




# pylj: A teaching tool for classical atomistic simulation

Andrew R. McCluskey<sup>1, 2</sup>, Benjamin J. Morgan<sup>1</sup>, Karen J. Edler<sup>1</sup>, and Stephen C. Parker<sup>1</sup>

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

DOI: [10.21105/jose.00019](https://doi.org/10.21105/jose.00019)

## Software

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

Submitted: 24 June 2018

Published: 15 July 2018

## License

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

## Summary

pylj is an educational software to introduce students to classical atomistic simulation using a Lennard-Jones potential model.(Jones 1924) pylv is written in Python (using Cython for pairwise interactions) and uses Jupyter notebooks (Kluyver et al. 2016) and Matplotlib for visualisation (see example below).(Hunter 2007) It can be easily deployed in a computer laboratory, and students interact with it without needing to use the command line, as would be the case for other molecular dynamics packages like Gromacs, LAMMPS, or DLPOLY.(Berendsen, Spoel, and Drunen 1995)(Plimpton 1995)(Smith, Yong, and Rodger 2002) We provide example notebooks in the repository, showing how to use pylv to simulated a 2D gas system using either molecular dynamics or Monte-Carlo methods. A variety of other applications are possible.

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 atomistic simulation. pylv offers 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. As well as using pylv to introduce students to classical atomistic simulation itself, this software can also be used to discuss material properties showing how they may be probed using simulation. pylv enables easy student interaction with 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.

## Usage

pylj can be applied in teaching through a variety of methods including lectures or small group tutorials. However, we believe that the most common use of pylv will be in a teaching laboratory setting, with this in mind we suggest two main areas that could make use of pylv:

1. teaching about simulation algorithms, the mechanics of simulation, and allowing students to build a simulation themselves,

2. showing how simulation can be used to study physical attributes of chemical systems, consolidating learning typically introduced during lectures.

Some examples of these can be found in the [examples](#) directory, in particular the [molecular dynamics](#) and [Monte-Carlo](#) examples show how pylj may be used in the teaching of molecular simulations, while the [ideal gas law](#) notebook is exemplary of how pylj can be used to present, in an interactive fashion, the deviation from the ideal gas law that is observed following the gas-to-liquid transition. We envision that pylj is applicable to be used at all levels of undergraduate and postgraduate chemistry and physics course, particularly in computational chemistry and statistical mechanics courses

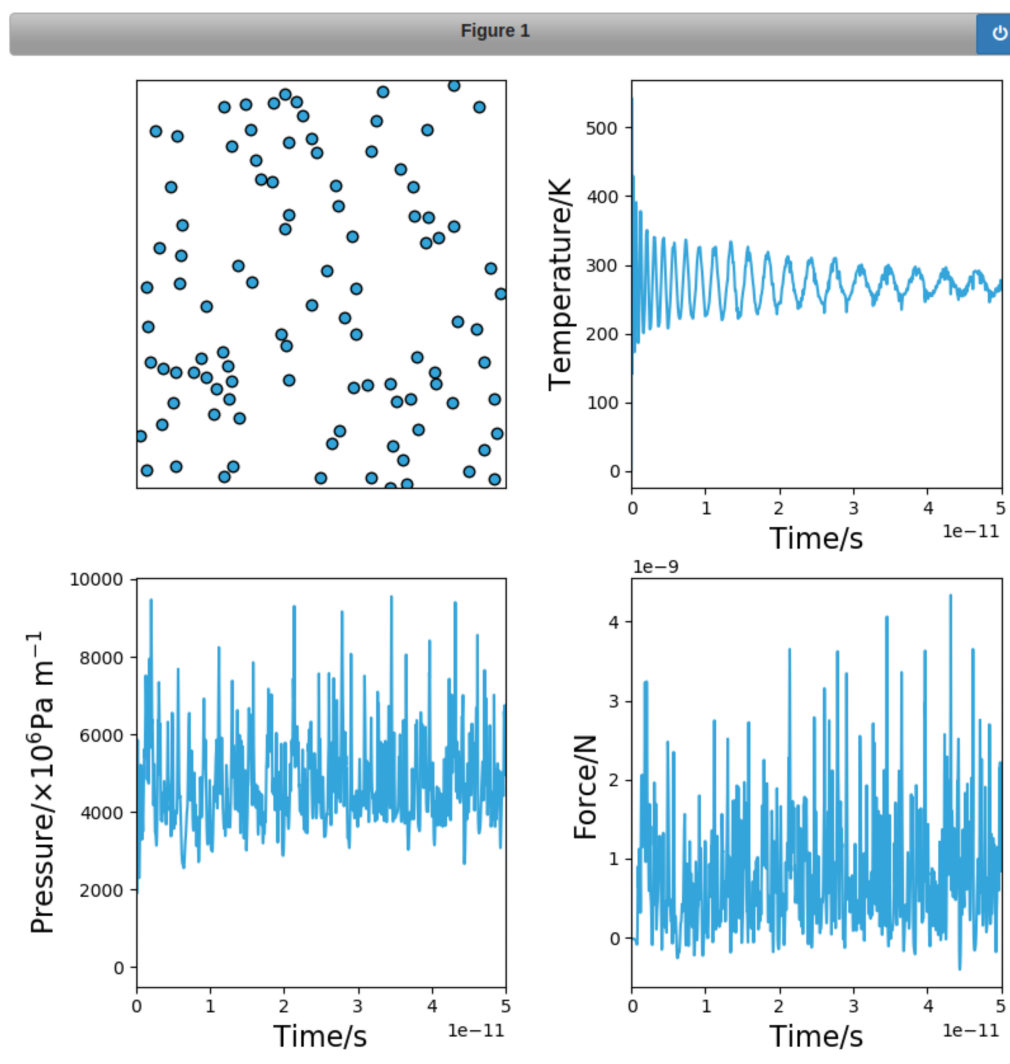


Figure 1. An example of the Interactions sampling class during a molecular dynamics simulation. (McCluskey 2018)

## 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](https://doi.org/https://doi.org/10.1016/0010-4655(95)00042-E).
- 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>.
- Jones, J. E. 1924. "On the Determination of Molecular Fields. II. From the Equation of State of a Gas." *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>.
- 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>.
- 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>.