

Introducing classical molecular dynamics simulation to users of scattering

A. R. McCluskey,^{1,2,*} A. R. Symington,¹ J. Grant,³ B. J. Morgan,¹ S. C. Parker,¹ and K. J. Edler^{1,†}

¹*Department of Chemistry, University of Bath, Claverton Down, Bath, BA2 7AY, UK*

²*Diamond Light Source, Harwell Campus, Didcot, OX11 0DE, UK*

³*Computing Services, University of Bath, Claverton Down, Bath, BA2 7AY, UK*

(Dated: November 19, 2018)

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.

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 [1–8]. Figure 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.

Normally, the users of scattering and diffraction techniques have a background in experimental science, with little or no formalised training in computational modelling techniques. This may be problematic, as it can lead to the use of molecular simulation as a black-box, with little understanding or consideration for the underlying methodologies. The use of molecular simulation in this fashion can quickly lead to the inclusion of severe, systematic errors. This has led to the development of tools, such as WAXSiS or SASSIE [9–11], that present easy-to-use interfaces designed to reduce the risk of these errors.

In addition to the development of software packages, another method to further limit systematic errors, are lectures or courses presenting and introduction to molecular simulation for scattering and diffraction users. For example, the annual ISIS Neutron Training Course now included an “Introduction to Molecular Dynamics for Neutron Scattering” module. This details the fundamentals of classical molecular dynamics simulation, before presenting some previous applications of these methods in neutron science and allowing the students to gain a familiarity with the SASSIE package. While such courses and modules are an important aspect of training, they are limited both in numbers of students that may attend and location, meaning that not scattering and diffraction users may easily access them.

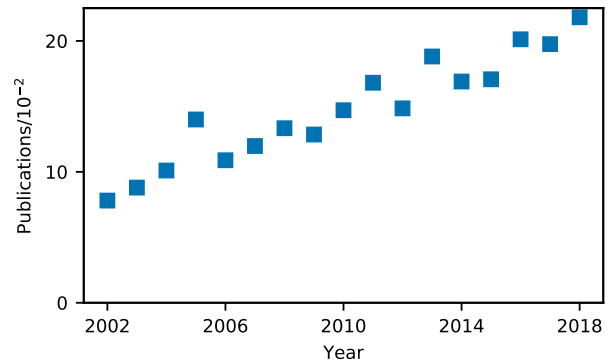


FIG. 1. The annual growth of the percentage of publication that mention “small angle scattering” which also mention “molecular dynamics. Determined from the number of results from a Google Scholar search.

There has been a recent movement within the scientific and engineering communities towards technology-enhanced open educational resources (OERs). These are courses, lectures, or learning modules that are made free available online for use by anyone. This allows the permission for others to engage in the “5R activities”: Retain, Reuse, Revise, Remix, and Redistribute [12]. The act of publishing an OER increases the reach of a particular resource as it allows others to pick it up, modify it, and use the modified version in their own teaching. The availability of the Jupyter Notebook framework [13] has facilitated the availability of OERs by improving the ability for students to interact directly with the resource using, often but not always, the Python programming language [14].

Herein, we present an online, open-source, interactive module aimed to introduce members of the scattering and diffraction community to molecular dynamics simulation. This course follows six lessons, covering classical methods, introducing molecular dynamics, and showing how this method may interact with scattering in a multi-modal fashion. We leverage the open-source Python li-

brary pylj [15] to give a visual, and programmatic representation of the interaction between these two techniques. In this paper, we will discuss the implementation of the module, and detail how the student (in this work *the student* refers to any student of the module regardless of career position) may get the most from the module.

ASSUMED PRIOR KNOWLEDGE

The OER, entitled “*The interaction between simulation and scattering*” makes use of the Python programming language, this allows for interactive examples of the mathematical and algorithm content to be included. A result of this is that, to be able to fully utilise the course, some knowledge, of willingness to learn, Python is required. However, we have attempted to develop the course in such a fashion that an in depth knowledge of Python is not required. It is anticipated that the users of this course would have some familiarity with undergraduate chemistry or physics, and the associated level of mathematics. This is particularly important when considering the nature, and chemical rationale, of the classical interaction potentials.

MODULE CONSTRUCTION

The module is currently available online, at https://arm61.github.io/sim_and_scat. These webpages were written as a series of Jupyter Notebooks and compiled using the build-system developed within the University of Bath. The use of this system allows for Python code blocks to be easily build into the lessons to give algorithmic details and facilitate interactivity. It is expected that the student would use the online material alongside a locally running Jupyter Notebook (the installation of which is detail in the module introduction). Interacting with the module in this fashion will allow the student to copy the Python code from the web interface, run it locally, and alter the code as they see fit.

The module is provided under a CC-BY-SA-4.0 license, and builds on the growing library of open educational resources. The open-source nature of this license means that anyone may use the material to enhance their own educational platform, and experts in the field may contribute to improve the module. The source code for the module is available at https://github.com/arm61/sim_and_scat.

MODULE OUTLINE

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

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 [15], and launch the Jupyter Notebook interface [13]. 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.

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 [16, 17]. 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.

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 [18] and the Lennard-Jones potential model [16]. 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.

pylj and the interaction with scattering

The final aspect of the module makes use of the Python package pylj [15]. This is a 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 [19, 20].

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).

* a.r.mccluskey@bath.ac.uk

† k.edler@bath.ac.uk

- [1] J. Pan, F. A. Heberle, S. Tristram-Nagle, M. Szymanski, M. Koepfinger, J. Katsaras, and N. Kučerka, *BBA - Biomembranes* **1818**, 2135 (2012).
- [2] L. Boldon, F. Laliberte, and L. Liu, *Nano Rev.* **6**, 25661 (2015).
- [3] J. Hub, *Curr. Opin. Struct. Biol.* **49**, 18 (2018).
- [4] M. T. Ivanović, L. K. Bruetzel, J. Lipfert, and J. S. Hub, *Ange. Chem. Int. Ed.* **57**, 5635 (2018).
- [5] A. East, A. E. Mechaly, G. H. M. Huysmans, C. Bernarde, D. Tello-Manigne, N. Nadeau, A. P. Pugsley, A. Buschiazzi, P. M. Alzari, P. J. Bond, and O. Francetic, *Structure* **24**, 92 (2016).
- [6] M. E. Wall, A. H. van Benschoten, N. K. Sauter, P. D. Adams, J. S. Fraser, and T. C. Terwilliger, *PNAS* **111**, 17887 (2014).
- [7] M. E. Wall, *IUCrJ* **5**, 172 (2018).
- [8] M. Satoh, H. Saburi, T. Tanaka, Y. Matsuura, H. Naitow, R. Shimozone, N. Yamamoto, H. Inoue, N. Nakamura, Y. Yoshizawa, T. Aoki, R. Tanimura, and N. Kunishima, *FEBS Open Bio* **5**, 557 (2015).
- [9] P.-C. Chen and J. S. Hub, *Biophys. J.* **107**, 435 (2014).
- [10] C. J. Knight and J. S. Hub, *Nucleic Acids Res.* **43**, W225 (2015).
- [11] S. J. Perkins, D. W. Wright, H. Zhang, E. Brookes, J. Chen, T. C. Irving, S. Krueger, D. J. Barlow, K. J. Edler, D. J. Scott, N. J. Terrill, S. M. King, P. D. Butler, and J. E. Curtis, *J. Appl. Crystallogr.* **49**, 1861 (2016).
- [12] D. Wiley, “Open content - definition,” <http://opencontent.org/definition/> (2018), this material was created by David Wiley and published freely under a Creative Commons Attribution 4.0 license at <http://opencontent.org/definition/>. Accessed: 2018-11-19.
- [13] T. Kluyver, B. Ragan-Kelley, F. Pérez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay, P. Ivanov, D. Avila, S. Abdalla, and C. Willing, in *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, edited by F. Loizides and B. Schmidt (IOS Press, 2016) pp. 87–90.
- [14] L. A. Barba, A. Wickenheiser, and R. Watkins, “CyberTraining: DSEThe Code Maker: Computational Thinking for Engineers with Interactive, Contextual Learning,” <https://doi.org/10.6084/m9.figshare.5662051.v1> (2017).
- [15] A. R. McCluskey, B. J. Morgan, K. J. Edler, and S. C. Parker, *JOSE* **1**, 19 (2018).
- [16] J. E. Lennard-Jones, *Proc. R. Soc. Lond. A* **106**, 463 (1924).
- [17] R. A. Buckingham, *Proc. R. Soc. Lond. A* **168**, 264 (1938).
- [18] W. C. Swope, H. C. Andersen, P. H. Berens, and K. R. Wilson, *J. Chem. Phys.* **76**, 648 (Appendix) (1982).
- [19] D. I. Svergun, *Acta. Cryst.* **A50**, 391 (1994).
- [20] M. C. Watson and J. E. Curtis, *J. Appl. Crystallogr.* **46**, 1171 (2013).