The Journal of Open Source Software

Symfem: a symbolic finite element definition library

Matthew W. Scroggs[1]

1 Department of Engineering, University of Cambridge

DOI: 10.21105/joss.03556

Software

- Review
- Repository
- Archive

Editor: Nikoleta Glynatsi

Reviewers:

- @iammix
- @sigvaldm

Submitted: 17 July 2021
Published: 24 August 2021

Summary

The finite element method (FEM) (Ciarlet, 2002) is a popular method for numerically solving a wide range of partial differential equations (PDEs). To solve a problem using FEM, the PDE is first written in a weak form, for example: find $u \in V$ such that for all $v \in V$,

$$\int_\Omega \nabla u \cdot \nabla v = \int_\Omega f v, \tag{1}$$

where f is a known function, and Ω is the domain on which the problem is being solved. This form is then discretised by defining a finite-dimensional subspace of V—often called V_h—and looking for a solution $u_h \in V_h$ that satisfies the above equation for all functions $v_h \in V_h$. These finite-dimensional subspaces are defined by meshing the domain of the problem, then defining a set of basis functions on each cell in the mesh (and enforcing any desired continuity between the cells).

For different applications, there are a wide range of finite-dimensional spaces that can be used. Symfem is a Python library that can be used to symbolically compute basis functions of these spaces. The symbolic representations are created using Sympy (Meurer et al., 2017), allowing them to be easily manipulated using Sympy's functionality once they are created.

Statement of need

In FEM libraries, it is common to define basis functions so that they, and their derivatives, can quickly and efficiently be evaluated at a collection of points, thereby allowing full computations to be completed quickly. The libraries FIAT (Kirby, 2004) and Basix (Richardson et al., 2021)—which are part of the FEniCS project (Alnæs et al., 2015)—implement this functionality as stand-alone libraries. Many other FEM libraries define their basis functions as part of the core library functionality. It is not common to be able to compute a symbolic representation of the basis functions.

Symfem offers a wider range of finite element spaces than other FEM libraries, and the ability to symbolically compute basis functions. There are a number of situations in which the symbolic representation of a basis function is useful: it is easy to confirm, for example, that the derivatives of the basis functions have a certain desired property, or check what they are equal to when restricted to one face or edge of the cell.

Symfem can also be used to explore the behaviour of the wide range of spaces it supports, so the user can decide which spaces to implement in a faster way in their FEM code. Additionally, Symfem can be used to prototype new finite element spaces, as custom spaces can easily be added, then it can be checked that the basis functions of the space behave as expected.

As basis functions are computed symbolically in Symfem, it is much slower than the alternative libraries. It is therefore not suitable for performing actual finite element calculations. It should instead be seen as a library for research and experimentation.

References

Alnæs, M. S., Blechta, J., Hake, J., Johansson, A., Kehlet, B., Logg, A., Richardson, C., Ring, J., Rognes, M. E., & Wells, G. N. (2015). The FEniCS project version 1.5. *Archive of Numerical Software*, *3*(100), 9–23. https://doi.org/10.11588/ans.2015.100.20553

Ciarlet, P. G. (2002). *The finite element method for elliptic problems*. https://doi.org/10.1137/1.9780898719208

Kirby, R. C. (2004). Algorithm 839: FIAT, a new paradigm for computing finite element basis functions. *ACM Transactions on Mathematical Software*, *30*(4), 502–516. https://doi.org/10.1145/1039813.1039820

Meurer, A., Smith, C. P., Paprocki, M., Čertík, O., Kirpichev, S. B., Rocklin, M., Kumar, A., Ivanov, S., Moore, J. K., Singh, S., Rathnayake, T., Vig, S., Granger, B. E., Muller, R. P., Bonazzi, F., Gupta, H., Vats, S., Johansson, F., Pedregosa, F., … Scopatz, A. (2017). SymPy: Symbolic computing in Python. *PeerJ Computer Science*, *3*, e103. https://doi.org/10.7717/peerj-cs.103

Richardson, C., Scroggs, M., & Wells, G. (2021). *Basix: FEniCS runtime basis evaluation library*. https://github.com/FEniCS/basix