The Journal of Open Source Software

turtleFSI: A Robust and Monolithic FEniCS-based Fluid-Structure Interaction Solver

Aslak W. Bergersen[1], Andreas Slyngstad[1], Sebastian Gjertsen[1], Alban Souche[1], and Kristian Valen-Sendstad[1]

1 Department of Computational Physiology, Simula Research Laboratory, Fornebu, Norway

DOI: 10.21105/joss.02089

Software

- Review
- Repository
- Archive

Editor: Kevin M. Moerman

Reviewers:

- @chennachaos
- @JaroslavHron

Submitted: 06 February 2020
Published: 16 June 2020

Summary

It is often sufficient to study fluids (Moin & Mahesh, 1998) and solids (Holzapfel, 2002) in isolation to gain fundamental insights into a physical problem, as other factors may play a secondary role and can be neglected. On the other hand, there are certain phenomena or situations where the stresses on or by a fluid or a solid can lead to large deformations, and the interaction between fluids and solids are essential (Le Tallec & Mouro, 2001). Computational fluid-structure interaction (FSI) is an active field of research with much focus on numerical accuracy, stability, and convergence rates. At the same time, there is also a sweet spot in between these areas of research where there is a need to experiment with FSI without having an in-depth, bottom-up mathematical understanding of the problem, but where a physical insight might suffice. Therefore, the aim was to develop a fully monolithic and robust entry-level research code with ease-of-use targeted towards students, educators, and researchers.

FEniCS (Logg, Mardal, & Wells, 2012) has emerged as one of the leading platforms for development of scientific software due to the close connection between mathematical notation and compact computer implementation, where highly efficient C++ code is compiled during execution of a program. Combined with the out-of-the-box entry-level high-performance computing capabilities, FEniCS was a natural choice of computing environment. Compared to other open-source FSI solvers (Heil & Hazel, 2006; Jasak, Jemcov, Tukovic, & others, 2007; Malinen & Råback, 2013), turtleFSI is written in only a couple of hundred lines of high-level Python code, in contrast to tens of thousands of lines of low-level C++ code. This provides full transparency and a unique opportunity for researchers and educators to modify and experiment with the code, while still providing out of the box entry-level high-performance computing capabilities. Furthermore, because of the close resemblance between mathematics and code in FEniCS, users can make additions or modifications with ease.

The turtleFSI solver relies on a fully monolithic approach in the classical arbitrary Lagrangian-Eulerian formulation, and we used the generalized theta scheme for temporal discretization and P2P1P2 elements for velocity, pressure, and displacement, respectively. We implemented and evaluated four different mesh lifting operators, ranging from a simple and efficient second-order Laplace equation, most suitable for small deformations, to more sophisticated and computationally expensive 4th order bi-harmonic equations that can handle larger mesh deformations. We used The Method of Manufactured Solutions to verify the implementation. The obtained results are formally second-order accurate (L2) in space and time (Wick, 2011), respectively, and we demonstrate that all building blocks of code exhibit desired properties. The solver's validity was confirmed using the classical Turek Flag benchmark case (Turek & Hron, 2006) with a good agreement – including a diverged numerical solution for long term evolution under certain conditions, as expected. For a complete justification of computational approaches and further details, we refer to (Gjertsen, 2017; Slyngstad, 2017). We demonstrate adequate

strong scaling up to 64 cores (from one cluster node), although the latter is problem size-dependent. In the online documentation, we provide benchmarks, tutorials, and simple demos. The naive FEniCS implementation provides full transparency with compact code, which can easily be adapted to other 2D or 3D FSI problems.

In conclusion, turtleFSI is not a superior FSI solver in terms of speed, but it is a robust entry-level FSI solver and performs exactly as designed and intended; 'slow and steady wins the race'.

turtleFSI in Action

turtleFSI comes with several problem files, found under /turtleFSI/problems/, to illustrate the usage and document the Turek flag benchmarks used to validate the implementation of the solver. Here are some illustrations of the execution and outputs expected from the solver.

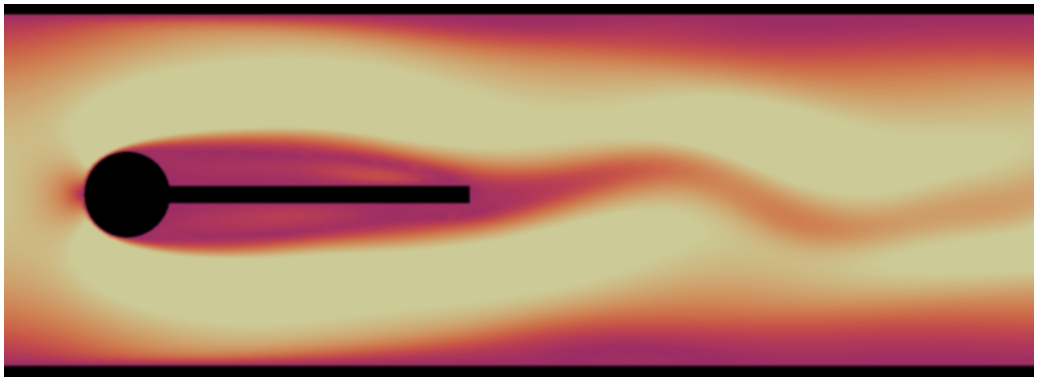
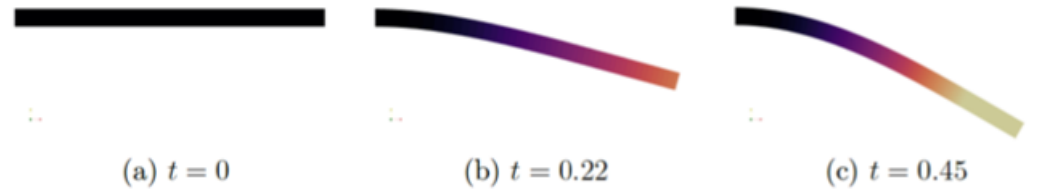


Figure 1: Fluid dynamics benchmark snapshot. Simulation executed with the command: `turtleFSI --problem TF_cfd`



(a) $t = 0$ (b) $t = 0.22$ (c) $t = 0.45$

Figure 2: Solid mechanics benchmark snapshots. Simulation executed with the command: `turtleFSI --problem TF_csm`

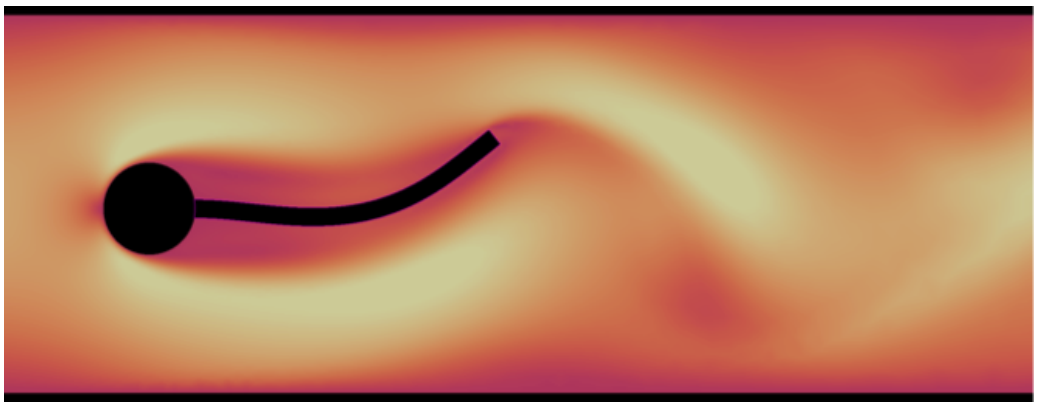


Figure 3: Full fluid-structure interaction benchmark snapshot. Simulation executed with the command: `turtleFSI --problem TF_fsi`

Acknowledgements

The study was supported by The Research Council of Norway through the Center for Biomedical Computing (grant 179578), the Centre for Cardiological Innovation (grant number 203489), and the SIMMIS project (grant number 262827). Simulations were performed on the Abel Cluster (University of Oslo and the Norwegian metacenter for High Performance Computing (NOTUR), project nn9316k), and the Experimental Infrastructure for Exploration of Exascale Computing (eX3) cluster (Norwegian Research Council grant 270053).

References

Gjertsen, S. (2017). Development of a verified and validated computational framework for fluid-structure interaction: Investigating lifting operators and numerical stability. *MSc Thesis, University of Oslo*.

Heil, M., & Hazel, A. L. (2006). Oomph-lib–an object-oriented multi-physics finite-element library. In *Fluid-structure interaction* (pp. 19–49). Springer. doi:10.1007/3-540-34596-5_2

Holzapfel, G. A. (2002). Nonlinear solid mechanics: A continuum approach for engineering science. *Meccanica*, *37*(4), 489–490. doi:10.1023/A:1020843529530

Jasak, H., Jemcov, A., Tukovic, Z., & others. (2007). OpenFOAM: A c++ library for complex physics simulations. In *International workshop on coupled methods in numerical dynamics* (Vol. 1000, pp. 1–20). IUC Dubrovnik Croatia.

Le Tallec, P., & Mouro, J. (2001). Fluid structure interaction with large structural displacements. *Computer methods in applied mechanics and engineering*, *190*(24), 3039–3067. doi:10.1016/S0045-7825(00)00381-9

Logg, A., Mardal, K.-A., & Wells, G. (2012). *Automated solution of differential equations by the finite element method: The fenics book*. (Vol. 84). Springer Science & Business Media. doi:10.1007/978-3-642-23099-8

Malinen, M., & Råback, P. (2013). Elmer finite element solver for multiphysics and multiscale problems. *Multiscale Model. Methods Appl. Mater. Sci.*, *19*, 101–113.

Moin, P., & Mahesh, K. (1998). Direct numerical simulation: A tool in turbulence research. *Annual review of fluid mechanics*, *30*(1), 539–578. doi:10.1146/annurev.fluid.30.1.539

Slyngstad, A. (2017). Verification and validation of a monolithic fluid-structure interaction solver in fenics. A comparison of mesh lifting operators. *MSc Thesis, University of Oslo*.

Turek, S., & Hron, J. (2006). Proposal for numerical benchmarking of fluid-structure interaction between an elastic object and laminar incompressible flow. In *Fluid-structure interaction* (pp. 371–385). Springer. doi:10.1007/3-540-34596-5_15

Wick, T. (2011). Fluid-structure interactions using different mesh motion techniques. *Computers & Structures*, *13*(89), 1456–1467. doi:10.1016/j.compstruc.2011.02.019