

Introducing classical molecular dynamics simulation to users of scattering

Andrew R. McCluskey,^{a,b*} Adam R. Symington,^a James Grant,^c Benjamin J. Morgan,^a Stephen C. Parker^a and Karen J. Edler^{a*}

^aDepartment of Chemistry, University of Bath, Claverton Down, Bath, BA2 7AY UK, ^bDiamond Light Source, Diamond House, Rutherford Appleton Laboratory, Harwell Oxford, OX11 0DE, UK, and ^cDepartment of Chemistry, University of Bath, Claverton Down, Bath, BA2 7AY UK. Correspondence e-mail: a.r.mccluskey@bath.ac.uk, k.edler@bath.ac.uk

Classical molecular dynamics simulations are becoming a popular technique for the multi-modal analysis of scattering techniques; such as small angle scattering and diffraction. However, few users of these techniques have formalised training in these methodologies, resulting in frequent use of molecular dynamics simulations as a black box technique. This work discusses an open educational resource designed to introduce classical molecular dynamics to users of scattering, describing possible sources of error in the method. Furthermore, we cover some of the methods that can be used to enable simulation techniques to facilitate in the analysis of scattering data.

© 0000 International Union of Crystallography
Printed in Singapore – all rights reserved

1. Introduction

The use of molecular dynamics simulations to aid in the analysis of experimental data; particular from small angle scattering and diffraction, has grown significantly over the past ten years (Pan *et al.*, 2012; Boldon *et al.*, 2015; Hub, 2018; Ivanović *et al.*, 2018; East *et al.*, 2016; Wall *et al.*, 2014; Wall, 2018; Satoh *et al.*, 2015). Figure 1 shows the growth in the percentage of small angle scattering (SAS) publications that mention molecular dynamics. It can be seen that there has been a linear growth in such publications, with more than 20% of all SAS publications mentioning molecular dynamics as of 2018. This alone is a clear indication of the importance simulation techniques are having on the field of scattering and diffraction.

Usually users of small angle scattering and diffraction techniques have a background in experimental science, with little to no formalised training in computational modelling. This is particularly problematic, as it leads to some users treating molecular simulation as a black-box, using tools such as WAXSiS or SASSIE (Chen & Hub, 2014; Knight & Hub, 2015; Perkins *et al.*, 2016). Use of molecular simulation without a solid understanding of the fundamentals can lead to the inclusion of severe systematic errors.

Recently, there has been a growth in courses that introduce the background of molecular simulation alongside the use of the packages mentioned above. For example, the annual ISIS Neutron Training Course now includes an “Introduction to Molecular Dynamics for Neutron Scattering” course, which details the fundamentals of classical molecular dynamics simulation, before detailing some previous applications in neutron science and allowing the student to gain a familiarity with the SASSIE software. However, courses such as the ISIS Neutron Training Course are limited both in numbers and location meaning not all users of molecular simulation based software packages are able to easily access them.

The utility of learning options is also limited by availability, in order to reduce that need there has been a movement towards technology enhanced open educational resources (OER). The use of the Jupyter Notebook framework (Kluyver *et al.*, 2016) has facilitated this by allowing direct student interaction with the learning resources using, often but not always, the Python programming language.

The work herein describes an online, open-source, interactive module to introduce members from the scattering and diffraction communities to molecular dynamics simulations. This course follows six lessons, covering classical methods and introducing molecular dynamics methods. Then the Python library pylj (McCluskey *et al.*, 2018) is used to introduce the interaction between molecular dynamics simulation and scattering/diffraction in the context on the Debye equation (Debye, 1915). We will discuss implementation and detail how the student, regardless of career position, can get the most from the module.

2. Assumed prior knowledge

The interaction between simulation and scattering course makes use of the Python programming language to give interactive examples for the mathematical and algorithmic content. Therefore some knowledge, or willingness to learn, Python is desired. However, we have aimed to write the course in such a fashion that in depth Python knowledge is not required. Some knowledge of undergraduate chemistry or physics is necessary to fully understand the nature of the classical interaction potentials. Furthermore, the level of mathematics required for an undergraduate science degree is assumed.

3. Module construction

The module is currently hosted online at https://arm61.github.io/sim_and_scatt. A Jupyter Notebook based

course building system developed at the University of Bath was used to allow the course to be written as a series of Jupyter Notebooks, which are then compiled into the webpages. This allowed Python code blocks to be built into the lessons to show algorithmic details in practice.

The expectation is that the user would use the online material alongside a Jupyter Notebook running locally (installation of this is detailed in the introduction to the course). This would allow the user to copy the Python code from the webpages, run it locally, and alter it as they see fit.

The module is licensed under CC-BY-SA-4.0, and is therefore a contribution to the growing library of open educational resources. The open-source nature of this license means that particularly interested parties are able to easily use the module to enhance their own educational platform. Additionally, experts in the field may wish to contribute to it, or fork their own version that may grow independently. The source code is available at https://github.com/arm61/sim_and_scatt.

4. Module outline

The module follows a simple outline, with code blocks that gradually build up the students understanding of the concepts.

4.1. Getting started

The first page, nominally Lesson 0, introduces the course and outlines the content. It also gives directions on how to install the necessary software, namely Anaconda Python, and the pylj package (McCluskey *et al.*, 2018), and launch the Jupyter Notebook interface (Kluyver *et al.*, 2016). It is important that this is introduced early in the module such that students can immediately be actively involved in the interactive experience of the lessons.

4.2. Classical methods

Once the students computer is able to run the necessary software, concepts of classical simulation methods are introduced. These include the nature of potential models and show some examples such as the Lennard-Jones and Buckingham potentials (Lennard-Jones, 1924; Buckingham, 1938). The course gives a brief description of how a potential model is determined, including mention to the parameterisation from higher accuracy quantum mechanical calculations. The existence of off-the-shelf potential models is mentioned alongside a discussion of the problems associated with these as they also may require system specific optimisation. Finally, we mention mixing rules for off-the-shelf potential models, and again discuss possible problems that a user may encounter related to system specificity.

4.3. Molecular dynamics

Once the basics of classical methods have been introduced, the student is presented with molecular dynamics. This is shown by gradually building up a one-dimension molecular dynamics simulation using the Velocity-Verlet algorithm (Swope *et al.*, 1982) and the Lennard-Jones potential model (Lennard-Jones, 1924). This begins by detailing the requirement to find

the forces on the particles, before introducing how the acceleration is determined for each particle. Finally, the different aspects of the determination of the particle positions and velocities using the Velocity-Verlet algorithm are outlined.

4.4. pylj and the interaction with scattering

The final aspect of the module makes use of the Python package pylj (McCluskey *et al.*, 2018). This is an open-source package designed to introduce students to classical atomistic simulation. First, the student is given a working pylj molecular dynamics simulation to run on locally, then they are shown how to use the custom plot functionality of pylj to present information about the simulation. In the final lesson, the Debye equation is introduced and the students are shown how to implement it within pylj. Using pylj, the students may then observe the effect of temperature on both the structure of the argon particles, in addition to the scattering profile that is generated from the Debye equation. Finally, we mention the algorithms that may be used to improve the efficiency of the calculation of the scattering profile, such as the Fibonacci sequence or the Golden Vectors method (Svergun, 1994; Watson & Curtis, 2013).

5. Future outlook

In future, we hope that the open-source nature of the module will allow interested parties to use and build upon it. Furthermore, we welcome student input to improve implementation and pedagogical aspects of the module. We hope to use this module within teaching at courses such as the ISIS Neutron Training Course and the Diamond SAS workshop, from which feedback may be obtained directly.

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

- Boldon, L., Laliberte, F. & Liu, L. (2015). *Nano Rev.* **6**, 25661.
- Buckingham, R. A. (1938). *Proc. R. Soc. Lond. A*, **168**, 264–283.
- Chen, P.-C. & Hub, J. S. (2014). *Biophys. J.* **107**, 435–447.
- Debye, P. J. W. (1915). *Ann. Phys.* **351**, 809–823.
- East, A., Mechaly, A. E., Huysmans, G. H. M., Bernarde, C., Tello-Manigne, D., Nadeau, N., Pugsley, A. P., Buschiazzi, A., Alzari, P. M., Bond, P. J. & Francetic, O. (2016). *Structure*, **24**, 92–104.
- Hub, J. (2018). *Curr. Opin. Struct. Biol.* **49**, 18–36.
- Ivanović, M. T., Bruetzel, L. K., Lipfert, J. & Hub, J. S. (2018). *Ange. Chem. Int. Ed.* **57**, 5635–5639.
- Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S. & Willing, C. (2016). In *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, edited by F. Loizides & B. Schmidt, pp. 87–90. IOS Press.
- Knight, C. J. & Hub, J. S. (2015). *Nucleic Acids Res.* **43**, W225–W230.
- Lennard-Jones, J. E. (1924). *Proc. R. Soc. Lond. A*, **106**, 463–477.
- McCluskey, A. R., Morgan, B. J., Edler, K. J. & Parker, S. C. (2018). *JOSE*, **1**, 19.
- Pan, J., A. Heberle, F., Tristram-Nagle, S., Szymanski, M., Koepfinger, M., Katsaras, J. & Kučerka, N. (2012). *BBA - Biomembranes*, **1818**, 2135–2148.
- Perkins, S. J., Wright, D. W., Zhang, H., Brookes, E., Chen, J., Irving, T. C., Krueger, S., Barlow, D. J., Edler, K. J., Scott, D. J., Terrill, N. J., King, S. M., Butler, P. D. & Curtis, J. E. (2016). *J. Appl. Crystallogr.* **49**, 1861–1875.

Satoh, M., Saburi, H., Tanaka, T., Matsuura, Y., Naitow, H., Shimozono, R., Yamamoto, N., Inoue, H., Nakamura, N., Yoshizawa, Y., Aoki, T., Tanimura, R. & Kunishima, N. (2015). *FEBS Open Bio*, **5**, 557–570.

Svergun, D. I. (1994). *Acta. Cryst.* **A50**, 391–402.

Swope, W. C., Andersen, H. C., Berens, P. H. & Wilson, K. R. (1982). *J. Chem. Phys.* **76**, 648 (Appendix).

Wall, M. E. (2018). *IUCrJ*, **5**, 172–181.

Wall, M. E., van Benschoten, A. H., Sauter, N. K., Adams, P. D., Fraser, J. S. & Terwilliger, T. C. (2014). *PNAS*, **111**, 17887–17892.

Watson, M. C. & Curtis, J. E. (2013). *J. Appl. Crystallogr.* **46**, 1171–1177.

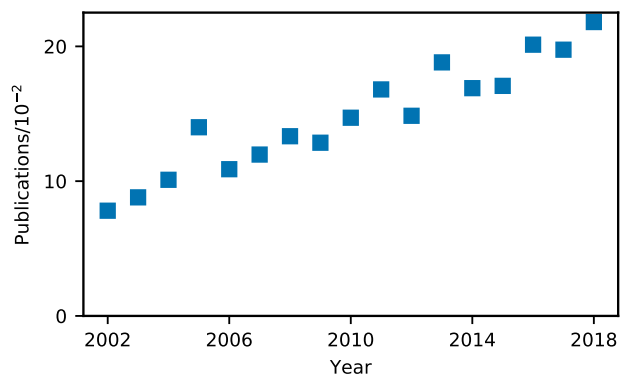


Figure 1

The annual growth of the percentage of publication that mention “small angle scattering” which also mention “molecular dynamics. Determined from the number of results from a Google Scholar search.