How do we make large opensource projects sustainable?
(lessons learned from 25 years of deal.II)

Luca Heltai <luca.heltai@sissa.it>

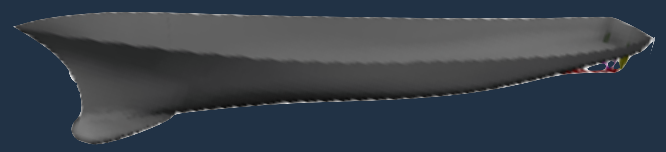
SISSA — International School for Advanced Studies (www.sissa.it)

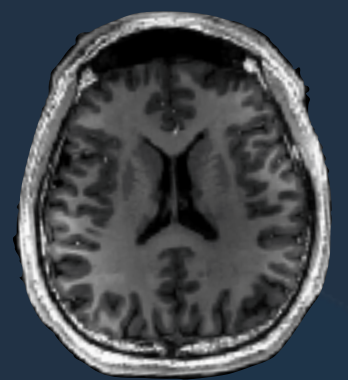
Associate professor of Numerical Analysis — SISSA mathLab (mathlab.sissa.it)
Founder and former Director of the Master in High Performance Computing (www.mhpc.it)
One of the principal developers of deal.II (www.dealii.org)

SISSA

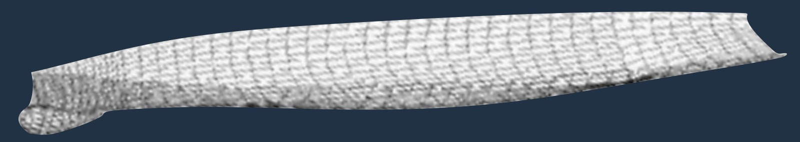
Our typical workflow

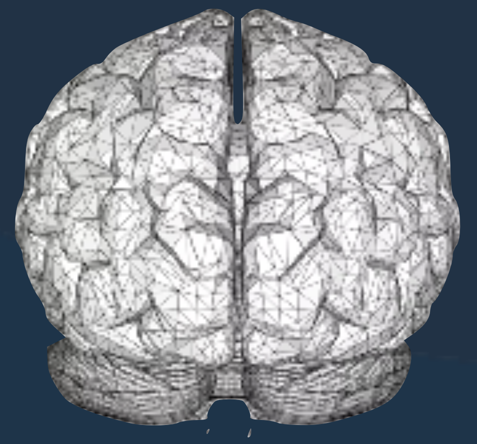
CAD
Medical Image
Geometry definition
...





Computational
Mesh





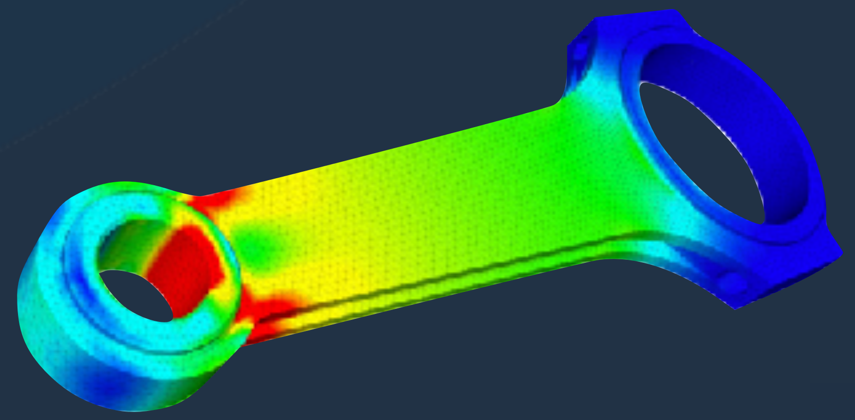
Mathematical Model

$$-\nabla \cdot (\kappa \nabla u) = f \qquad \text{in } \Omega$$
$$u = g \qquad \text{on } \Gamma \equiv \partial\Omega$$

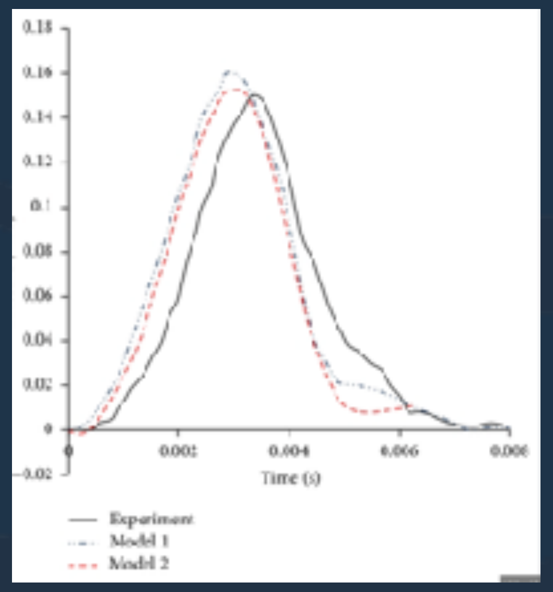
Numerical
Discretisation

$$Au = F$$

Simulation



Postprocessing



Improve
repeat

Our typical workflow

Industry is a user of the entire framework

- Ready-made solutions
- Integration with existing tools, frameworks, inputs, etc.
- Robustness, reliability, and resilience to errors

Academia is a developer of parts of the framework

- Willing to experiment
- Need to show new methods and algorithms
- Publish or perish

SISSA

The ideal (industry ready) CSE software (FEM based)

Realistic, complex geometries in 2d and 3d
Multiphysics
Efficient linear and non-linear solvers

Adaptive meshes
Higher order elements

Efficient multigrid solvers (algebraic and/or geometric)
Scalability to thousands of processors
Efficient use of current hardware

How can we make all of this happen in a single code?

From scratch it would be roughly **~100K** lines of code, if not more.

Research Software in Academia

The bitter reality!

Written by graduate students
without a good overview of existing software
with little software experience
with little incentive to write high quality code

Maintained by postdocs
with little time
who consider the software primarily as a tool to publish

Advised by faculty
with no time
oftentimes also with little software experience

Typical life span of most research projects: 5/7 years

SISSA

Why are we so bad? — I

Greg Wilson, D. A. Aruliah, C. Titus Brown, Neil P. Chue Hong, Matt Davis, Richard T. Guy, Steven H. D. Haddock, Kathryn D. Huff, Ian M. Mitchell, Mark D. Plumbley, Ben Waugh, Ethan P. White, Paul Wilson, 2014.

Best Practices for Scientific Computing.
PLoS Biol 12, e1001745:

"Scientists spend an increasing amount of time building and using software. However, **most scientists are never taught how to do this efficiently**.

As a result, many are **unaware** of tools and practices that would allow them to write **more reliable and maintainable** code with less effort."

SISSA

Why are we so bad? — II

Tobias Kuipers, 2016,
Why you need to know about code maintainability
O'Reilly and Software Improvement Group

- On average, a good developer manages 10K lines per year
- Software that does not evolve, stops being useful
(if you don't continue to update and modify an existing software system or component, it will eventually stop working)
- Rughly 15% of the code needs to be updated every year to remain functional
(Updating the operating system/changing compiler, updating dependency libraries, etc. require changes in the code base!)

Every 66K lines of code require a full time developer!

SISSA

We must change perspective

1. (re)use existing software (libraries)!

2. keep the amount of code you must maintain at a minimum

3. write code with *sustainability* in mind

- Use the work of others (standing on giants’ shoulders)
- **Do not** reinvent the wheel (don’t write your own non-linear solvers!)
- Make sure that your work can be useful (and therefore maintained!) also by others

SISSA

We must change perspective

"User" perspective

- frameworks, rather than libraries
    - fenics
    - firedrake
    - ngsolve
- graphical user interfaces
    - low entry barrier for new users
    - high entry barrier from users to developers

"Developer" perspective

- libraries, rather than frameworks
    - BLAS/LAPACK/MPI
    - PETSc/Trilinos/
    - deal.II/libmesh/dune
- command line interfaces
    - high entry barrier for new users
    - low entry barrier from users to developers

dealii.org

SISSA

Bangerth, W., Heister, T., 2013. *What makes computational open source software libraries successful?* Comput. Sci. Disc. 6, 015010

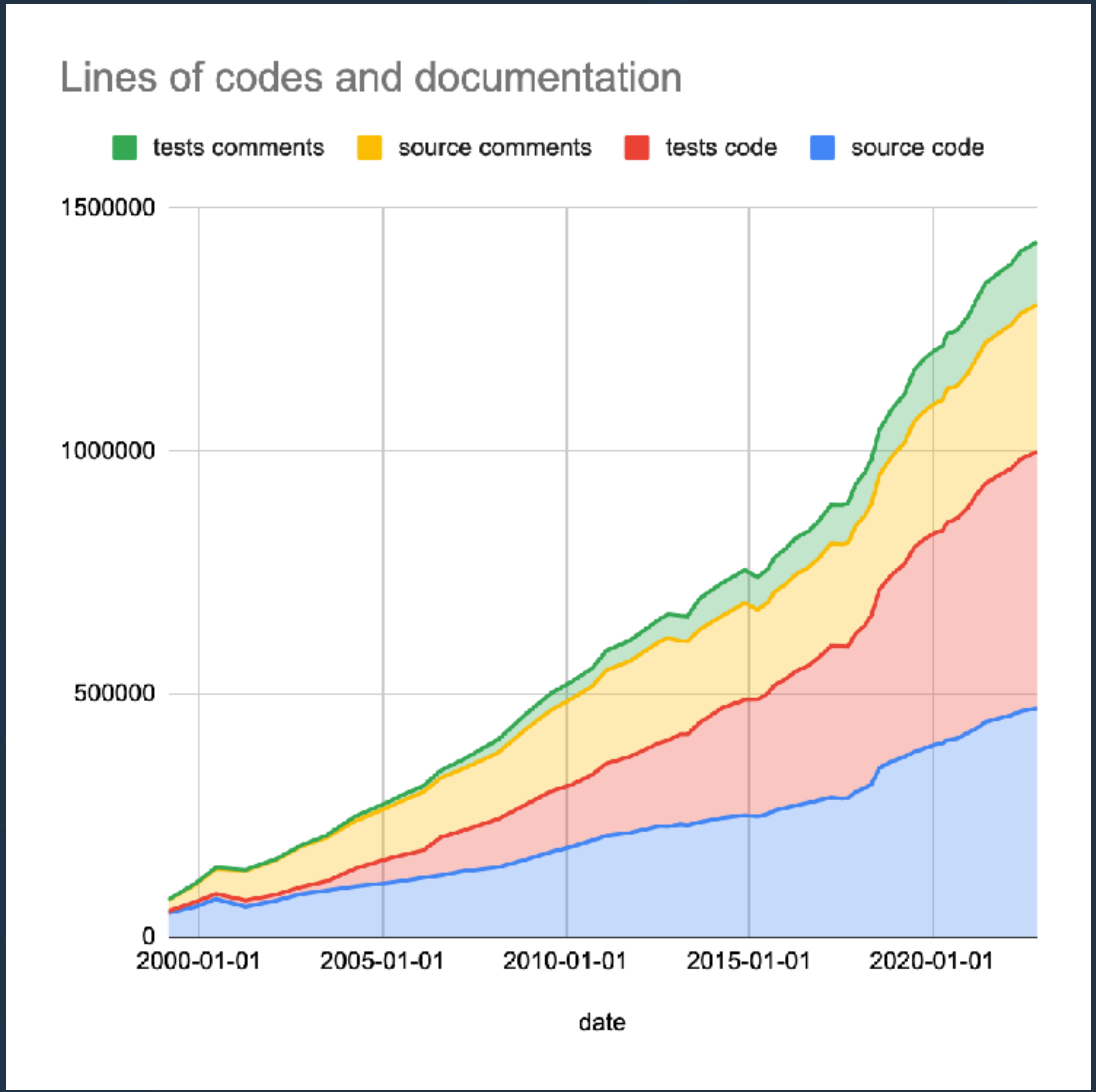
“A common **belief** is that writing successful software packages is largely the result of simply ‘**being a good programmer**’ when in fact there are **many other factors involved**”

- community building
- improving the quality and utility of the code
- writing good documentation
- project management
- licenses
- **attracting new developers**…

SISSA

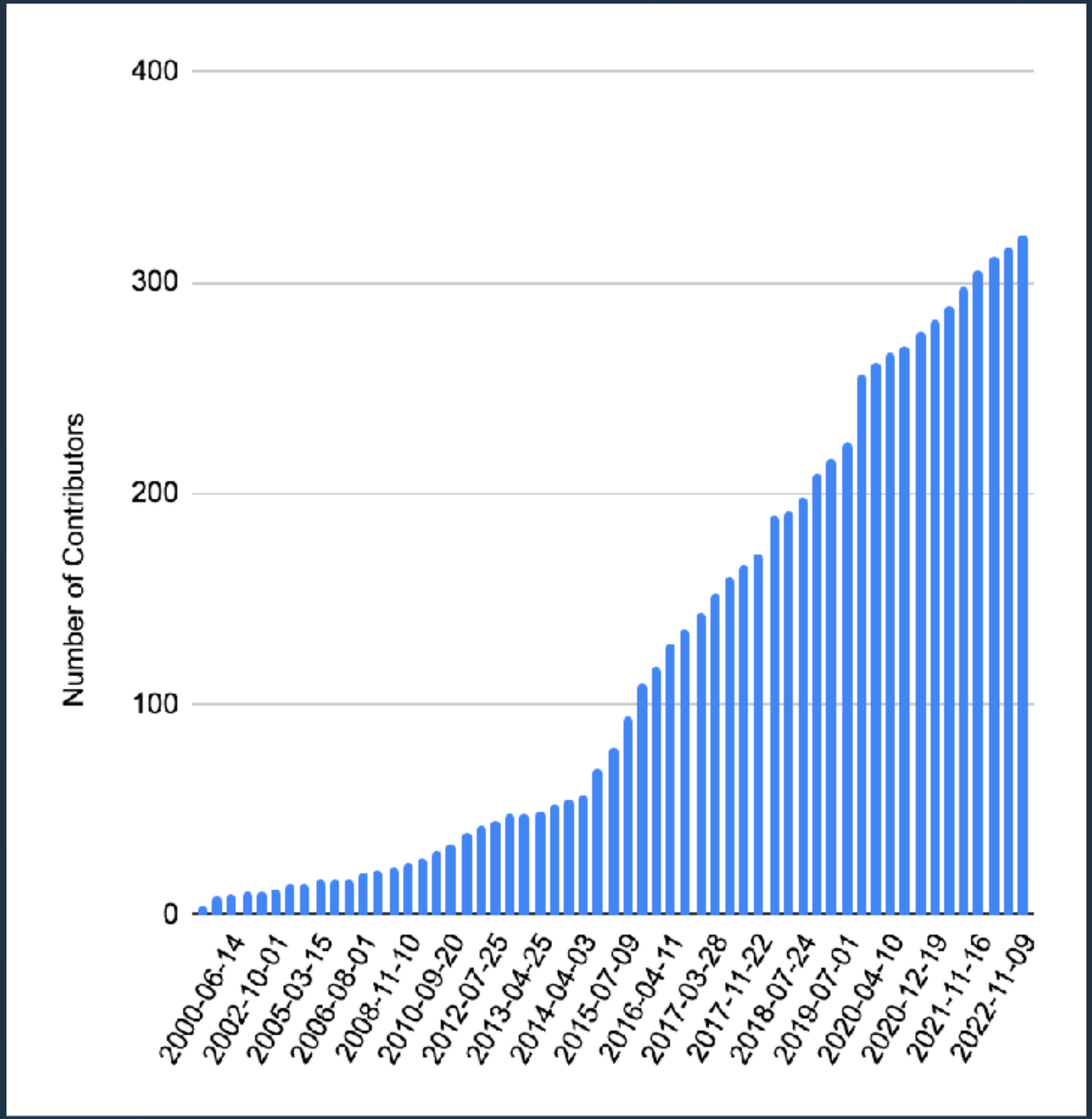
Some numbers about deal.II

- It is truly a **Vibrant** project:
  - +3k new lines of code per month since the beginning (today: 1M lines of pure code)
  - 1000+ pages of documentation (today: 500k lines of comments)
  - An average of **5 pull requests per day, every day**
  - **New schools, courses, training, and video lectures (popping up every year)**
  - More than 1000 people on the mailing list
  - 300+ contributors
  - 100+ downloads per month
  - 200+ publications per year since 2016 (a total of more than 2.2K publications)



Some numbers about deal.II

- It is truly a **Vibrant** project:
  - +3k new lines of code per month since the beginning (today: 1M lines of pure code)
  - 1000+ pages of documentation (today: 500k lines of comments)
  - An average of **5 pull requests per day, every day**
  - **New schools, courses, training, and video lectures (popping up every year)**
  - More than 1000 people on the mailing list
  - 300+ contributors
  - 100+ downloads per month
  - 200+ publications per year since 2016 (a total of more than 2.2K publications)



SISSA

Today: 13 Principal developers

- **Daniel Arndt**
  Oak Ridge National Laboratory, USA
- **Wolfgang Bangerth**
  Colorado State University, CO, USA
- **Bruno Blais**
  Polytechnique Montréal, Canada
- **Marc Fehling**
  Colorado State University, CO, USA
- **Rene Gassmoeller**
  University of Florida, FL, USA
- **Timo Heister**
  Clemson University, SC, USA
- **Luca Heltai**
  SISSA, Trieste, Italy
- **Martin Kronbichler**
  University of Augsburg, Germany
- **Matthias Maier**
  Texas A&M University,
  College Station, TX, USA
- **Peter Munch**
  University of Augsburg, Germany
- **Jean-Paul Pelteret**
- **Bruno Turcksin**
  Oak Ridge National Laboratory, USA
- **David Wells**
  University of North Carolina,
  Chapel Hill, NC, USA













Developers Emeriti

- **Denis Davydov**
- **Ralf Hartmann**
  DLR, Germany
- **Guido Kanschat**
  Heidelberg University, Germany
- **Toby D. Young**
  Polish Academy of Sciences, Poland

SISSA

Our experience:
a brief history of deal.II

- Bonn, 1992: Franz-Theo Suttmeier and Guido Kanschat start playing with FE: **DEAL** is born (Differential Equation Analysis Library) — 1993—2000
- Heidelberg, 1997: Guido Kanschat and Franz-Theo Suttmeier (postdocs at the time) teach **DEAL** to Wolfgang Bangerth and Ralf Hartman (diploma students)
- Heidelberg,1997: Wolfang starts a new project (**deal.II**) for his diploma thesis (some small things taken from DiffPack)
- Heidelberg, 1998: Guido contributes with expertise, and starts contributing code to **deal.II**
- Heidelberg,1999: Ralf used **DEAL** for his diploma, but wanted something different for his PhD (he starts working on **deal.II**)

The **deal.II** library is born — it goes public in 2000 (**deal.II v3.0.0** 124k locs)

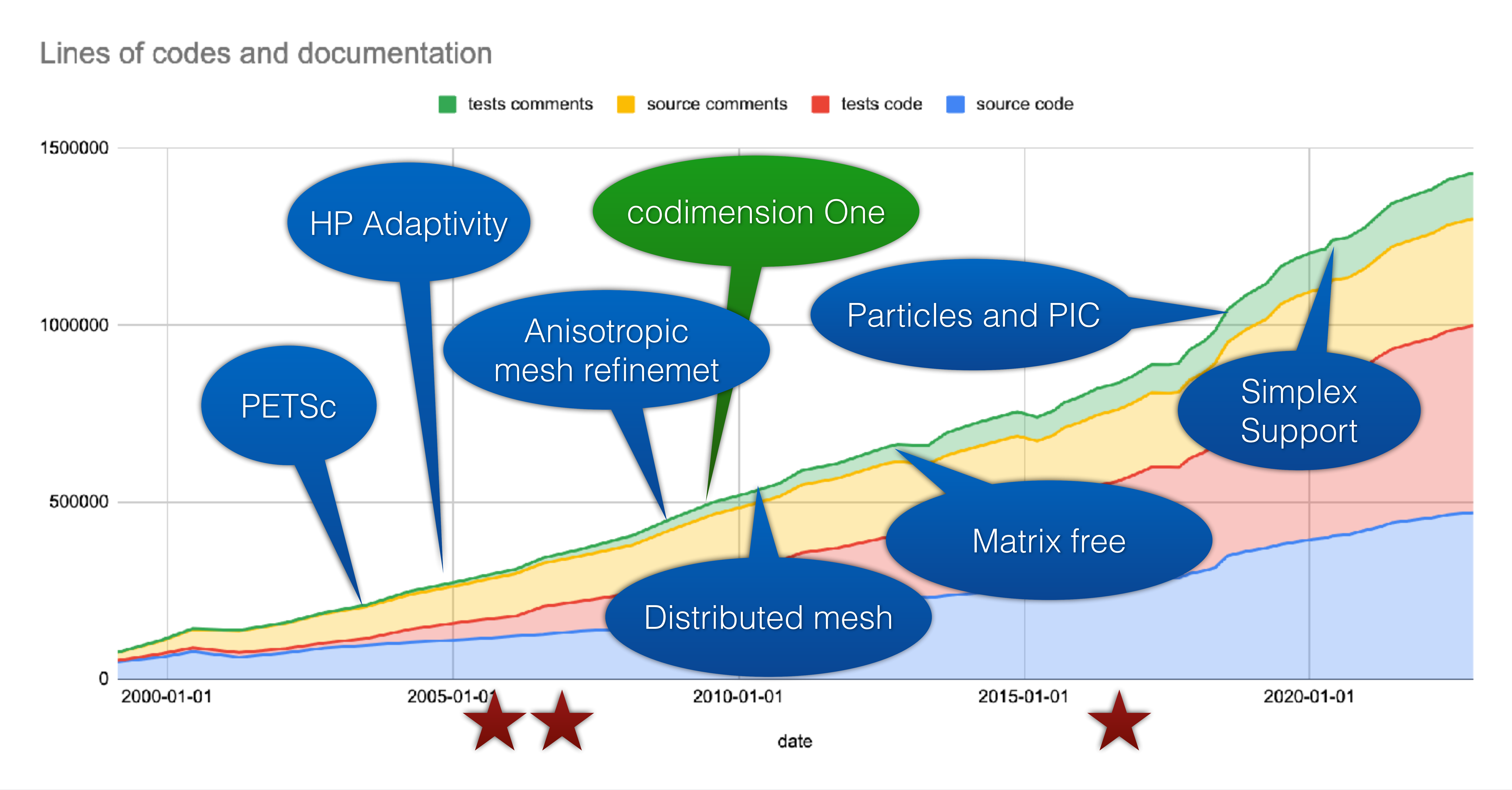
SISSA

A brief history of deal.II

- 1997: Start of project
- 2000: First public version (deal.II 3.0, 124k locs)
- 2002: www.dealii.org comes online (deal.II 3.4, 14 tutorials, 173k locs)
- 2004: First support for parallel computations (METIS+PETSc, 17 tutorials)
- 2005: First support for HP adaptivity
- 2006: deal.II becomes part of SPEC CPU 2006
- 2007: WB, GK, and RH win the J. H. Wilkinson Prize for Numerical Software
- 2008: Anisotropic mesh refinement
- 2009: Support for co-dimension one manifolds (deal.II 6.2, 34 examples, 400K locs)
- 2010: Distributed mesh support
- 2012: Matrix free framework
- 2017: deal.II becomes part of SPEC CPU 2017
- 2018: Particles and PIC support
- 2021: Simplex support (deal.II 9.3, **79** tutorials, **1.3M** locs)
- 2023: deal.II 9.5, **86** tutorials, **1.8M** locs

SISSA

A graphical history of deal.II



Our cumulative knowledge

remember ~**10K lines per developer per year?**

~100 years of one developer only working on code
~50 years of one developer only working on documentation

If we consider all linked open-source external libraries as well, we quickly get to **thousands of years!**

No small business, research group, or single developer can ever hope to compete!

SISSA

Why did I choose deal.II?

- I was a PhD student in Pavia (with prof. Daniele Boffi), starting in **2003**
- I liked programming, and I was given total freedom to choose my working tools
- I was working on fluid-structure interaction problems with non-matching methods

No opensource tools was readily available for what I wanted to do.
I needed to build something on my own.

SISSA

Why did I choose deal.II?

- I stumbled upon the deal.II web page and read through the first three tutorials
- I tried compiling the library, and it compiled without issues (unlike many other libraries I had tried before)
- I worked my way through the first tutorials and I saw, used, and touched first hand, *for the first time*, the things that I had been told in classes

I never went back!

SISSA

My start with deal.II

- First trivial commit: a small change in a header file — 2004
- First meaningful contribution: interface with Function Parser library — 2005
- First major contribution: co-dimension one — 2009

SISSA

We wanted BEM

- My first Industrial project @ SISSA: Rinave — 2007—2009
    - Goal: replace an old “Panel Method” code, order zero fortran 4 + fortran 77
- I wanted to keep using deal.II for the project!

We wanted BEM

- Problem:
  - deal.II was built for dim-dimensional grids embedded in dim-dimensional space (dim=1,2,3)
  - it was not possible to work on a surface mesh, embedded in three dimensions
- Our solution:
  - we modified almost every file of deal.II to add an additional template parameter to work on surface meshes
    - we made all possible mistakes that could be made!
    - it took Wolfgang several weeks of code reviewing — and headaches! — before the changes were finally merged

SISSA

We learned a few things

As contributors to an open-source project

- Coordinating a large project is not easy — **technically**
    - pull requests
      *(GitHub hosting and first pull request on deal.II: **2014**)*
    - strict peer review from principal developers
      (not ourselves! we *cannot* merge our own patches)
    - strict feature testing
      (a pull request is not merged if some 12K tests don't pass)

SISSA

We learned a few things

As contributors to an open-source project

- Coordinating a large project is not easy — **socially**
  - how do we convince people to invest in contributing? (it takes an awful lot of time!)
  - how do we help people? (mailing list, video tutorials, webinars?)
  - do we accept all contributions? (who is going to maintain it?)

SISSA

We learned a few things

As educators

- The most precious and rare thing to come about is **competence**
  - one way to keep it around is to make its products open-source
  - we are *always* starving for good students/postdocs/researchers
  - **we started forming our own!**

SISSA

We learned a few things

As academic developers

- **Working with industry is not (always) easy**
  - different languages
  - different objectives
  - different timelines

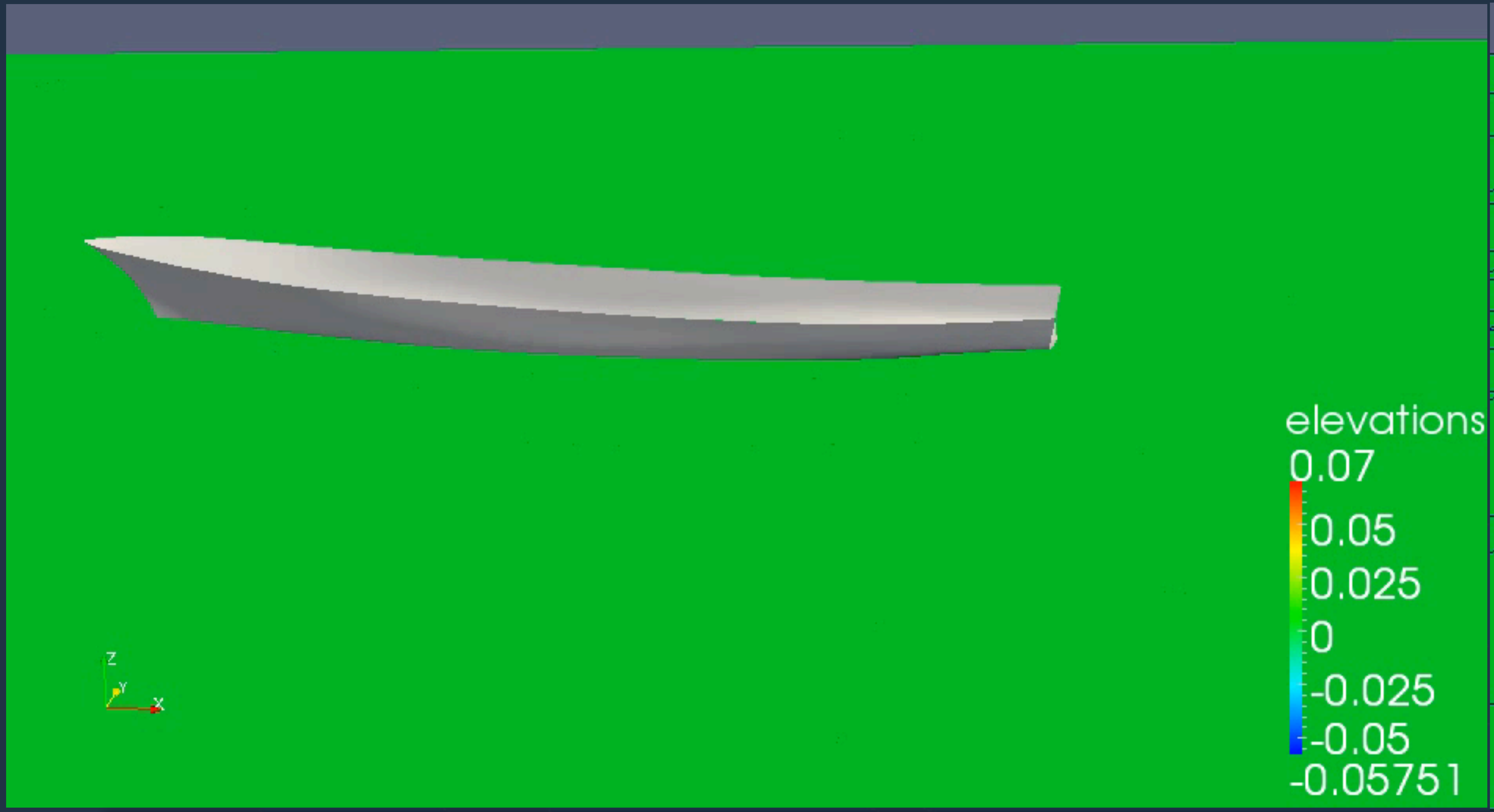
SISSA

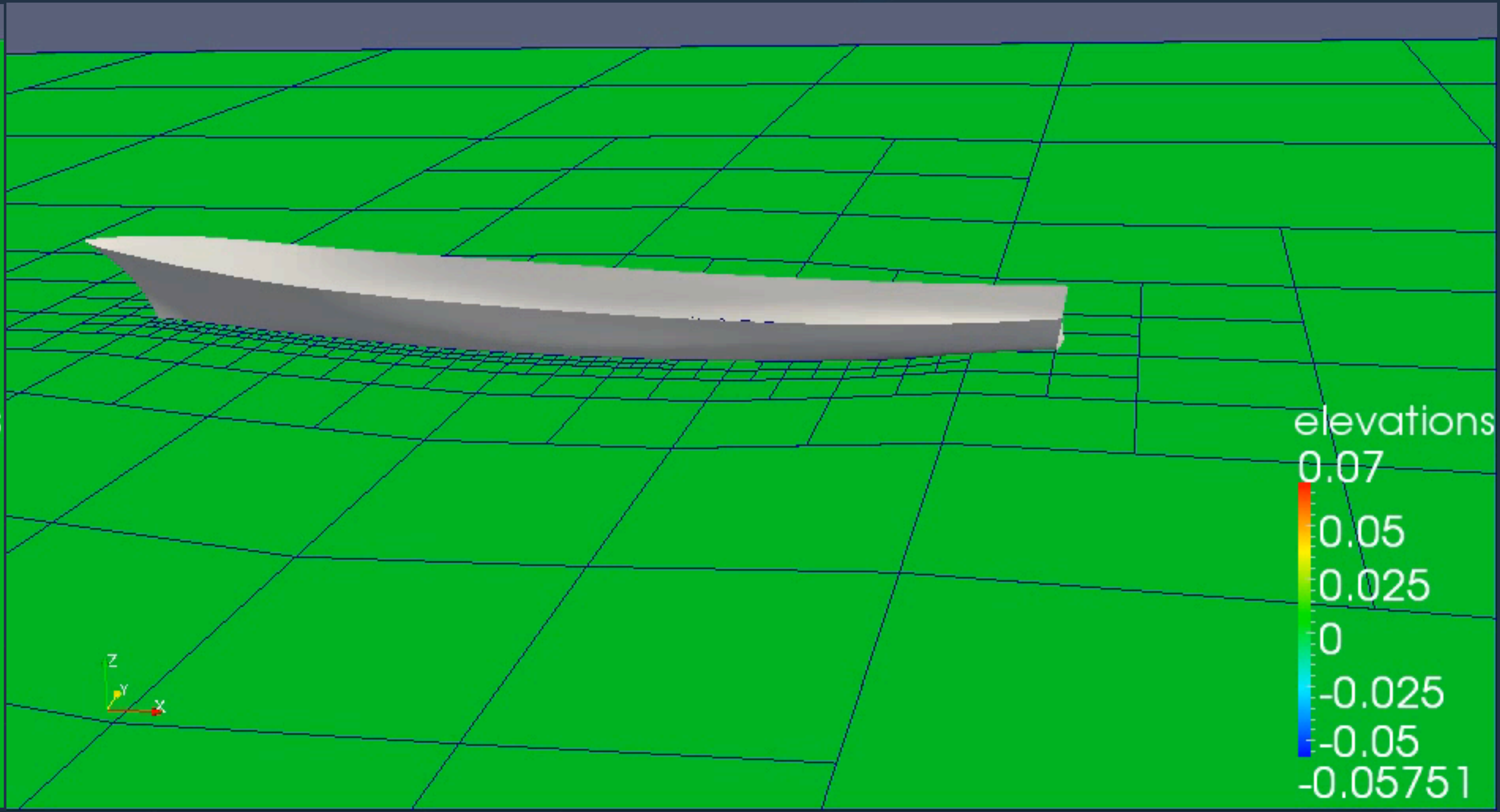
First (personal) success: WaveBEM

- Rinave — not entirely a successful industrial project
- OpenSHIP — first real “win-win” for us — WaveBEM was born
- OpenViewSHIP — consolidate
- … many projects later, our group mathLab@SISSA still collaborates with Fincantieri and CETENA on related topics!
(UBE, Prelica, SOPHYA, UBE2, SrT, etc.)

SISSA

First (personal) success: WaveBEM



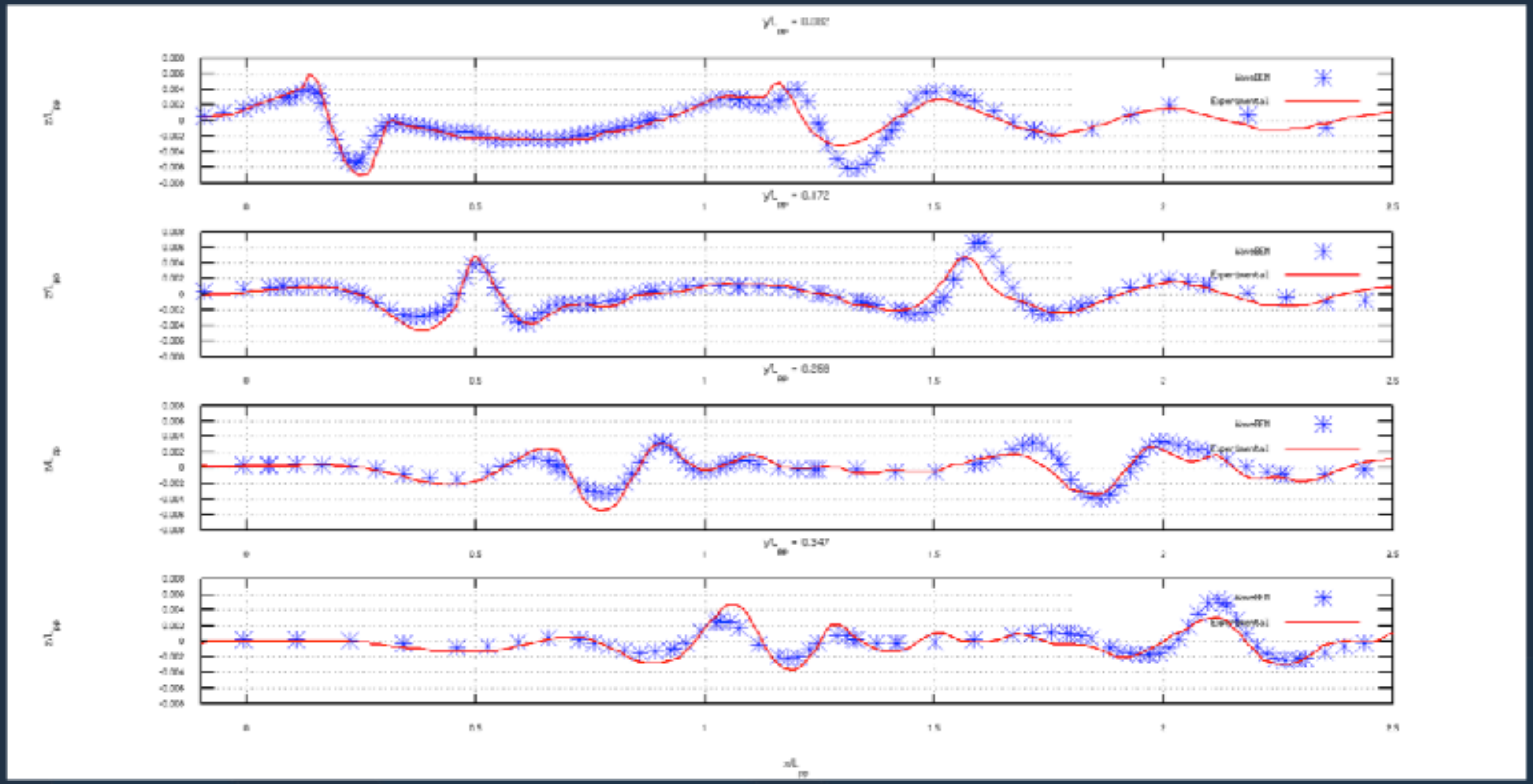


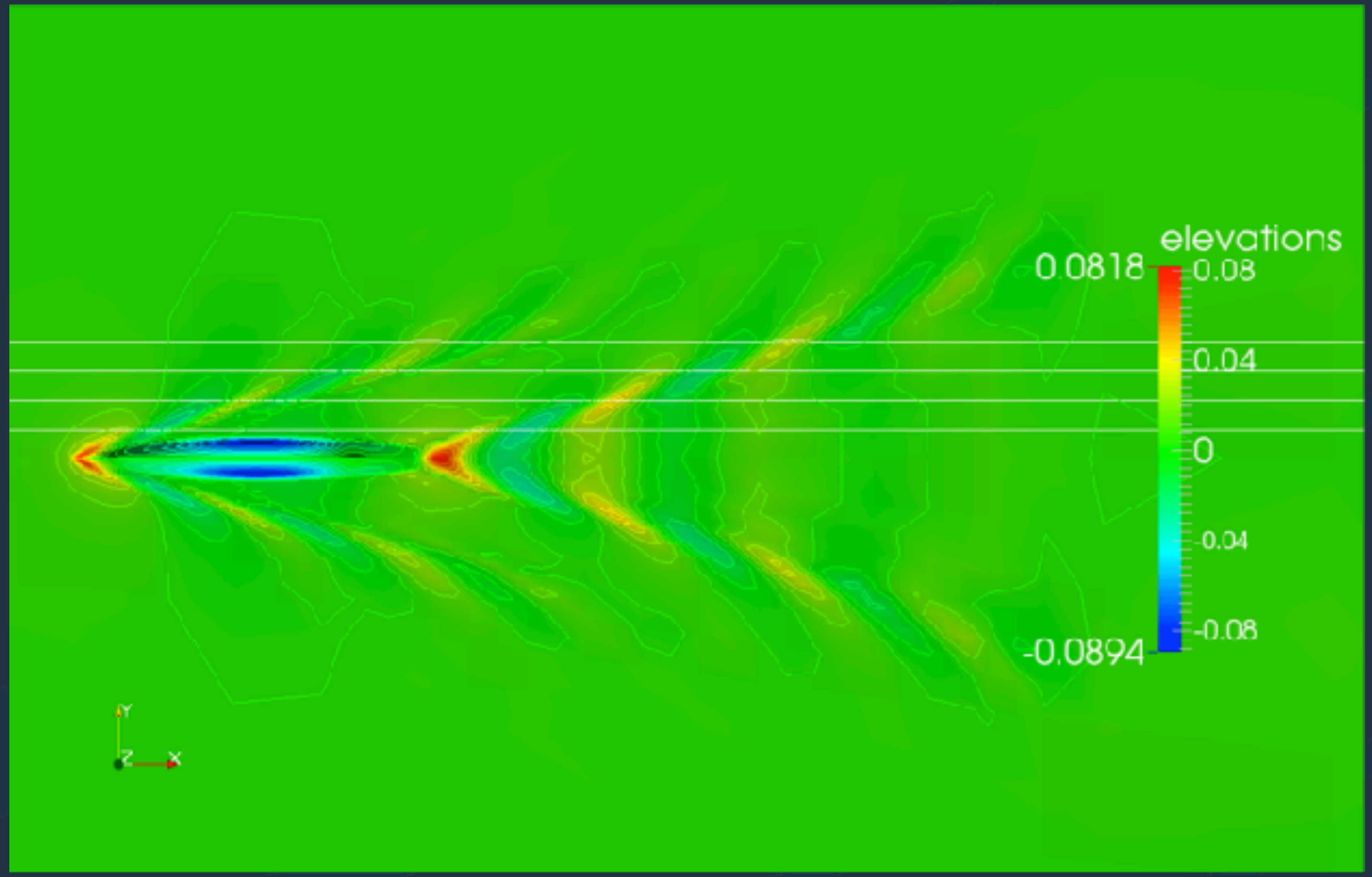
Arbitrary order, locally adaptive, free surface BEM solver for unsteady and nonlinear ship-wave interactions — 2010—2015

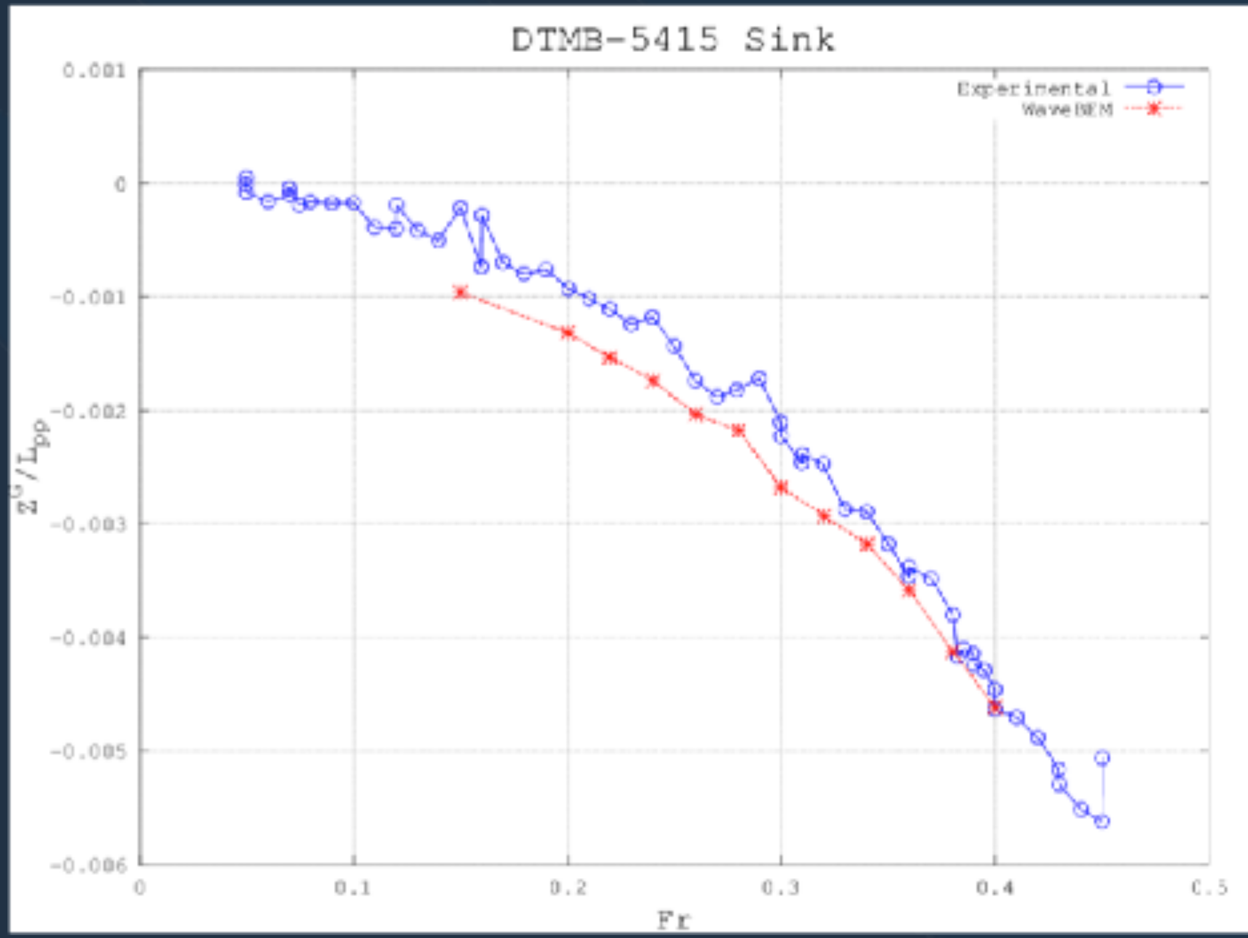
Andrea Mola, **LH**, and Antonio DeSimone, *A stable and adaptive semi-lagrangian potential model for unsteady and nonlinear ship-wave interactions*, Engineering Analysis with Boundary Elements 37 (2013), no. 1, 128 – 143.

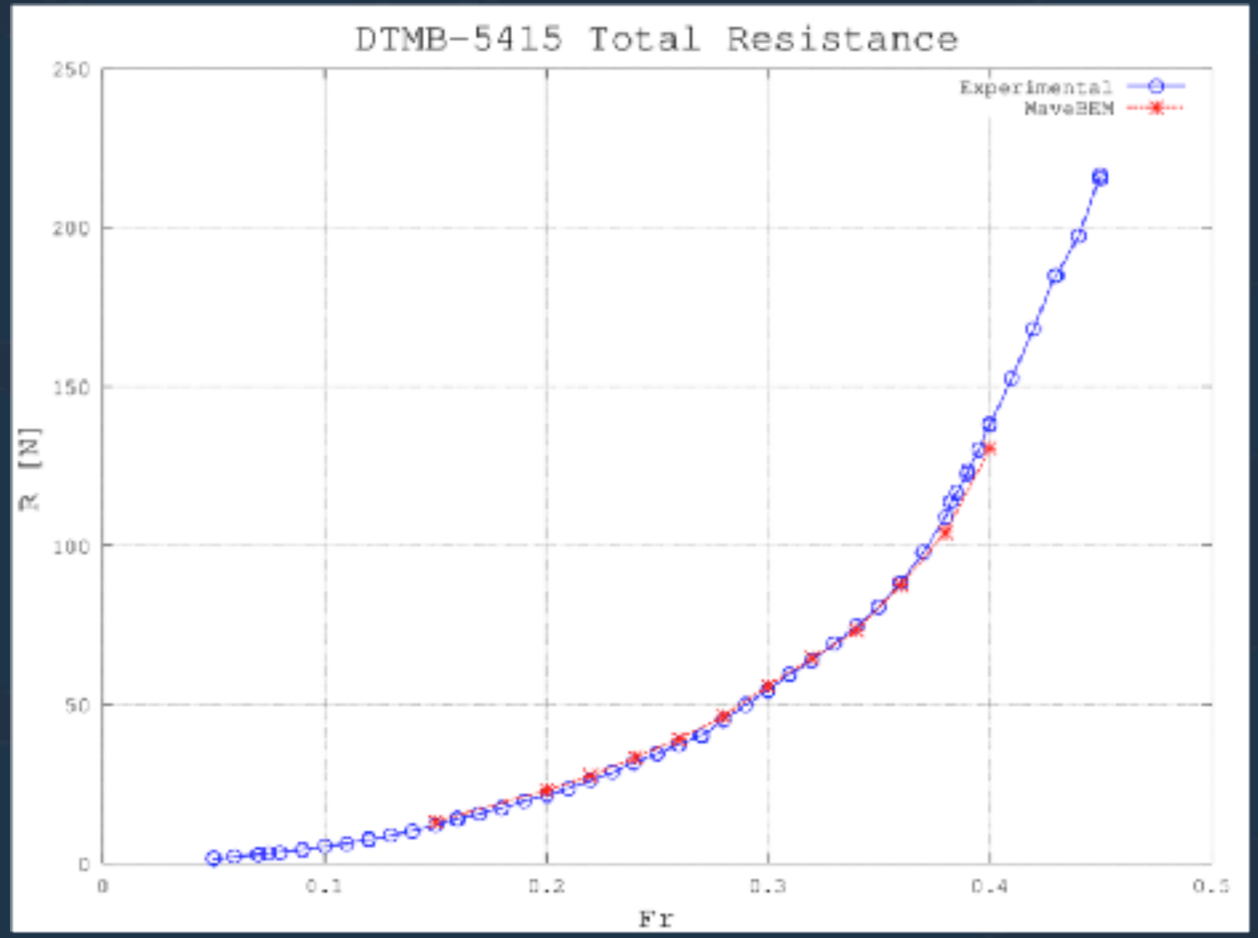
SISSA

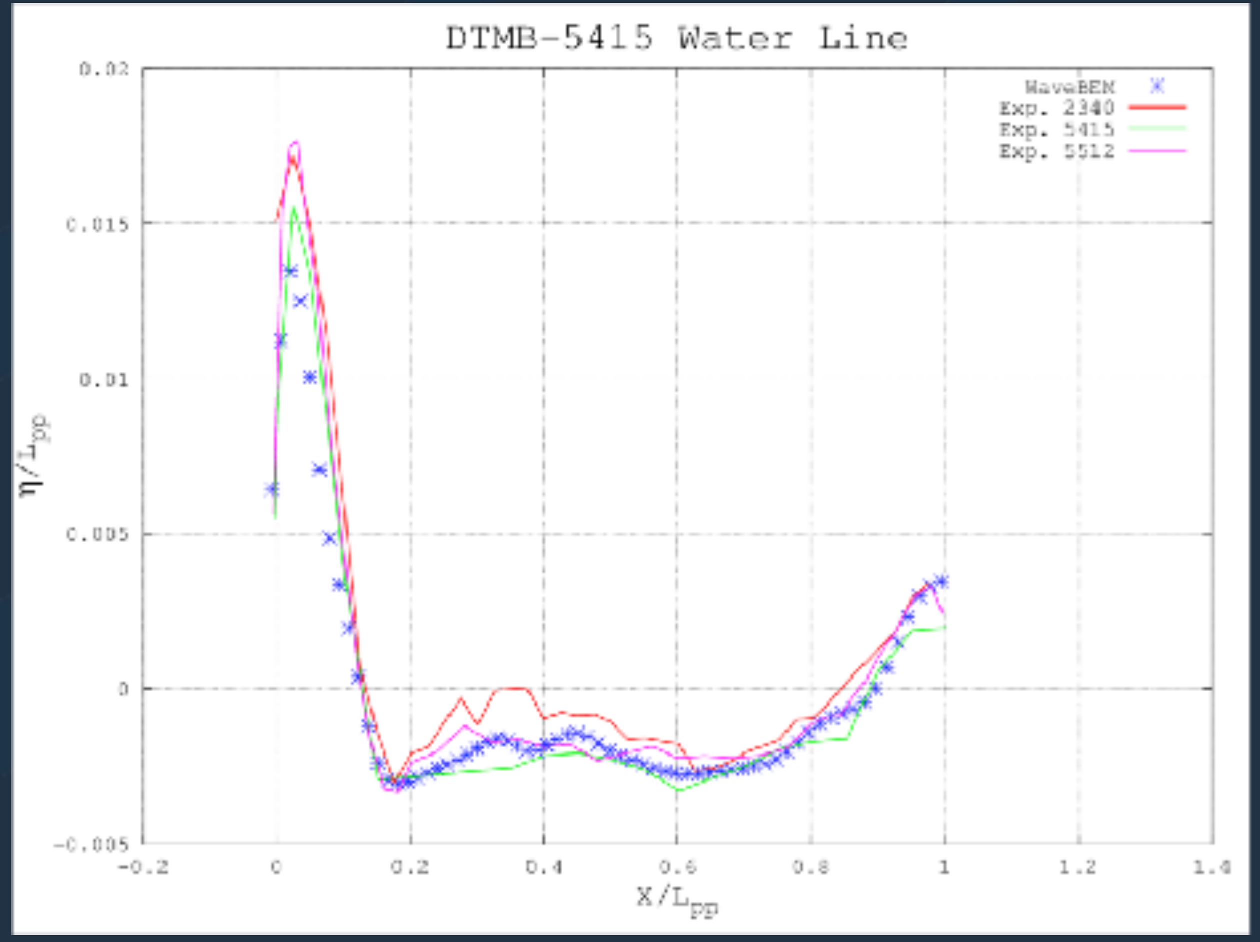
First (personal) success: WaveBEM











The policy of the principal developers: give back to deal.II

Whatever is needed for our projects, we port to deal.II if it can be useful to others

Example: Arbitrary order, locally adaptive, free surface BEM solver for unsteady and nonlinear ship-wave interactions — 2010—2015

Andrea Mola, **LH**, and Antonio DeSimone, *A stable and adaptive semi-lagrangian potential model for unsteady and nonlinear ship-wave interactions*, Engineering Analysis with Boundary Elements 37 (2013), no. 1, 128 – 143.

- 2009: Co-dimension one support (-> enable the definition of geometries and FEM on triangulations embedded in higher dimensions)
- 2014: OpenCASCADE (-> enables CAD geometry description to be embedded in the simulation)
- 2014: Manifold infrastructure (-> formalise how geometry and FEM should interact)
- 2017: Assimp (-> read and write several formats from computer graphics world)
- Interaction with GMSH (2004) and GMSH api (2021) (-> open-source grid generator)
- SUNDIALS (2017) (-> non-linear, ode, and dae support)
- Nanoflann (2017) (-> spacial indices, later replaced with boost::rtree and ArborX)
- boost::rtree (2018) (-> spacial indices)
- CGAL (2022) (-> computational geometry library, intersections, grid generation and improvement, etc)
- ….

SISSA

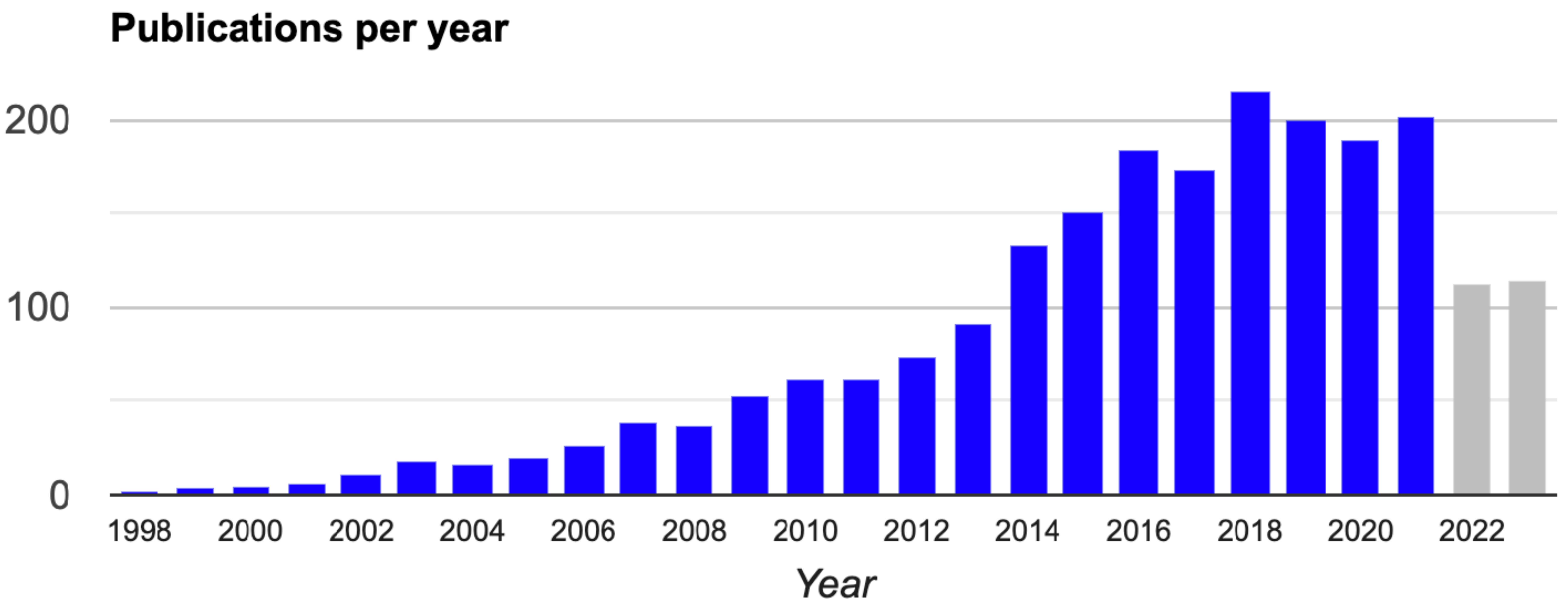
Did it pay back?

- We added support for co-dimension one for **BEM**
  - Andrea Bonito and Sebastian Pauletti started using it for **PDEs on surfaces** (step-38, two releases later)
- Rene Gassmöller added Particles support
  - Bruno Blais and myself combined the two things into Nitsche FSI
    (step-70, many releases later)
- My PhD code (created with deal.II 6.0.0) **runs on deal.II 9.3 40% times faster** *without changes on my side*

The whole is much greater than the sum of its parts!

SISSA

Publications referencing deal.II



Known publications using deal.II. Total: **2190**. (Gray bars: Incomplete data.)

Publications referencing deal.II

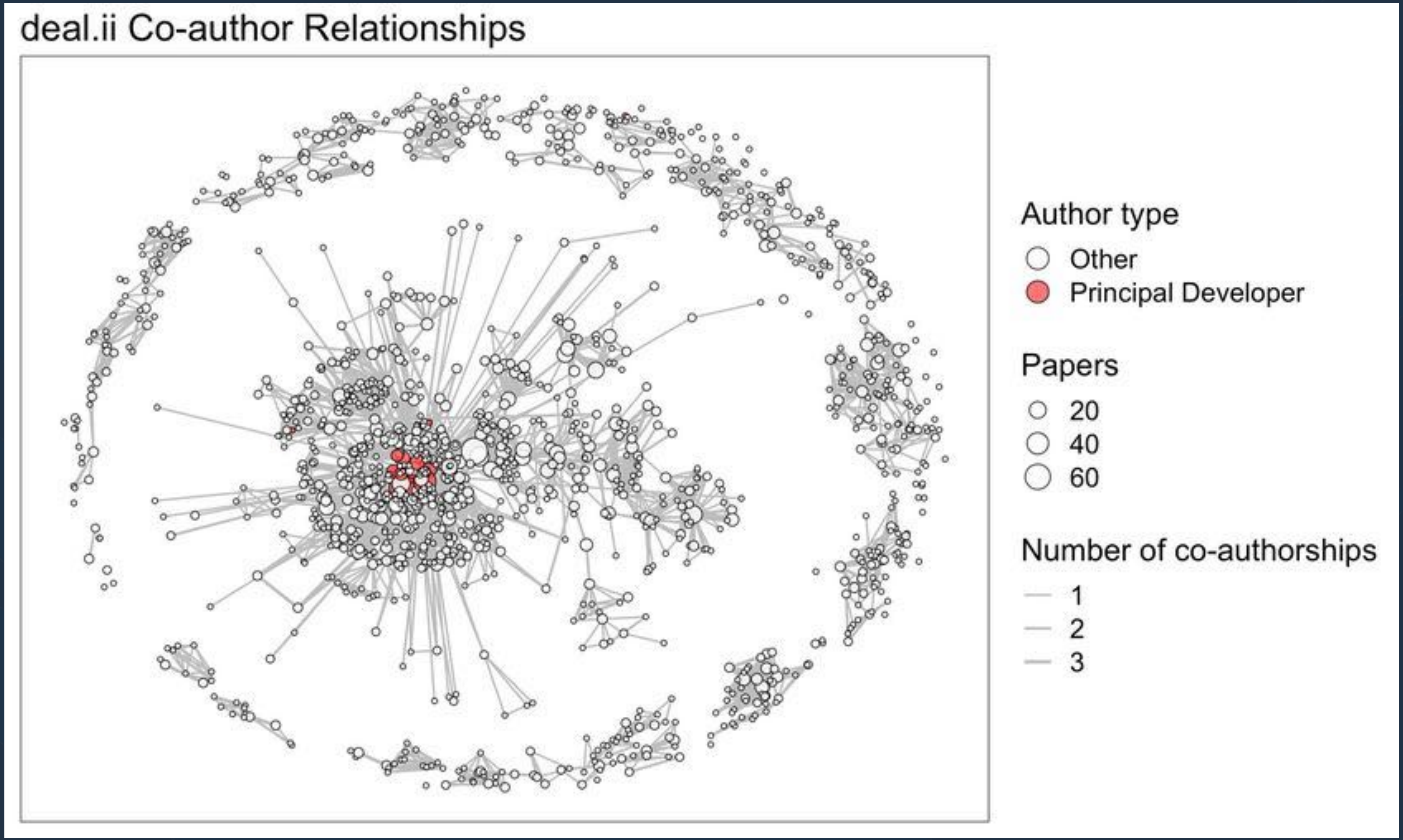


Image credit: Lorraine Hwang

How do we make this sustainable?

The little important things…

- **Strict** version control workflow prerequisites:
  - one feature = one pull request (*we have 5 PR per day/every day, on average!*)
  - strong enforcement of test passing (12K+ tests per PR)
  - strong enforcement of minimal indentation and documentation checks (clang-format and doxygen)
- **Strict** peer review
  - review starts when tests pass
  - no-one merge their own pull request
  - **no-documentation = no-merge**
  - **no-test = no-merge**

SISSA

Workshops, schools, and courses

The little important things…

- We use deal.II in our courses (FEM 101 through FEM 501)
- We try to teach best programming practices
- We encourage PhD students to write
  - sustainable code
  - shareable code
  - reusable code

SISSA

How do we attract new contributors?

The icing on the cake…

- **Students and postdocs** need recognitions for their software work
  - they become co-authors of new release papers if they contribute substantially to the library (one paper per year)
- **PIs** need recognition for their software work
  - making sustainable software *must* become part of evaluation criteria for promotions/grants/etc.
- **Making good CSE software is *not* an hobby — it is a job in its own right — also in academia**
- We need **more funding** for this to be sustainable

SISSA

Reproducibility

The icing on the cake…

- good science **needs** reproducible results **and** open source codes (no math paper is accepted with a "closed source theorem"! Why should a computational paper be accepted without the source code?)
- we are starting *now* to get "**reproducibility badges**" on articles that share the codes used to reproduce the results — deterministically (example: https://codeocean.com/capsule/1296846/tree/v1)
- @ deal.II, we try to encourage this in all possible ways (i.e., docker images are deployed at each successful merge of deal.II — these can be used to test the library, compile online, provide controlled environment for software deployment, etc.

SISSA

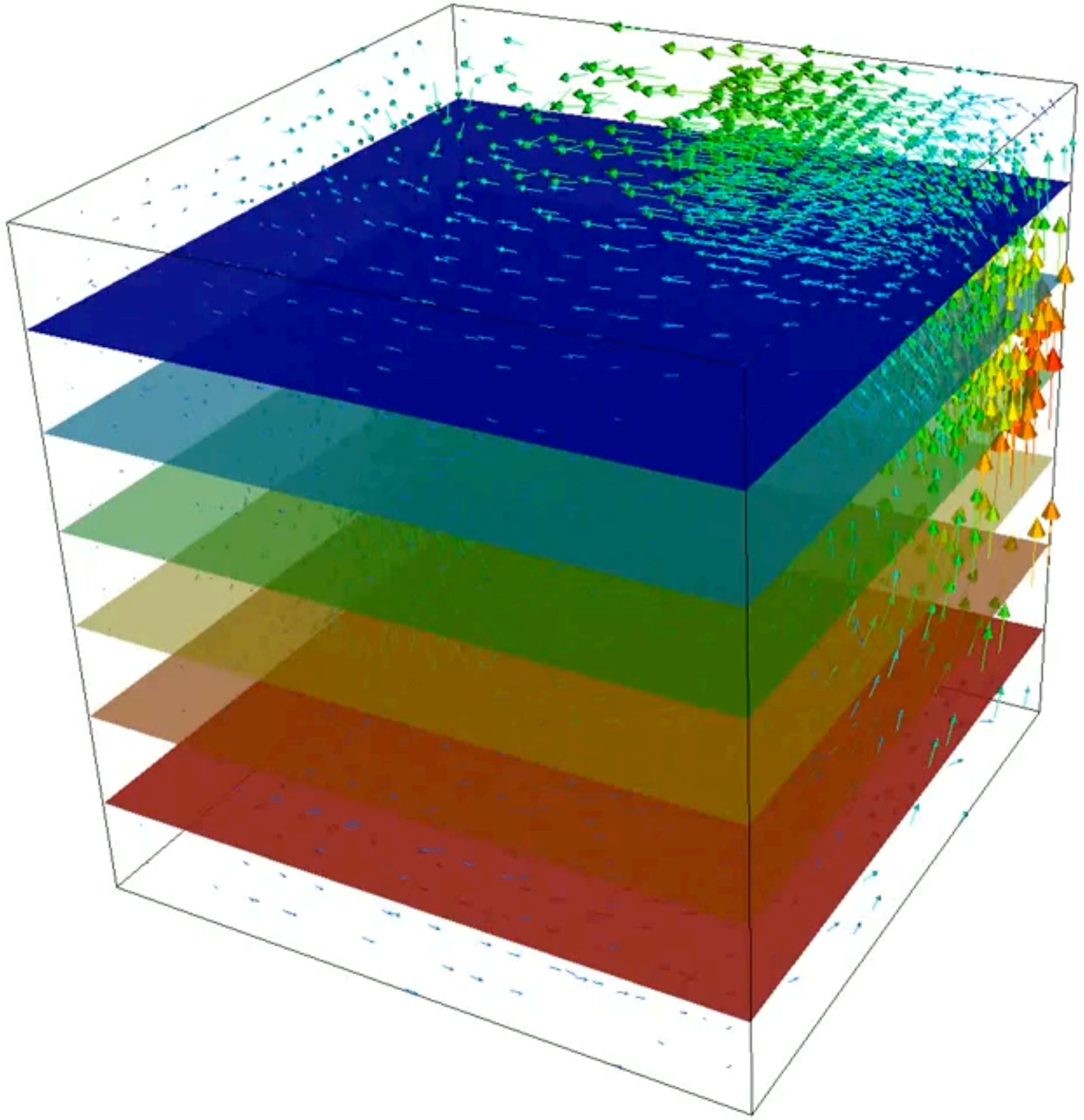
Aspect

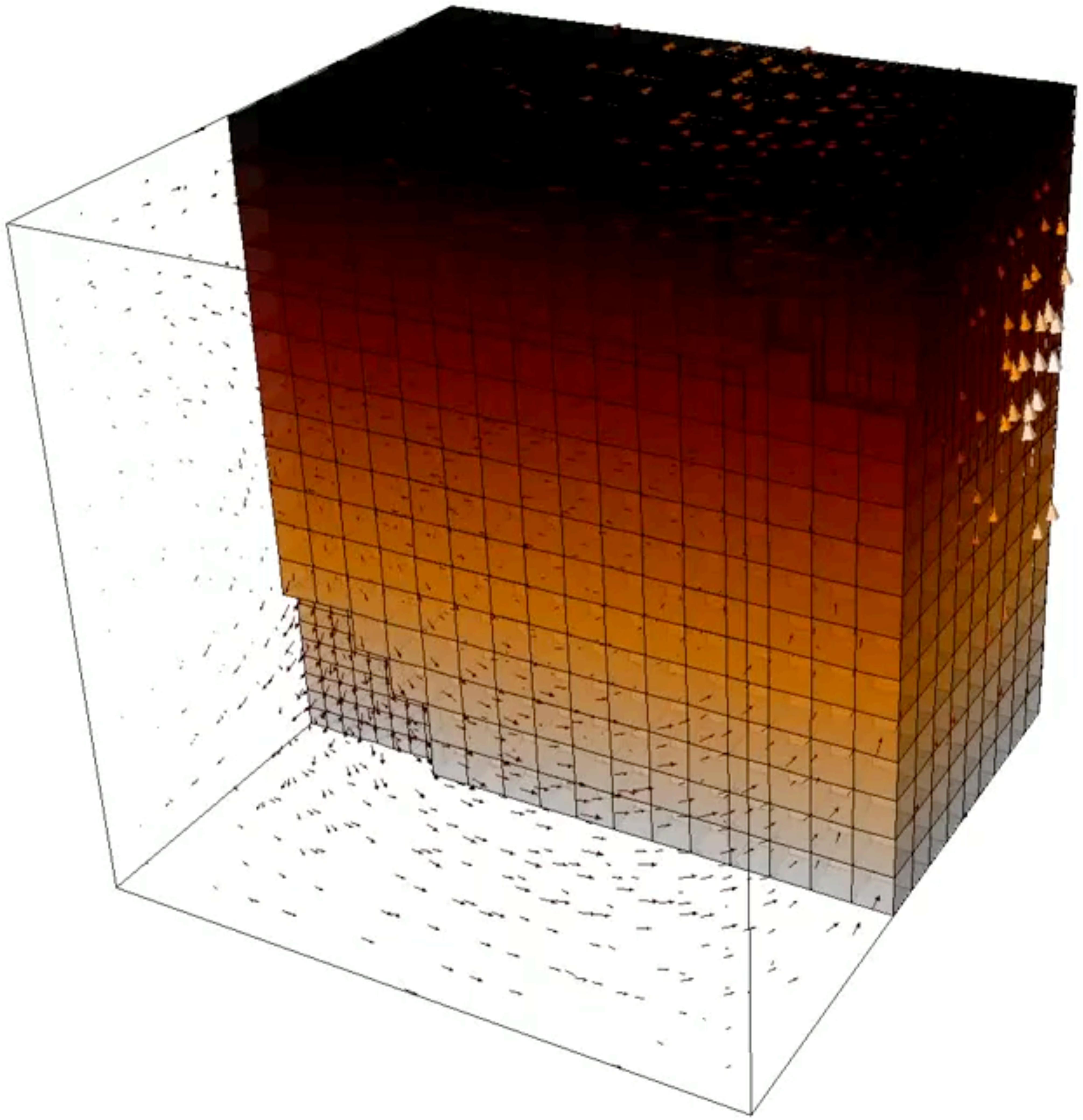
Advanced **S**olver for **P**roblems in **E**arth's **C**onvec**T**ion

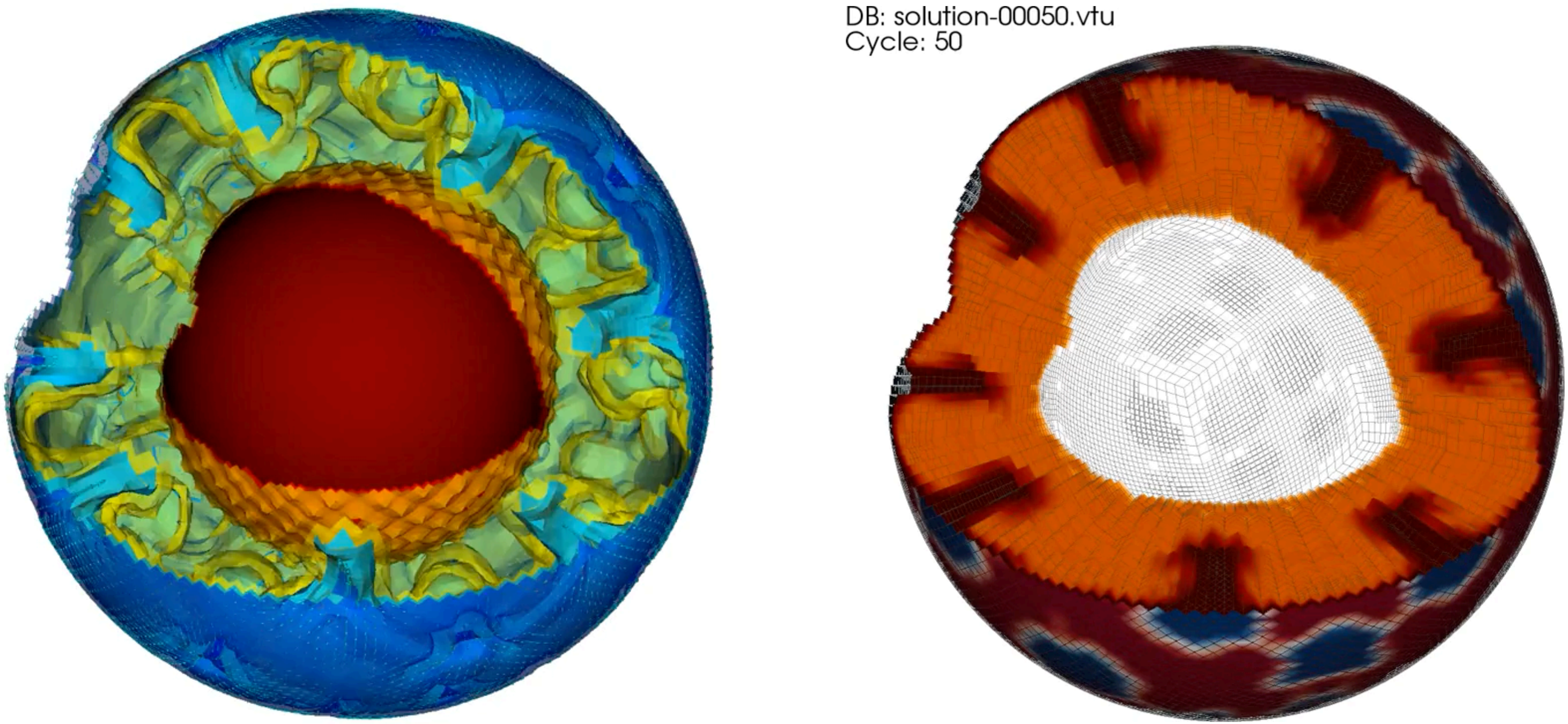
Martin Kronbichler, Timo Heister, and Wolfgang Bangerth. 2012. "**High Accuracy Mantle Convection Simulation through Modern Numerical Methods.**" Geophysical Journal International 191 (1) (August 21): 12–29. doi:10.1111/j.1365-246x.2012.05609.x. http://dx.doi.org/10.1111/j.1365-246X.2012.05609.x.

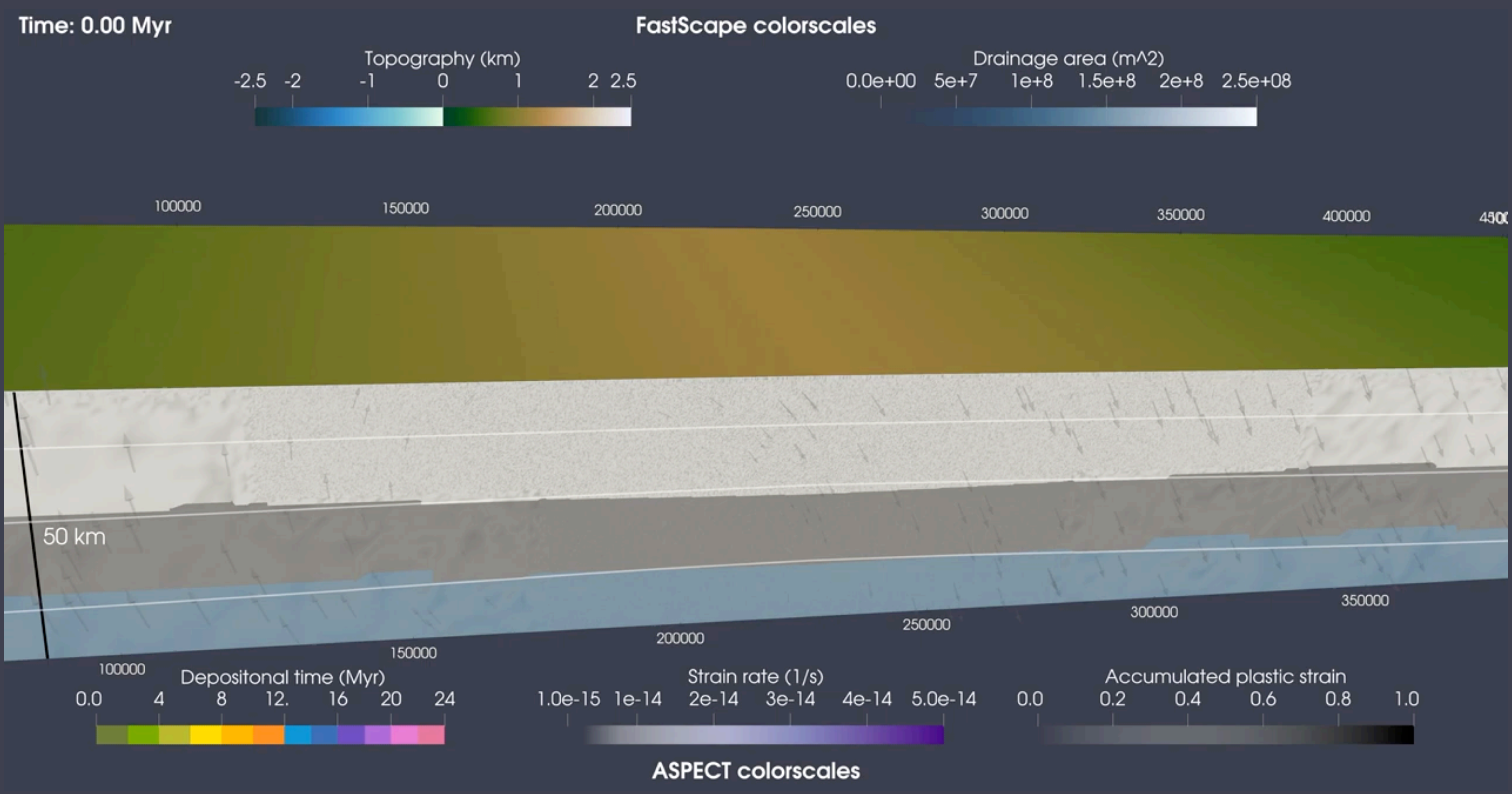
Timo Heister, Juliane Dannberg, Rene Gassmöller, and Wolfgang Bangerth. 2017. "**High Accuracy Mantle Convection Simulation through Modern Numerical Methods – II: Realistic Models and Problems**." Geophysical Journal International 210 (2) (May 9): 833–851.

Bangerth, Wolfgang, Juliane Dannberg, Menno Fraters, Rene Gassmoeller, Anne Glerum, Timo Heister, Robert Myhill, and John Naliboff. 2022. **ASPECT v2.4.0** (version v2.4.0)






DB: solution-00050.vtu
Cycle: 50



Evolution of Rift Systems and Their Fault Networks in Response to Surface Processes.

Derek Neuharth, Sascha Brune, Thilo Wrona, Anne Glerum, Jean Braun, Xiaoping Yuan. Tectonics. Volume 41, Issue3, 2022.

Ryujin

Efficient parallel 3D computation of the compressible Euler equations with an invariant-domain preserving second-order finite-element scheme
Matthias Maier and Martin Kronbichler
ACM Transactions on Parallel Computing, 2021, Vol 8, n 3,16:1—30

Second-order invariant domain preserving approximation of the compressible Navier—Stokes equations
Jean-Luc Guermond and Matthias Maier and Bojan Popov and
Ignacio Tomas
Computer Methods in Applied Mechanics and Engineering, 2021, Vol 375, n 1, 113608

On the implementation of a robust and efficient finite element-based parallel solver for the compressible Navier-stokes equations
Jean-Luc Guermond and Martin Kronbichler and Matthias Maier and Bojan Popov and Ignacio Tomas
Computer Methods in Applied Mechanics and Engineering2022, Vol 389, 114250.

https://github.com/conservation-laws/ryujin

SISSA

LifeX (project iHeart)

A comprehensive and biophysically detailed comp. model of the whole human heart electromechanics
M.Fedele, R.Piersanti, F.Regazzoni, M.Salvador, P.C.Africa, M.Bucelli, A.Zingaro, L.Dede', A.Quarteroni,

Modeling cardiac muscle fibers in ventricular and atrial electrophysiology simulations
R.Piersanti, P.C. Africa, M.Fedele, C.Vergara, L.Dede', A.F.Corno, A.Quarteroni
Journal: Computer Methods in Applied Mechanics and Engineering,
vol. 373, p. 113468-113500 (2021)

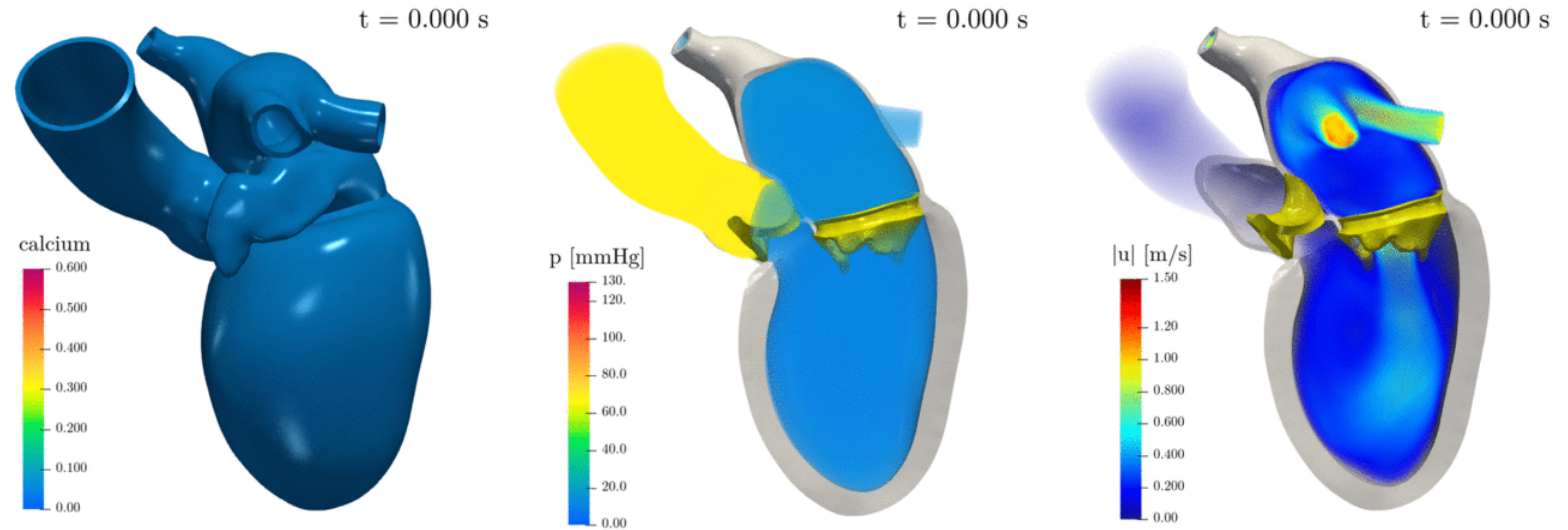
An electromechanics-driven fluid dynamics model for the simulation of the whole human heart
Zingaro, M. Bucelli, R. Piersanti, F. Regazzoni, L. Dede' and A. Quarteroni.

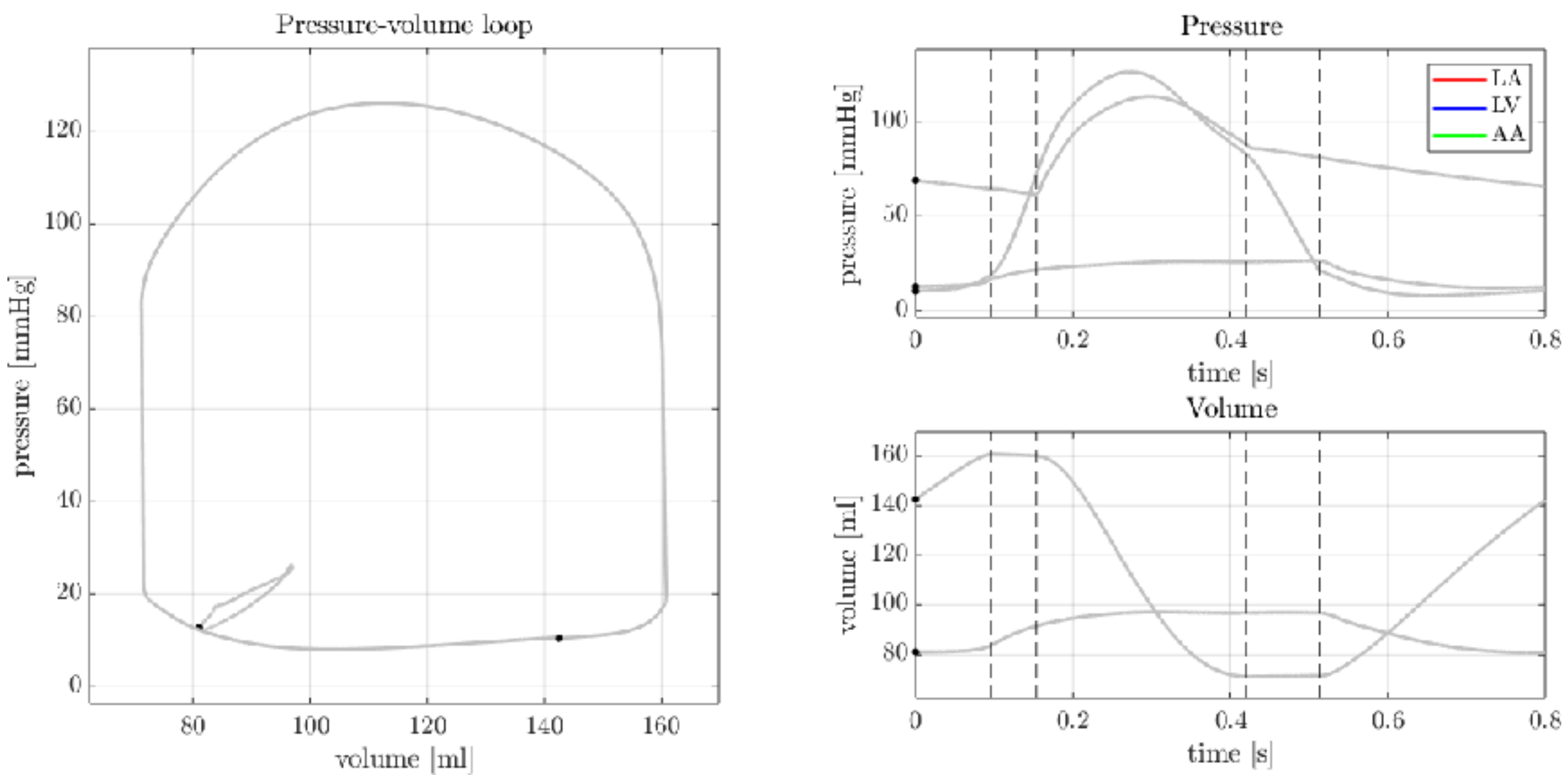
A geometric multiscale model for the numerical simulation of blood flow in the human left heart
Zingaro, I. Fumagalli, L. Dede', M. Fedele, P.C. Africa, A.F. Corno, A. Quarteroni.
Discrete and Continuous Dynamical System – Series S, 15(8), 2391-2427, 2022

A mathematical model that integrates cardiac electrophysiology, mechanics and fluid dynamics: application to the human left heart.
Bucelli, M., Zingaro, A., Africa, P. C., Fumagalli, I., Dede', L., & Quarteroni, A. (2022).
International Journal for Numerical Methods in Biomedical Engineering, e3678

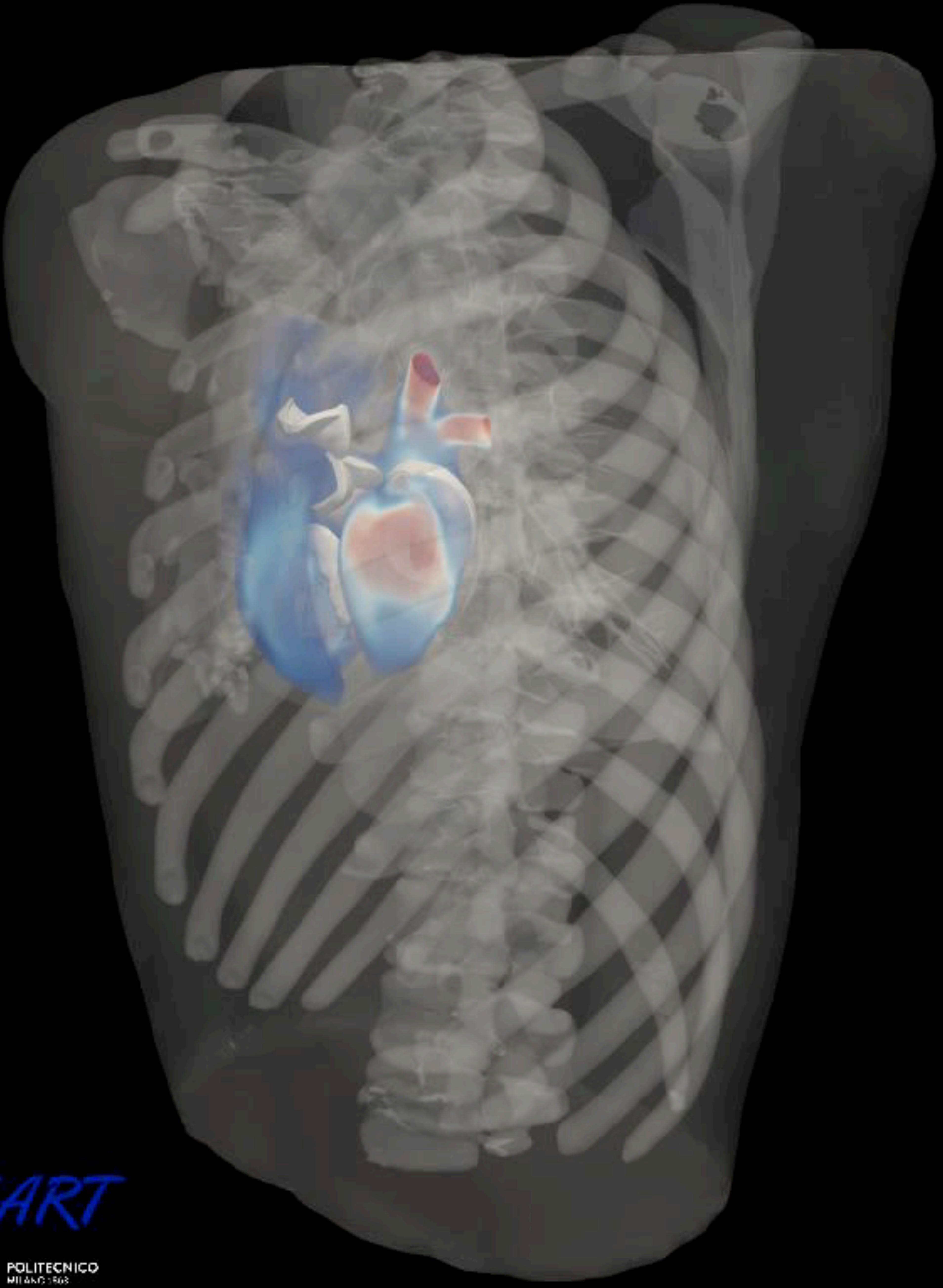
https://gitlab.com/lifex/lifex

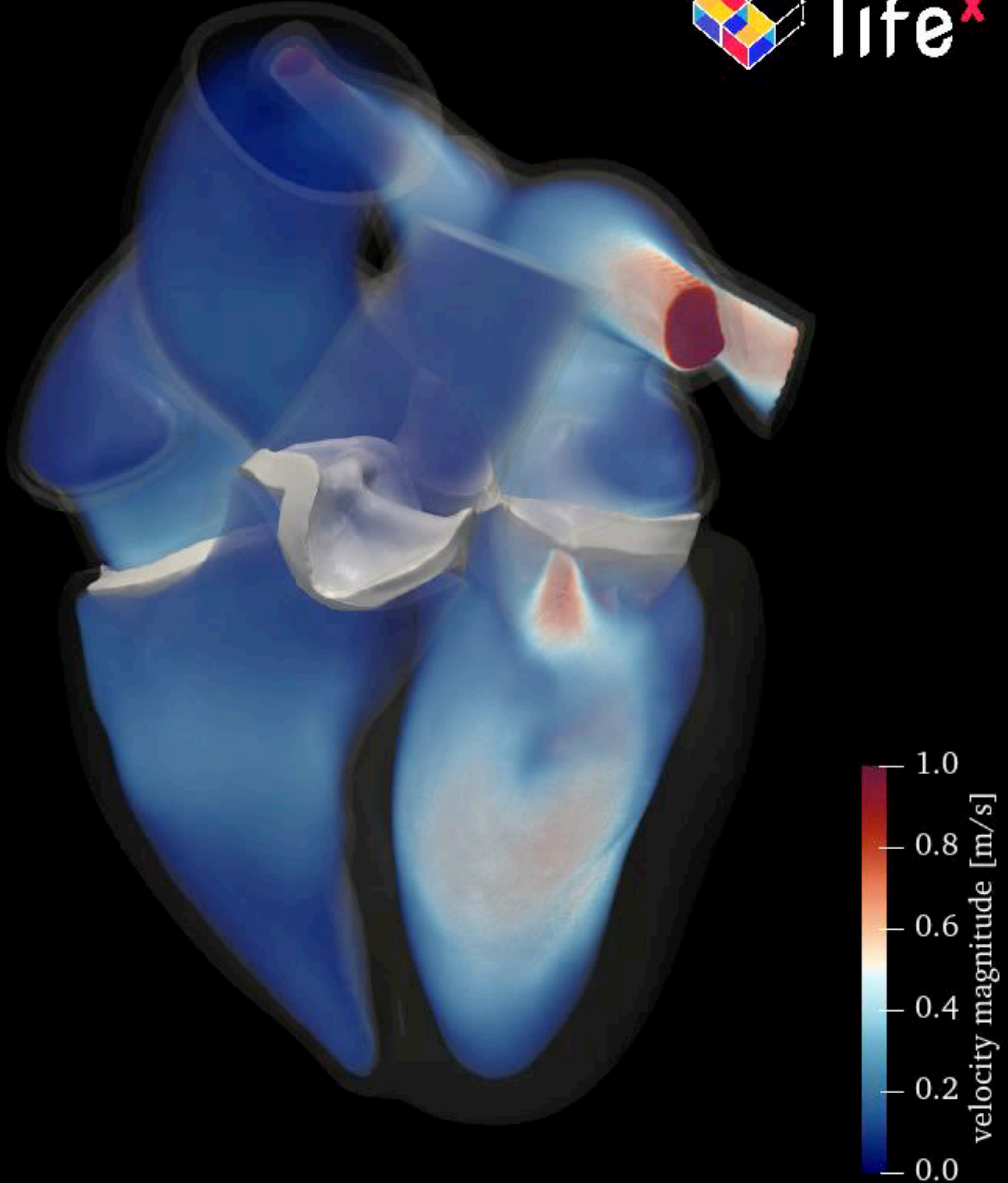
SISSA



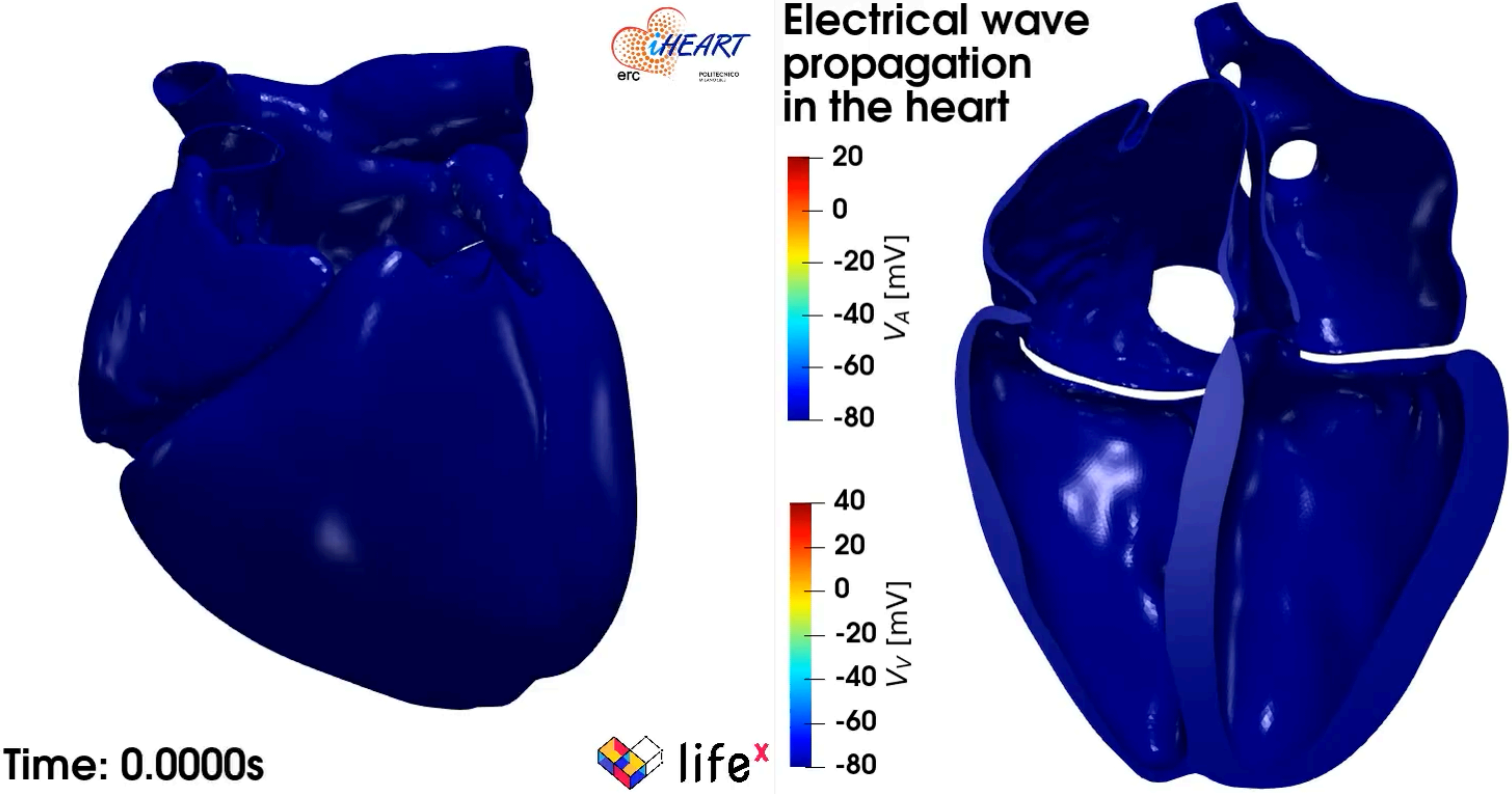


LifeX - iHeart




1.0
0.8
0.6
0.4
0.2
0.0
velocity magnitude [m/s]

Electrical wave propagation in the heart



Lethe-CFD

Lethe: An open-source parallel high-order adaptative CFD solver for incompressible flows
Bruno Blais, Lucka Barbeau, Valérie Bibeau, Simon Gauvin, Toni El Geitani, Shahab Golshan, Rajeshwari Kamble, Ghazaleh Mirakhori, Jamal Chaouki
SoftwareX, 2020, Vol 12, 100579

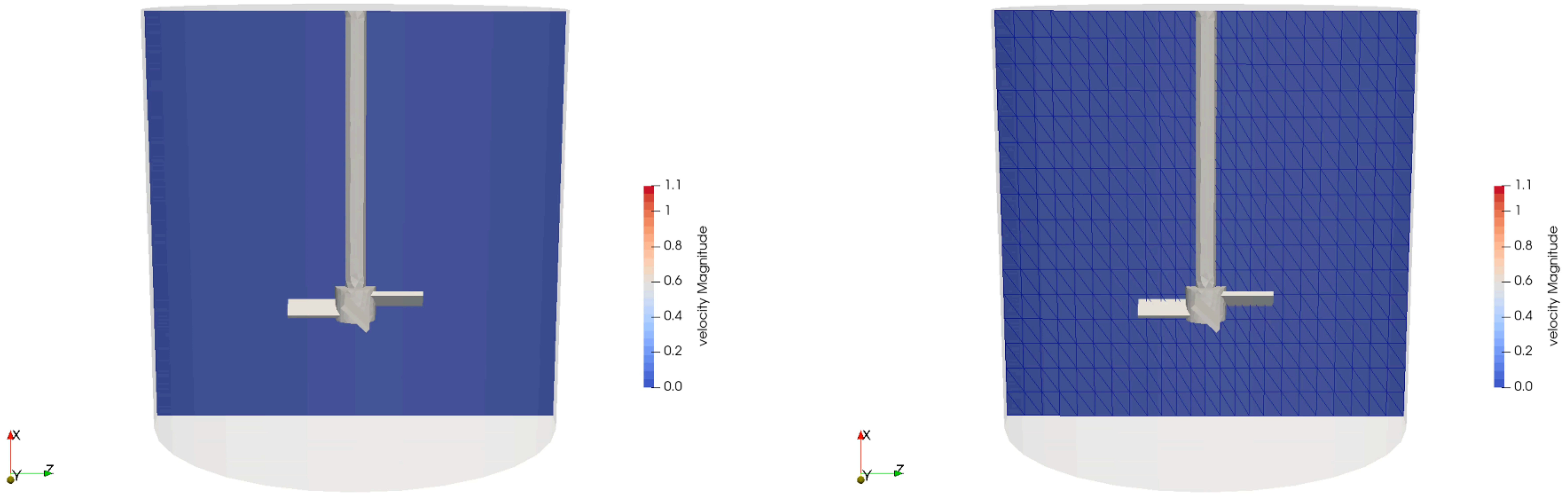
https://github.com/lethe-cfd/lethe

Combined with Particles + step-70 (LH, Bruno Blais, Rene Gassmöller)

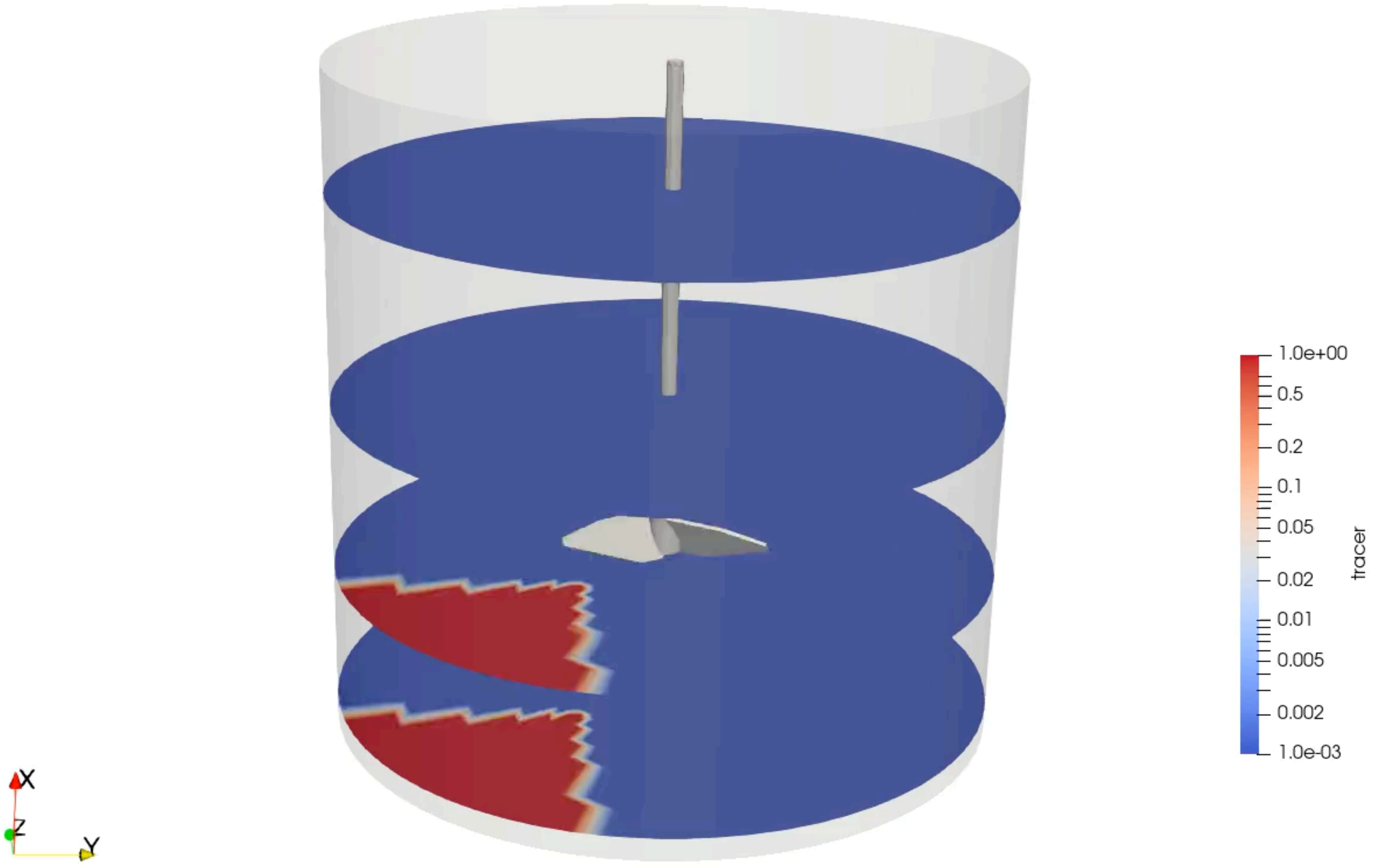
now Lethe-CFD supports DEM

SISSA

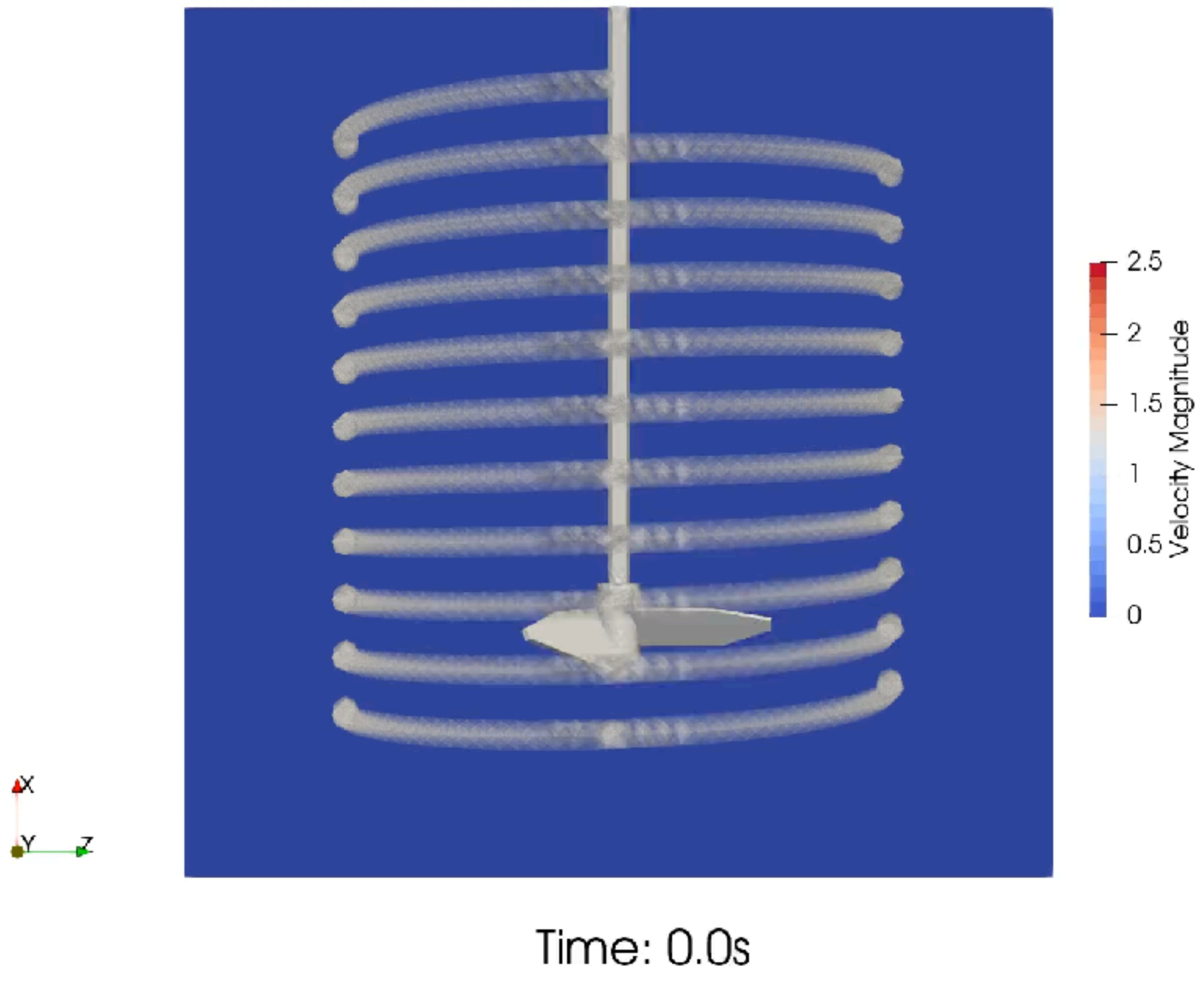
Time: 0.00

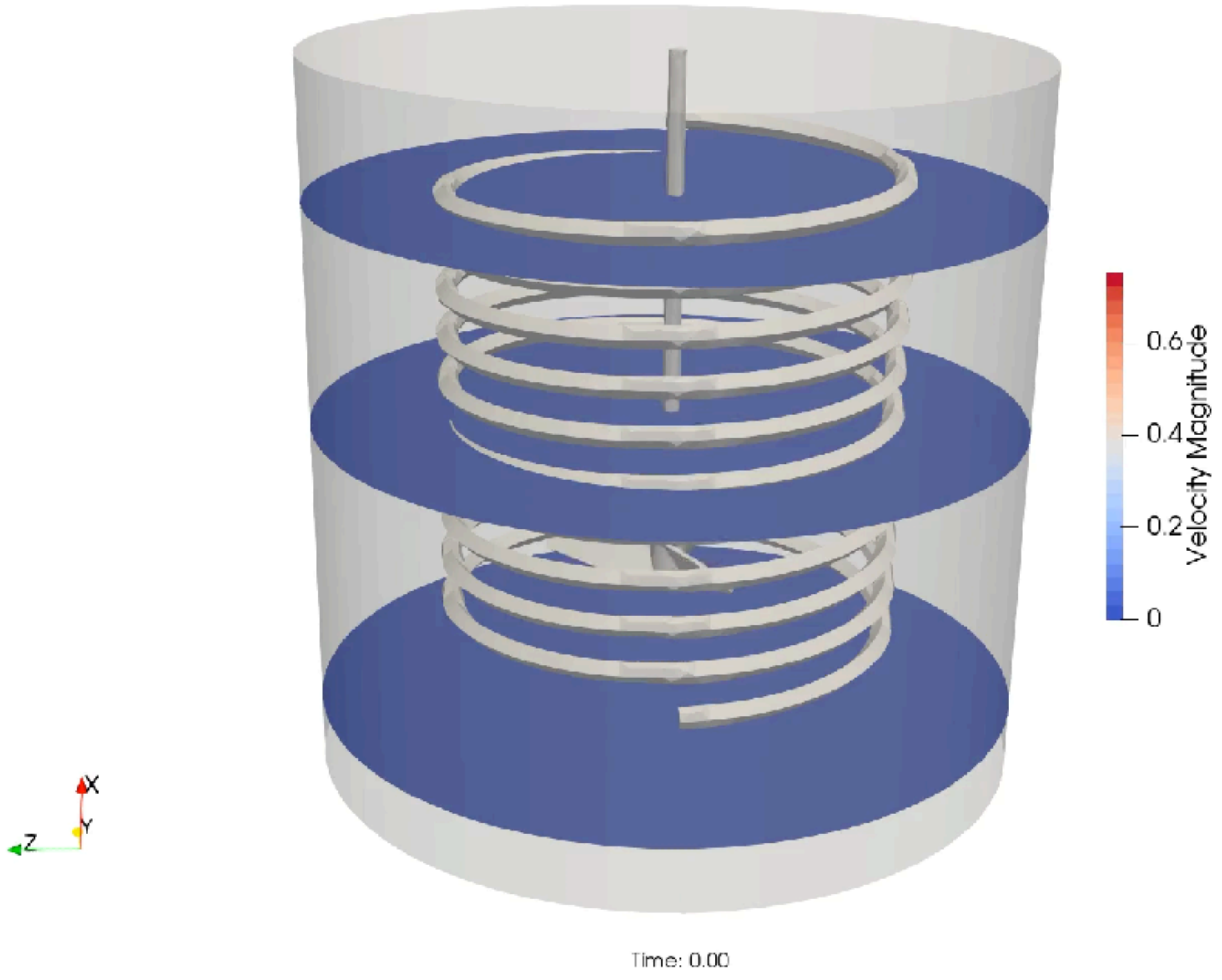


velocity Magnitude
1.1
1
0.8
0.6
0.4
0.2
0.0
X
Y
Z
velocity Magnitude
1.1
1
0.8
0.6
0.4
0.2
0.0
X
Y
Z



Time: 0.00s


2.5
2
1.5
1
0.5
0
Velocity Magnitude
X
Y
Z
Time: 0.0s


0.6
0.4
0.2
0
Velocity Magnitude
X
Y
Z
Time: 0.00

Exploiting high-contrast Stokes preconditioners to efficiently solve incompressible fluid-structure interaction problems,
M. Wichrowski, S.Stupkiewicz, P.Krzyzanowski, **LH,** 2023, International Journal for Numerical Methods in Engineering.

ALE - iFSI — RE = 2000
~30 sec wall time per 0.01 time step
10k dofs/(s cores) (3D)
32k dofs/(s cores) (2D)