

Artery.FE: An implementation of the 1D blood flow equations in FEniCS

Syver D. Agdestein¹, Kristian Valen-Sendstad¹, and Alexandra K. Diem¹

¹ Department of Computational Physiology, Simula Research Laboratory

DOI: [10.21105/joss.01107](https://doi.org/10.21105/joss.01107)

Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Submitted: 01 November 2018

Published: 18 December 2018

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC-BY](#)).

Summary

This package implements the 1D blood flow equations (Olufsen et al., 2000) using the finite element framework FEniCS (Alnaes et al., 2015). The package provides tools for modelling blood flow through a network of arteries by solving a 1D approximation of the cross-sectional area and flow rate of each artery.

FEniCS (Alnaes et al., 2015) is a finite element framework that is being increasingly used in cardiovascular and cardiac modelling. Thus, having a ready-to-use implementation of the 1D blood flow equations within the FEniCS framework is valuable for easy-to-use integration with whole-heart models of blood supply to the heart. To the best of our knowledge, this is not currently available. Most commonly, the 1D blood flow equations have previously been implemented using finite difference methods (Olufsen et al., 2000, Kolachalama, Bressloff, Nair, & Shearman (2007), Diem & Bressloff (2016)). Finite difference methods are easy to implement and thus easy to understand. However, the implementation of the grid is restricted to a fixed spatial step size. This limitation does not apply to the finite element method. In addition to providing tools for modelling artery networks and computing blood flow, this package may serve as a reference for others who intend to use FEniCS for vascular modelling. In a more general framework, the package may also be used as an example of how to solve equations in conservation form on multiple domains with shared boundary conditions.

A simulation is based on the parameter (.cfg) file provided by the user. The file consists of parameters for the geometry of the artery network, as well as intrinsic parameters for each artery. The user is also able to specify parameters for the solver, and output parameters. Flow through the artery network is driven by flow at the inlet of the root artery provided by the user.

The main part of the package is the class `Artery_Network`, which provides methods for describing the geometry of the network of arteries, meshing the arteries, computing boundary conditions, and solving the governing equations for cross-sectional area and flow rate. Each artery is defined by the `Artery` class. The boundary conditions are defined within the `Artery` class and evaluated in the `Artery_Network` class.

The module provides tools for running simulations, interpreting output and visualising data. An example run-file is set up to read parameters from a config (.cfg) file and to run the necessary functions from `Artery_Network` in the right order, and thus generating an output folder containing the solution to the 1D blood flow equations. Functions related to the manipulation and interpretation of output are provided in the file `utils.py`, while post-processing functions that create the appropriate figures from the solutions are provided in `postprocess.py`.

Method

To solve the 1D blood flow equations, a Crank-Nicolson finite difference scheme has been implemented in time, and a FEniCS-solver has been used for the spatial variable.

The dimensionalised problem may be written on conservation form:

$$\frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} = S(U)$$

where $U = (A, q)$, $F(U) = (q, \frac{q^2}{A} + f(r_0)\sqrt{A_0A})$, $S(U) = (0, -\frac{2\sqrt{\pi}}{\delta_b Re} \frac{q}{\sqrt{A}} + (2\sqrt{A}(\sqrt{\pi}f(r_0) + \sqrt{A_0} \frac{df}{dr_0}(r_0)) - A \frac{df}{dr_0}(r_0)) \frac{dr_0}{dx})$. A is the cross-sectional area, q is the flow.

The above system of equations governs the time-development of the cross-sectional artery and blood flow rate in one artery. The associated boundary conditions are as follows: The initial radii respect the relation $r_0(x) = Ru(\frac{Rd}{Ru})^{x/L}$ and the initial cross-sectional area for each artery is thus $A = \pi(r_0(x))^2$. A prescribed flow rate at the inlet of the root artery serves as the inlet boundary condition. For each artery, the initial flow is computed according to the artery's share of the cross-sectional area at the bifurcation. At each bifurcation point, the area and the flow is computed using Richtmyer's two-step Lax-Wendroff method, by solving a system of equations with 18 unknowns. At the terminal arteries, pressure is computed using a Windkessel model and prescribed as the outlet boundary condition. When the pressure is known, the area can be computed from the state equation: $p - p_0 = f(r_0)(1 - \sqrt{\frac{A_0}{A}})$. The derivation of the governing system of equations and boundary conditions can be found in (Olufsen et al., 2000, Kolachalama et al. (2007), Diem & Bressloff (2016)).

Acknowledgements

The authors acknowledge the European Community for its financial support for KVS and AKD in the framework of the project CUPIDO (www.cupidoproject.eu) H2020-NMBP-2016 720834, as well as the Simula Research Laboratory for financial support for SDA.

References

- Alnaes, M. S., Blechta, J., Hake, J., Johansson, A., Kehlet, B., Logg, A., Richardson, J., et al. (2015). The fenics project version 1.5. *Archive of Numerical Software*. doi:[10.11588/ans.2015.100.20553](https://doi.org/10.11588/ans.2015.100.20553)
- Diem, A. K., & Bressloff, N. W. (2016). VaMpy: A python package to solve 1D blood flow problems. *Journal of Open Research Software*, 5(17). doi:[10.5334/jors.159](https://doi.org/10.5334/jors.159)
- Kolachalama, V. B., Bressloff, N. W., Nair, P. B., & Shearman, C. P. (2007). Predictive haemodynamics in a one-dimensional human carotid artery bifurcation. Part i: Application to stent design. *IEEE Transactions on Biomedical Engineering*, 54(5), 802–812. doi:[10.1109/tbme.2006.889188](https://doi.org/10.1109/tbme.2006.889188)
- Olufsen, M. S., Peskin, C. S., Kim, W. Y., Pedersen, E. M., Nadim, A., & Larsen, J. (2000). Numerical simulation and experimental validation of blood flow in arteries with structured-tree outflow conditions. *Annals of Biomedical Engineering*, 28(11), 1281–1299. doi:[10.1114/1.1326031](https://doi.org/10.1114/1.1326031)