The Journal of Open Source Education

pylj: A teaching tool for classical atomistic simulation

Andrew R. McCluskey[1, 2], **Benjamin J. Morgan**[1], **Karen J. Edler**[1], **and Stephen C. Parker**[1]

1 Department of Chemistry, University of Bath **2** Diamond Light Source

DOI: 10.21105/jose.00019

Software

- Review
- Repository
- Archive

Submitted: 24 June 2018
Published: 06 July 2018

Summary

pylj is an educational software designed to introduce students to classical simulation using a Lennard-Jones potential model.("On the Determination of Molecular Fields. II. From the Equation of State of a Gas" 1924) Currently, pylj is capable of performing both molecular dynamics (NVE and NVT ensembles) and Monte-Carlo (NVT ensemble) simulations, for a two dimensional argon gas system. Written in Python (with Cythonic pairwise functions), visualisation occurs through a Jupyter notebook,(Kluyver et al. 2016) making use of the matplotlib library (an example of the Interactions visualisation environment is shown below).(Hunter 2007) This allows for easy deployment within a computational laboratory setting and immediate student interaction without an explaination of the command line, as is the case in other open-source molecular dynamics packages.(Berendsen, Spoel, and Drunen 1995)(Plimpton 1995)(Smith, Yong, and Rodger 2002) Example notebooks that show how pylj can be used to simulate both molecular dynamics and Monte-Carlo are available from the GitHub repository, although a large variety of possible applications are possible.

JOSE
The Journal of Open Source Education

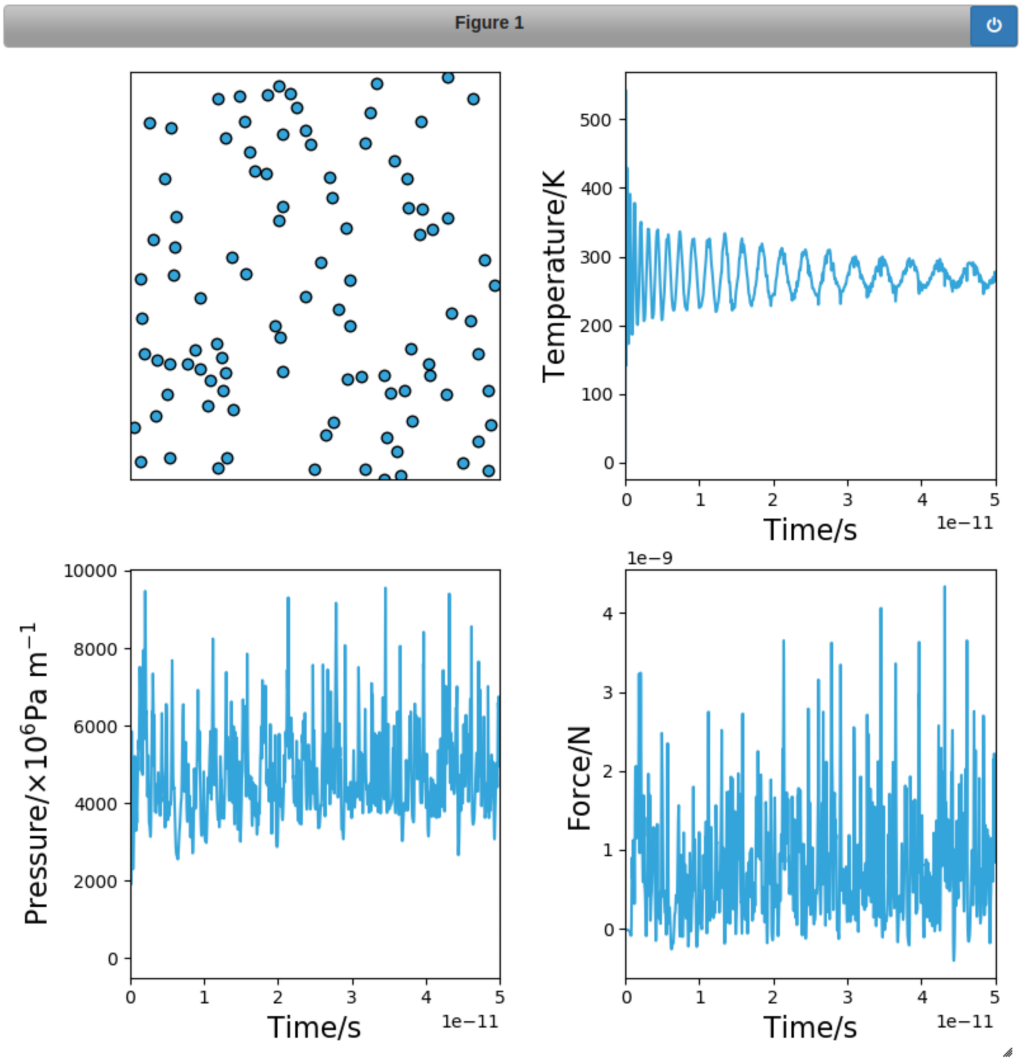

Figure 1. An example of the Interactions sampling class during a molecular dynamics simulation. (McCluskey 2018)

The `md` module currently allows a Velocity-Verlet (Frenkel and Smit 2002b) NVE simulation, which can be adapted to NVT by allowing the thermostating via velocity rescaling. The `mc` module allows a typical NVT Metropolis Monte-Carlo simulation(Frenkel and Smit 2002a), however there is scope for future development of VT, Grand-Canonical simulations.

The latest release of the software can be installed via pip, or from source at https://github.com/arm61/pylj. Full documentation can be found at https://pylj.readthedocs.io.

Statement of Need

As computational science becomes more and more important in the training of undergraduate chemistry and physics students, there is a larger need for easy to use, open-source, sustainable tools that enable student interaction with classical simulation. pylj allows this by offering a simple library of functions from which students can build their own molecular dynamics or Monte-Carlo simulation and easy visualisation, leveraging matplotlib and the Jupyter notebook framework. In addition to use in the introduction of the simulation itself, pylj can also be used to discuss material properties showing how they

may be probed using simulation. A few examples of how pylj could be used in a teaching laboratory can be found in the examples directory. These include an introduction to each of molecular dynamics and Monte Carlo simulations algorithms as well as an example of how pylj could be used to investigate the ideal gas law and the deviation from it at high particle densities. pylj enables easy student interaction with these commonplace simulation methods in a fast and open-source way, while offering educators tools to build custom visualisation environments allowing attention to be drawn to areas of specific focus for a given application.

Acknowledgements

A.R.M. is grateful to the university of Bath and Diamond Light Source for co-funding a studentship (Studentship Number STU0149). B.J.M. acknowledges support from the Royal Society (Grant No. UF130329).

References

Berendsen, H.J.C., D. van der Spoel, and R. van Drunen. 1995. "GROMACS: A Message-Passing Parallel Molecular Dynamics Implementation." *Computer Physics Communications* 91 (1):43–56. https://doi.org/https://doi.org/10.1016/0010-4655(95)00042-E.

Frenkel, Daan, and Berend Smit. 2002a. "Chapter 3 - Monte Carlo Simulations." In *Understanding Molecular Simulation (Second Edition)*, edited by Daan Frenkel and Berend Smit, Second Edition, 23–61. San Diego: Academic Press. https://doi.org/10.1016/B978-012267351-1/50005-5.

———. 2002b. "Chapter 4 - Molecular Dynamics Simulations." In *Understanding Molecular Simulation (Second Edition)*, edited by Daan Frenkel and Berend Smit, Second Edition, 63–107. San Diego: Academic Press. https://doi.org/10.1016/B978-012267351-1/50006-7.

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

Kluyver, Thomas, Benjamin Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, et al. 2016. "Jupyter Notebooks – a Publishing Format for Reproducible Computational Workflows." Edited by F. Loizides and B. Schmidt. IOS Press. https://doi.org/10.3233/978-1-61499-649-1-87.

McCluskey, Andrew. 2018. "fig1," June. https://doi.org/10.6084/m9.figshare.6571583.v1.

"On the Determination of Molecular Fields. II. From the Equation of State of a Gas." 1924. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 106 (738). The Royal Society:463–77. https://doi.org/10.1098/rspa.1924.0082.

Plimpton, Steve. 1995. "Fast Parallel Algorithms for Short-Range Molecular Dynamics." *Journal of Computational Physics* 117 (1):1–19. https://doi.org/https://doi.org/10.1006/jcph.1995.1039.

Smith, W., C.W. Yong, and P.M. Rodger. 2002. "DL_POLY: Application to Molecular Simulation." *Molecular Simulation* 28 (5):385–471. https://doi.org/10.1080/08927020290018769.