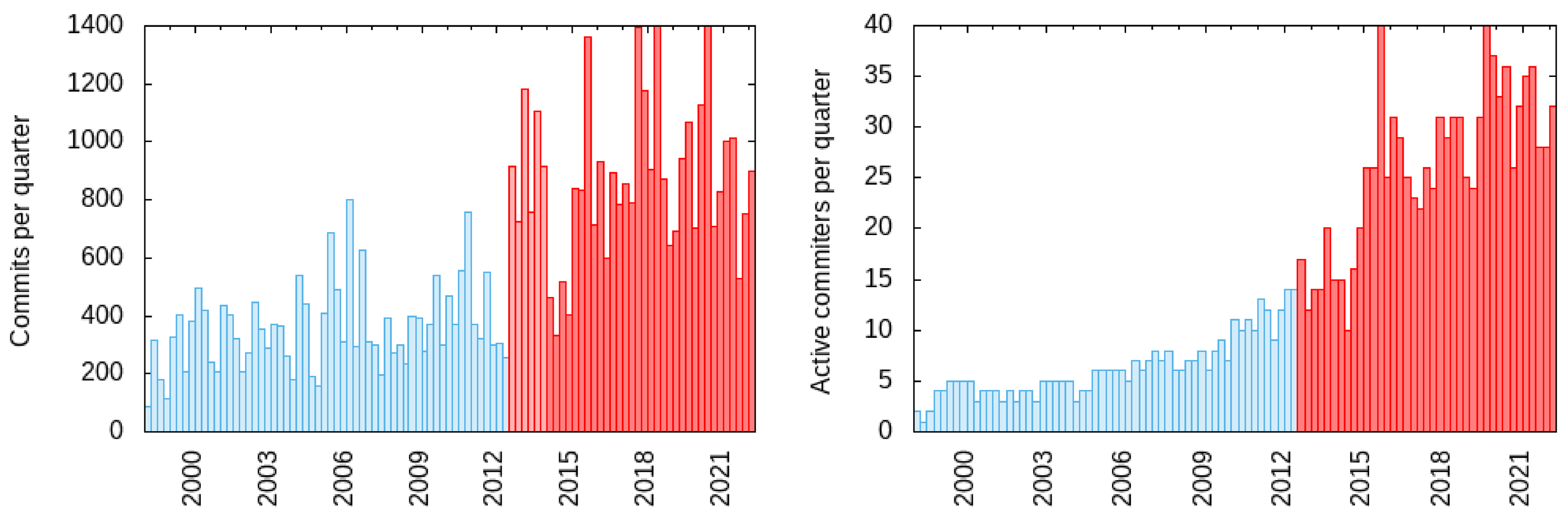
An important part of the deal.II project (and of this NSF grant) is to support the user community by building on the already extensive documentation, and by running training events. We also grow great attention to mentoring users into developers.

All of this is working: there are now ~1,400 members on the mailing list, and we know of ~225 publications per year generated using deal.II:



Known publications using deal.II. Total: **1793**. (Gray bars: Incomplete data.)

The following two figures show the number of commits and committers per quarter – demonstrating that many more developers commit many more commits than just a few years ago.



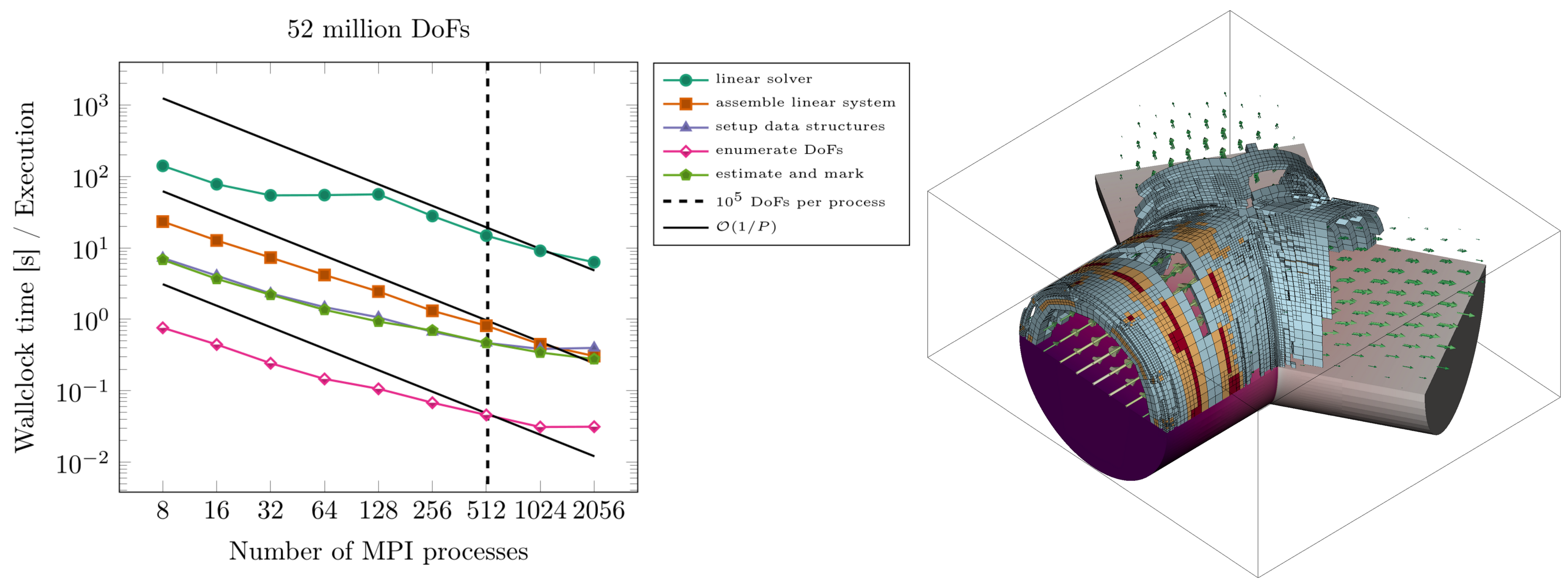
In support of our user community, we also ran a Summer School and User/Developer Meeting in 2019, with 60+ participants. 3 of the 5 Summer School lecturers were women, and the majority of participants were women and underrepresented minorities. We continued to host User/Developer Meetings annually in 2020 and 2021.

https://www.dealii.org

Technical achievements 2018-2022

Massively parallel *hp*-adaptivity: We have extended the implementation of *hp*-adaptivity to the parallel case, making many more complex models run on large clusters.

Left figure: Example for strong scaling of an *hp*-adaptive problem involving the Laplace equation. Right figure: Distribution of polynomial degrees in an *hp*-adaptive scheme for a Stokes flow problem.



Parallel geometric multigrid: We have implemented massively parallel, geometric multigrid methods for the solution of linear systems. The algorithm works for arbitrary elements, matrix-based and matrix-free, and is based on MPI or hybrid multi-threading. We found that novel global coarsening methods are typically 2-3x faster for AMR examples compared to traditional local smoothing.

Scalability: We introduced MPI Large-Count to support large I/O for graphical output and checkpointing with $>2^{31}$ counts. `MPI_*_c` methods from MPI 4 standard had to be reimplemented in MPI 3.0 routines for compatibility. I/O tuning gives 10x faster parallel output.

Infrastructure: We run ~13,000 tests several times per day, and have extended the number of platforms and tests that are run before each pull request can be committed. We cover Linux, Mac OS X, and Windows systems with gcc, clang, Intel, and MSVC compilers.

Future plans

Continue to build educational materials: deal.II has outstanding documentation, tutorials, and video lectures. We will continue to build these in order to broaden the number of starting points for new and existing users.

Continue to improve scalability: Running simulations, postprocessing, and visualizing uncovers new bottlenecks with every jump in scale. We will continue to find and fix these to make deal.II ready for even bigger machines.