The Journal of Open Source Software

pulse: A python package based on FEniCS for solving problems in cardiac mechanics

Henrik Nicolay Topnes Finsberg[1]

1 Simula Research Laboratory, Oslo, Norway

DOI: 10.21105/joss.01539

Software

- Review
- Repository
- Archive

Submitted: 25 May 2019
Published: 24 September 2019

Summary

`pulse` is a software package based on FEniCS (Logg, Mardal, & Wells, 2012) that aims to solve problems in cardiac mechanics (but is easily extended to solve more general problems in continuum mechanics). `pulse` is a result of the author's PhD thesis (H. N. Finsberg, 2017), where most of the relevant background for the code can be found.

While FEniCS offers a general framework for solving PDEs, `pulse` specifically targets problems in continuum mechanics. Therefore, most of the code for applying compatible boundary conditions, formulating the governing equations, choosing appropriate spaces for the solutions and applying iterative strategies, etc., are already implemented, so that the user can focus on the actual problem he/she wants to solve rather than implementing all the necessary code for formulating and solving the underlying equations.

The user can pick any of the built-in meshes or choose a custom user-defined mesh. The user also need to provide appropriate markers for the boundaries where the boundary conditions will be applied, as well as microstructural information (i.e., information about muscle fiber orientations) if an anisotropic model is to be used. Examples of how to create custom idealized geometries as well as appropriate microstructure can be found in another repository called ldrb, which uses the Laplace-Dirichlet Rule-Based (LDRB) algorithm (Bayer, Blake, Plank, & Trayanova, 2012) for assigning myocardial fiber orientations.

Next the user needs to select a material model or create a custom material model, and define appropriate boundary conditions (Dirichlet, Neumann, or Robin boundary conditions). Finally a `MechanicsProblem` is built using the geometry, material, and boundary conditions. Figure 1 shows the different components involved as well as how they are related.

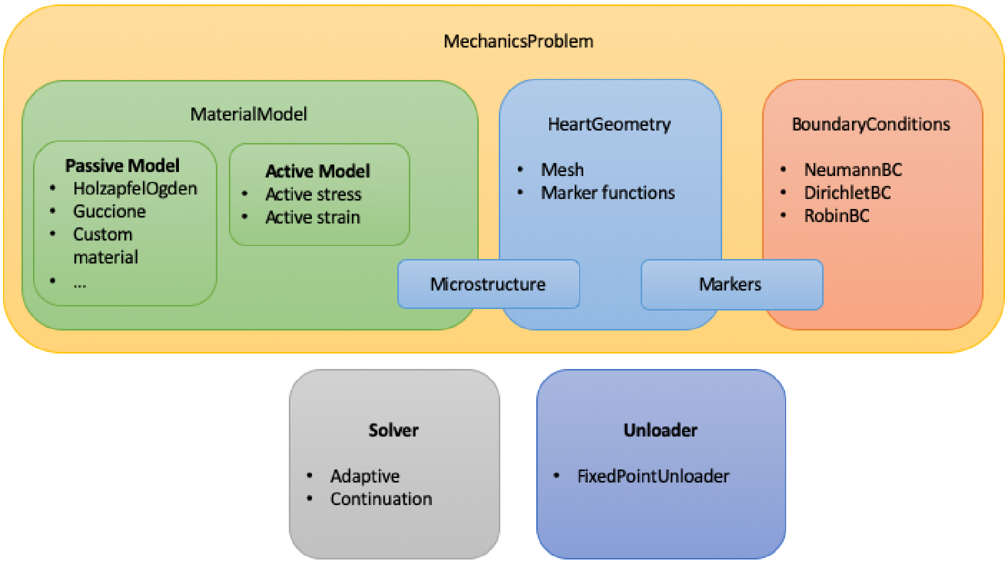

Figure 1: Visualization of the different components that are part of the `pulse` library.

The problem is solved using some iterative strategy, either with an incremental load technique with fixed or adaptive stepping and/or using with a continuation technique (H. N. Finsberg, 2017).

It is also possible to estimate the unloaded zero-pressure geometry (Bols et al., 2013). This is of particular importance if the geometry being used is taken from a medical image of a patient. In this case, the geometry is subjected to some load due to the blood pressure, and therefore in order to correctly assess the stresses, one need to first find the unloaded geometry.

Papers using this code includes (H. Finsberg et al., 2018a) and (H. Finsberg et al., 2018b).

A collection of different demos showing how to use the `pulse` library is found in the repository, including an implementation of a cardiac mechanics bechmark (Land et al., 2015), how to use a custom material model, and how to use a compressible model rather than the default incompressible model.

References

Bayer, J. D., Blake, R. C., Plank, G., & Trayanova, N. A. (2012). A novel rule-based algorithm for assigning myocardial fiber orientation to computational heart models. *Annals of Biomedical Engineering*, *40*(10), 2243–2254. doi:10.1007/s10439-012-0593-5

Bols, J., Degroote, J., Trachet, B., Verhegghe, B., Segers, P., & Vierendeels, J. (2013). A computational method to assess the in vivo stresses and unloaded configuration of patient-specific blood vessels. *Journal of Computational and Applied Mathematics*, *246*, 10–17. doi:10.1016/j.cam.2012.10.034

Finsberg, H. N. (2017). Patient-specific computational modeling of cardiac mechanics. *Series of dissertations submitted to the Faculty of Mathematics and Natural Sciences, University of Oslo.*

Finsberg, H., Balaban, G., Ross, S., Håland, T. F., Odland, H. H., Sundnes, J., & Wall, S. (2018a). Estimating cardiac contraction through high resolution data assimilation of a personalized mechanical model. *Journal of Computational Science*, *24*, 85–90. doi:10.1016/j.jocs.2017.07.013

Finsberg, H., Xi, C., Tan, J. L., Zhong, L., Genet, M., Sundnes, J., Lee, L. C., et al. (2018b). Efficient estimation of personalized biventricular mechanical function employing gradient-based optimization. *International Journal for Numerical Methods in Biomedical Engineering*, *34*(7), e2982. doi:10.1002/cnm.2982

Land, S., Gurev, V., Arens, S., Augustin, C. M., Baron, L., Blake, R., Bradley, C., et al. (2015). Verification of cardiac mechanics software: Benchmark problems and solutions for testing active and passive material behaviour. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *471*(2184), 20150641. doi:10.1098/rspa.2015.0641

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.