The Journal of Open Source Software

PyGDH: Python Grid Discretization Helper

Kenneth Higa[1], **Vincent S. Battaglia**[1], **and Venkat Srinivasan**[2]

1 Energy Technologies Area, Lawrence Berkeley National Laboratory **2** Argonne Collaborative Center for Energy Storage Science, Argonne National Laboratory

DOI: 10.21105/joss.02744

Software

- Review
- Repository
- Archive

Editor: Melissa Weber Mendonça

Reviewers:

- @JamieJQuinn
- @zhaowei0566

Submitted: 30 September 2020
Published: 02 March 2021

Summary

Mathematical models expressed in the form of discretized equations play an important role in many scientific disciplines. In our experience, few domain scientists have sufficient background in numerical computing (or the time required to acquire such a background) to use many flexible and powerful but complex open source packages, such as FEniCS (Alnæs et al., 2015) and OpenFOAM (The OpenFOAM Foundation Ltd, n.d.). Many user-friendly open source packages, such as FiPy (J. E. Guyer & Warren, 2009), and many commercial packages, such as COMSOL Multiphysics (COMSOL AB, n.d.) and Simcenter STAR-CCM+ (Siemens Digital Industries Software, n.d.), provide limited flexibility in the equations that users can express. Additionally, the use of commercial packages, which by nature do not perform calculations transparently, can hinder reproducibility, which is vital to the scientific process. PyGDH ("pigged") is a Python 2 / Python 3 (Python Software Foundation, 1991–2020) package that is meant to be accessible to scientists who might not be specialists in scientific computing, while approaching the level of flexibility associated with writing dedicated programs tailored to solving specific problems. The PyGDH User's Guide provides detailed instructions for creating numerical models, including a brief introduction to necessary command line and Python (Python Software Foundation, 1991–2020) skills, and discussions of discretization and validation. Note that PyGDH emphasizes flexibility and simplicity over performance, and was not designed for high-performance applications or models describing complex spatial domains.

Statement of need

PyGDH was created to address the following requirements:

- uses a widely understood, easily learned language, Python (Python Software Foundation, 1991–2020), to allow users with limited scientific computing background to conveniently express mathematical problems to be solved using the finite volume method; this is a common and frequently simple but effective discretization approach, for which (Patankar, 1980) provides an excellent introduction
- to support flexibility, extensibility, and reuse, retains full access to an underlying general-purpose, object-oriented language with broad adoption, Python (Python Software Foundation, 1991–2020), and its wealth of libraries
- allows users to directly describe arbitrary discretized equations (that might be nonlinear and/or time-dependent) to be solved using nonlinear solvers, with convenient representations that are similar to standard mathematical representations but which are in fact executable numerical code used directly by the solver
- requires minimal numerical "bookkeeping" work by users
- allows convenient reuse and postprocessing of calculation results
- allows specification of problems on zero-, one-, and simple two-dimensional spatial domains

- has few strict dependencies, specifically Python (Python Software Foundation, 1991–2020), NumPy (Oliphant, 2006),(Van Der Walt et al., 2011), and SciPy (Virtanen et al., 2020), to simplify installation for novice users
- open source to ensure that all calculations are transparent and to encourage dissemination, verification, and reuse of numerical models; PyGDH was released under the OSI-approved BSD-3-Clause-LBNL license.

We anticipate that PyGDH will be useful to students learning about scientific computing and researchers who want to explore the behavior of potentially complicated equations on simple spatial domains. As an example of the capabilities of this package, please see the work for which PyGDH was originally created (Higa & Srinivasan, 2015).

Acknowledgments

This work was supported by the Assistant Secretary for Energy Efficiency and Renewable Energy, Advanced Manufacturing Office and Office of Vehicle Technologies of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

We thank the developers of the Python programming language (Python Software Foundation, 1991–2020) and its standard libraries, the developers of the NumPy (Oliphant, 2006),(Van Der Walt et al., 2011), SciPy (Virtanen et al., 2020), h5py (Andrew Collette and contributors, 2014), and HDF5 (The HDF Group, 1997–2020) libraries (and the libraries on which these are based), the developers of the Sphinx Python Documentation Generator (G. Brandl and the Sphinx team, 2007–2020), Gnuplot (Thomas Williams, Colin Kelley and many others, 1986–2020), matplotlib (Hunter, 2007), and the Cython compiler (Behnel et al., 2011), as well as developers throughout the Open Source community who have made this work possible.

We hope that PyGDH will encourage scientific openness and strengthen public confidence in the work of the scientific community by providing an open, simple, and transparent problem-solving environment in which researchers can describe mathematical models in compact forms that are easily disseminated. We thank Greg Wilson of Software Carpentry for promoting openness in software development within the scientific community and providing inspiration to release this work as open-source software.

We also thank Ashlea Patterson, Sunil Mair, Anne Suzuki, Fabiola Lopez, Andrew Veenstra, Tiffany Ho and Fiona Stewart for improving the user experience by providing valuable feedback on early versions of the PyGDH documentation and software.

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). https://doi.org/10.11588/ans.2015.100.20553

Andrew Collette and contributors. (2014). *HDF5 for Python*. https://www.h5py.org/

Behnel, S., Bradshaw, R., Citro, C., Dalcin, L., Seljebotn, D. S., & Smith, K. (2011). Cython: The Best of Both Worlds. *Computing in Science Engineering*, *13*(2), 31–39. https://doi.org/10.1109/MCSE.2010.118

COMSOL AB. (n.d.). *Comsol Multiphysics ®*. http://www.comsol.com

G. Brandl and the Sphinx team. (2007–2020). *Sphinx Python Documentation Generator*. https://www.sphinx-doc.org/en/master/

Higa, K., & Srinivasan, V. (2015). Stress and Strain in Silicon Electrode Models. *J. Electrochem. Soc.*, *162*(6), A111–A1122. https://doi.org/10.1149/2.0091507jes

Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

J. E. Guyer, D. W., & Warren, J. A. (2009). FiPy: Partial Differential Equations with Python. *Computing in Science & Engineering*, *11(3)*, 6–15. https://doi.org/10.1109/MCSE.2009.52

Oliphant, T. E. (2006). *A Guide to NumPy* (Vol. 1). Trelgol Publishing USA.

Patankar, S. (1980). *Numerical Heat Transfer and Fluid Flow*. CRC Press.

Python Software Foundation. (1991–2020). *Python*. http://www.python.org

Siemens Digital Industries Software. (n.d.). *Simcenter STAR-CCM+*. https://www.plm.automation.siemens.com/global/en/products/simcenter/STAR-CCM.html

The HDF Group. (1997–2020). *Hierarchical Data Format, version 5*. https://www.hdfgroup.org/HDF5/

The OpenFOAM Foundation Ltd. (n.d.). *OpenFOAM*. https://openfoam.org/

Thomas Williams, Colin Kelley and many others. (1986–2020). *Gnuplot*. http://www.gnuplot.info/

Van Der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). The NumPy Array: A Structure for Efficient Numerical Computation. *Computing in Science & Engineering*, *13*(2), 22. https://doi.org/10.1109/mcse.2011.37

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … SciPy 1.0 Contributors. (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*, *17*, 261–272. https://doi.org/10.1038/s41592-019-0686-2