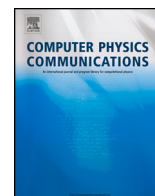




Contents lists available at ScienceDirect

# Computer Physics Communications

journal homepage: [www.elsevier.com/locate/cpc](http://www.elsevier.com/locate/cpc)

Computer Programs in Physics



## TTCF4LAMMPS: A toolkit for simulation of the non-equilibrium behaviour of molecular fluids at experimentally accessible shear rates <sup>☆,☆☆</sup>

Luca Maffioli <sup>a,\*</sup>, James P. Ewen <sup>b</sup>, Edward R. Smith <sup>c</sup>, Sreeba Varghese <sup>e</sup>, Peter J. Davis <sup>d</sup>, Daniele Dini <sup>b</sup>, B.D. Todd <sup>a</sup>

<sup>a</sup> Department of Mathematics, School of Science, Computing and Engineering Technologies, Swinburne University of Technology, P.O. Box 218, Hawthorn, 3122, Victoria, Australia

<sup>b</sup> Department of Mechanical Engineering, Imperial College London, South Kensington Campus, London, SW7 2AZ, London, United Kingdom

<sup>c</sup> Mechanical and Aerospace Engineering, Brunel University London, Kingston Lane, Uxbridge, UB8 3PH, London, United Kingdom

<sup>d</sup> School of Science, RMIT University, GPO Box 2476, Melbourne, 3001, Victoria, Australia

<sup>e</sup> Sorbonne Université, CNRS, Physicochimie des Électrolytes et Nanosystèmes Interfaciaux, UMR 8234 PHENIX, F-75005, Paris, France

### ARTICLE INFO

#### Keywords:

Shear rate

Viscosity

Shear stress

Slip

Transient time correlation function (TTCF)

Nonequilibrium molecular dynamics (NEMD)

Friction

Tribology

### ABSTRACT

We present TTCF4LAMMPS, a toolkit for performing non-equilibrium molecular dynamics (NEMD) simulations to study the fluid behaviour at low shear rates using the LAMMPS software. By combining direct NEMD simulations and the transient-time correlation function (TTCF) technique, we study the fluid response to shear rates spanning 15 orders of magnitude. We present two examples for simple monatomic systems: one consisting of a bulk liquid and another with a liquid layer confined between two solid walls. The small bulk system is suitable for testing on personal computers, while the larger confined system requires high-performance computing (HPC) resources. We demonstrate that the TTCF formalism can successfully detect the system response for arbitrarily weak external fields. We provide a brief mathematical explanation for this feature. Although we showcase the method for simple monatomic systems, TTCF can be readily extended to study more complex molecular fluids. Moreover, in addition to shear flows, the method can be extended to investigate elongational or mixed flows as well as thermal or electric fields. The high computational cost needed for the method is offset by the two following benefits: i) the cost is independent of the magnitude of the external field, and ii) the simulations can be made highly efficient on HPC architectures by exploiting the parallel design of the algorithm. We expect the toolkit to be useful for computational researchers striving to study the nonequilibrium behaviour of fluids under experimentally-accessible conditions.

#### Program summary

*Program title:* TTCF4LAMMPS

*CPC Library link to program files:* <https://doi.org/10.17632/hh2rkcxbrf.1>

*Developer's repository link:* <https://github.com/edwardsmith999/TTCF4LAMMPS>

*Licensing provisions:* GNU General Public License 3

*Programming language:* Python 3

*Nature of problem:* Measuring the nonequilibrium behaviour of bulk and confined fluids under experimentally accessible strain rates in non-equilibrium molecular dynamics (NEMD) simulations.

*Solution method:* Creating a Python-based code that utilises the transient-time correlation function method and the LAMMPS software to enable the bulk fluid properties (e.g. viscosity) and confined fluid interfacial properties (e.g. shear stress and slip velocity) to be computed at low shear rates with NEMD.

<sup>☆</sup> The review of this paper was arranged by Prof. W. Jong.

<sup>☆☆</sup> This paper and its associated computer program are available via the Computer Physics Communications homepage on ScienceDirect (<http://www.sciencedirect.com/science/journal/00104655>).

\* Corresponding author.

E-mail address: [lmaffioli@swin.edu.au](mailto:lmaffioli@swin.edu.au) (L. Maffioli).

<https://doi.org/10.1016/j.cpc.2024.109205>

Received 22 December 2023; Received in revised form 3 April 2024; Accepted 9 April 2024

Available online 15 April 2024

0010-4655/© 2024 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

Over the last few decades [1], nonequilibrium molecular dynamics (NEMD) simulations have provided atomic-scale insights into important fluid behaviour under shear such as turbulence [2], cavitation [3], boundary slip [4], and shear thinning [5,6]. One significant limitation of these NEMD simulations is that high shear rates are required to obtain satisfactory signal-to-noise ratios [7,8]. In general, the shear rates required are higher than those that can be applied experimentally by several orders of magnitude, which prevents direct experimental validation of the simulations [5,6]. Equilibrium molecular dynamics methods (EMD) can effectively characterize the transport coefficients in the limit of vanishing external field, both in bulk and confined fluids [9–11]. However, they are valid only in the linear response regime. While simple fluids retain a linear response up to relatively strong driving forces, complex molecular fluids might exhibit a nonlinear behaviour even at weak fields. This problem can be overcome using the transient-time correlation function (TTCF) formalism [12,13], which is a non-linear generalization of the Green-Kubo formula [9,10]. TTCF utilises the time correlation between the so-called dissipation function of the system and the transient response of any arbitrary phase variable after an external field is activated. Its main advantage is that it is *exact*, i.e. valid at any regime. As such, the method forms a bridge between the equilibrium techniques where no external field is applied and direct NEMD where strong fields are required [14]. TTCF has been successfully applied to study the rheology of bulk monatomic fluids [15–18], molecular fluids [19,20], and liquid metals [21,22] at low shear rates. In addition to shear flows, TTCF has been used to study elongational flows [23] and mixed flows [24]. It has also been applied to investigate monatomic [25–27] and molecular fluids [28] confined between sliding solid surfaces. TTCF can also be used to investigate other types of external fields such as electric and colour fields [29–32], of strengths that are closer to those applied experimentally, compared to direct NEMD simulations. Evans et al. [13] derived the theoretical framework for a wider range of nonequilibrium phenomena including heat transfer in inhomogeneous systems and the relaxation from a nonequilibrium thermodynamic state.

Despite their clear benefits over traditional NEMD techniques, TTCF-based NEMD simulations have only been utilised by a handful of research groups worldwide, arguably due to the complexity of the method implementation. Here, we present TTCF4LAMMPS, a toolkit for studying the non-equilibrium behaviour of fluids under weak external fields using the open source Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS) software [33]. The method can be made very efficient on HPC architectures by using the embarrassingly parallel setup. We demonstrate the application of the TTCF method to two example systems consisting of simple monatomic fluids under shear: one bulk and one confined.

## Mathematical framework

Traditional NEMD simulations require the user to monitor the system over a single nonequilibrium trajectory from which the response is measured. A time average is then performed over the data sampled from the steady state. The process can be optimized by running in parallel multiple independent copies of the same system, where each one starts from a different initial condition. Each system is in the same thermodynamic state (temperature, density, etc.), is subjected to the same force and hence undergoes analogous transient toward the nonequilibrium steady state. The time average is then replaced by an ensemble average, where the signal at time  $t$  is the average of the response across all the systems at time  $t$ . The Transient Time Correlation Function (TTCF) technique is instead based on the following result of nonequilibrium statistical mechanics [12]:

$$\langle B(t) \rangle = \langle B(0) \rangle + \int_0^t \langle \Omega(0) B(s) \rangle ds \quad (1)$$

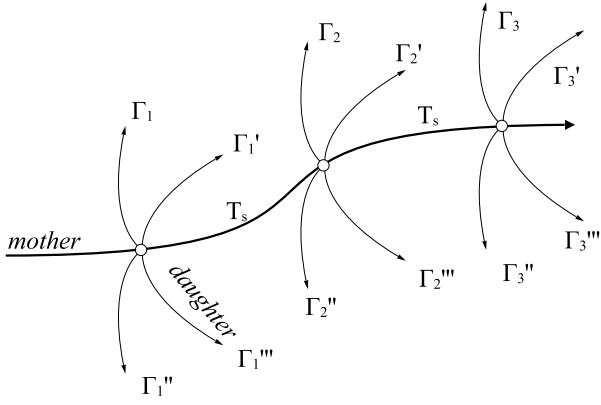
where  $B(t)$  is the arbitrary quantity measured from the system. The formula states an identity between the phase space average (left hand side) of  $B(t)$  and the integral of the time correlation between the same variable  $B(t)$  and the so-called dissipation function  $\Omega$  (right hand side).  $t = 0$  represents the start of the nonequilibrium trajectory, when the external field is switched on and the system is driven out of equilibrium.  $\Omega$  is equal to  $\beta \dot{H}^{ad}$ , with  $\beta = 1/k_B T$  and  $\dot{H}^{ad}$  the adiabatic time derivative of the internal energy, that is, the derivative of the mechanical energy without accounting for any thermostat term.  $k_B$  and  $T$  are the Boltzmann constant and the temperature set by a thermostat. For the application of TTCF, two conditions must be met: first, the initial conditions for the nonequilibrium trajectories must be sampled from the equilibrium probability distribution of the system. The simplest way to get such a sample is to follow the system over an equilibrium *mother* trajectory, from which the microscopic state  $\Gamma(t_i)$  is periodically sampled.  $\Gamma(t_i)$  is the collection of all positions and momenta of the particles at time  $t_i$ , and is used as the initial condition for the  $i$ -th nonequilibrium run, or *daughter* trajectory. To get good statistics, many initial states and corresponding daughter trajectories must be produced. The mother trajectory is hence a simple tool to generate the correct ensemble over which to compute the phase space average. In Eq. (1),  $t = 0$  corresponds to the initial state of the nonequilibrium run, when the system is still at equilibrium and the external force is activated. The response  $B(t)$  is then monitored over each daughter trajectory, correlated with the dissipation function  $\Omega$  at  $t = 0$ , the average is performed at each time step across all the trajectories, and integrated as per Eq. (1). The second requirement is that the system is *mixing*, that is,  $\Omega(0)$  and  $B(t)$  must eventually decorrelate,  $\langle \Omega(0) B(t) \rangle \rightarrow \langle \Omega(0) \rangle \langle B(t) \rangle$  for  $t \rightarrow \infty$ . Since at  $t = 0$  the system is at equilibrium,  $\langle \Omega(0) \rangle = 0$ , and the convergence of the integral is ensured. The mixing requirement prevents the direct application of the TTCF algorithm to systems characterized by long time correlations, which is common, for instance, in the solid state. Since  $\Omega(0)$  is null only on average ( $\langle \Omega(0) \rangle = 0$  but in general  $\Omega(0)_i \neq 0$ ), a finite sample can hardly guarantee  $\langle \Omega(0) \rangle = 0$  and hence the perfect convergence of the integral. The issue can be avoided by generating further initial conditions from each sampled state  $\Gamma_i$  by using the following transformations:

$$\begin{aligned} \Gamma_i &= (\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{p}_x, \mathbf{p}_y, \mathbf{p}_z) \longrightarrow \Gamma'_i = (\mathbf{x}, \mathbf{y}, \mathbf{z}, -\mathbf{p}_x, -\mathbf{p}_y, -\mathbf{p}_z) \\ \Gamma_i &= (\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{p}_x, \mathbf{p}_y, \mathbf{p}_z) \longrightarrow \Gamma''_i = (-\mathbf{x}, \mathbf{y}, \mathbf{z}, -\mathbf{p}_x, \mathbf{p}_y, \mathbf{p}_z) \\ \Gamma_i &= (\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{p}_x, \mathbf{p}_y, \mathbf{p}_z) \longrightarrow \Gamma'''_i = (-\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{p}_x, -\mathbf{p}_y, -\mathbf{p}_z) \end{aligned} \quad (2)$$

where the first mapping inverts the sign of the momentum for the particles. The second mapping is an  $x$ -reflection of the entire system, while the third is a combination of the first and the second. For a Couette flow applied in the  $xy$  plane,  $\Gamma''$  and  $\Gamma'''$  change the sign of  $\Omega(0)$  and hence the average over the four states (original + mappings) is identically null. It is important to note that these mappings, particularly the second and the third, are effective *only* if the system is subjected to a shear in the  $xy$ -plane. Different types of external forces may require other mappings. By generating several starting points from the same state, the mappings are also effective to optimize the sampling process. There are several different choices of transformations: each of the position and momentum component can be mirrored independently, and some authors even employed a permutation of the components [18] (if the system is periodic and has the same size along the  $x$  and  $z$  dimension,  $\mathbf{x}$  and  $\mathbf{z}$  and the corresponding momenta can be swapped), but any combination has to meet the condition  $\langle \Omega(0) \rangle = 0$ , and each mapping must have the same probability of the original state of being sampled from the mother trajectory. A simple alternative to the mapping method consists of replacing the integrand function with

$$\langle \Omega(0) B(s) \rangle - \langle \Omega(0) \rangle \langle B(s) \rangle \quad (3)$$

which automatically includes the correction for finite sample effects. This second expression is redundant if the mappings guarantees  $\langle \Omega(0) \rangle =$



**Fig. 1.** Schematic representation of the TTCF method. From the equilibrium mother trajectories, the microscopic state is sampled every  $N_s$  time units. Four daughter nonequilibrium trajectories depart from the initial state.

0), but was successfully applied in a previous work [26], where the proper transformations were not used. For completeness, we recall the explicit expression for TTCF and traditional NEMD (direct average, or DAV) measurements that will be used in the following:

$$\langle B(t) \rangle_{\text{DAV}} = \frac{1}{N} \sum_i^N B(t)_i$$

$$\langle B(t) \rangle_{\text{TTCF}} = \frac{1}{N} \left[ \sum_i^N B(0)_i + \int_0^t \sum_i^N \Omega(0)_i B(s)_i ds \right] \quad (4)$$

were  $N$  is the total number of nonequilibrium trajectories, including the mappings.

The general scheme of the TTCF algorithm is shown in Fig. 1. The system is followed along an equilibrium mother trajectory. After a period of thermalization, when the system has fully reached the thermodynamic equilibrium, the sampling process starts. Every  $T_s$  time units the microstate  $\Gamma(t)$  is stored, and used as an initial condition for the corresponding daughter trajectories, after being properly modified with the mappings selected. It is important to clarify that the dissipation function should be computed on the daughter trajectory, after the external field has been switched on. The delay  $T_s$  between samples should be long enough to ensure an independent set of initial states. The autocorrelation function of some benchmark quantities can be used to determine the necessary delay. These variables can be the same function  $B$  that will be monitored along the nonequilibrium segments, or  $\Omega(t)B(t)$ . The class of applicability of the TTCF formalism is very broad. Since molecular bonds are conservative forces or ideal constraints, they do not bring any contribution to the dissipation function, which is instead completely determined by the external force driving the system out of equilibrium. Moreover, the implementation of a TTCF algorithm is feasible for several types of external field [13].

## Software methodology

The number of daughter trajectories needed is typically in the order of tens of thousands or even millions. A naive implementation of the transient time correlation function would run LAMMPS for every daughter, storing all required quantities over the trajectory to disk. Each trajectory would then be loaded, the correlation with the initial dissipation function calculated and the totals summed to give the TTCF for that quantity. Hence, three major bottlenecks can be found which significantly impact the usage of the method on local machines: total simulation time (the required time to reach steady state), total memory size of the output (the number of variables considered possibly over a grid of chunks), and number of output files produced (number of daughters). While custom built codes can be designed to minimize the

size and number of output files, e.g. by including the averaging and integration process in the same script, this feature may sometimes be cumbersome to implement with molecular dynamics packages such as LAMMPS.

In this paper, we provide two different TTCF implementations. The first is the naive implementation using Linux shell scripting together with LAMMPS input scripts, designed to be run on an HPC clusters. It is intuitive but generates a large number of output files and requires a large amount of mass storage. The second paradigm is suitable for both local and HPC machines, streamlined using the LAMMPS Python interface. Its LAMMPS script is designed to generate both mother and daughter trajectories in a single run. It does not produce any output files except for the final result. It is more compact and optimized, and can be faster if the simulation itself is particularly quick, and the process of reading and writing on file becomes relatively intensive.

The first scheme works as follows: since the number of nonequilibrium runs required for the TTCF calculation is almost certainly much larger than the number of available cores, the total number of daughter trajectories is split into several independent, single-core runs. Each run is composed of an equilibrium mother trajectory which generates the required number of initial states, after which the corresponding nonequilibrium simulations are performed. Since each run is independent of the others, it can start as soon as a single core is available on the cluster. This scheme is “embarrassingly parallel”, and, as a result, the simulation is extremely fast to run, even if the overall amount of resources needed is large. The pseudocode of the corresponding workflow for a shell script is the script cycles over the required cores to launch

---

### Algorithm 1 SHELL SCRIPT.

---

```

1: for i = 1, ..., Ncores do
2:   select seed SEED(i) from list of random integers
3:   run LAMMPS MOTHER ( SEED(i) )
4:   produce Ndaughters initial states (files)
5:   for n = 1, ..., Ndaughters do
6:     for m = 1, ..., Nmaps do
7:       run LAMMPS DAUGHTER (n,m)
8:       take initial state (file) n, apply mapping m
9:       produce OUTPUT_FILEn,m
10:    end for
11:  end for
12:  perform partial average and TTCF integration
13: end for
14: gather total averages, total TTCF integration, statistics.

```

---

independent runs, as depicted in Algorithm 1. Each mother trajectory is initialized by a random number, accordingly select from a list previously created. The random number is the seed for the random velocities with which the particles are initialized at the start of the run. LAMMPS run then generates  $N_{daughters}$  files which contain the collection of the starting points for the daughter trajectories. The script then cycles over each state  $n$  and each mapping  $m$ . The LAMMPS daughter script reads the file  $n$  and applies the mapping  $k$ , and runs the nonequilibrium simulation from which the response is monitored, and stored in the file indexed as  $n, m$ . At the end of the double loop, the data within each core are averaged, and a first TTCF integration is performed over the partial results. Once the partial results have been produced, a final script cycles across each of them to perform the final average, the final TTCF integration, and to estimate the standard error of the results. The process of writing and reading the initial states should be performed with the LAMMPS command `write_restart` and `read_restart`, which ensure the whole system’s state is transferred from mother to daughter trajectory including the value of the thermostat, if relevant.

The second scheme is designed to maximize the efficiency. Hence, we manage and run the LAMMPS simulation via a Python script. The outputs are not written on file but directly managed by the Python code, which averages the data from all the daughter trajectories and stores

them until the end of the simulation, when the output file is eventually produced.

From a computational perspective, the only difference between mother and daughter trajectory is the action, on the latter, of the external field, and the computation of system's response. Hence, the same LAMMPS script can be employed to generate both mother and daughter trajectory, without the need to run the daughter trajectories via a different script.

The procedure works as follows. After the thermalization, the sampling process starts. The system is followed over the equilibrium thermostatted dynamics (*fix nvt* command) for the desired number of timesteps. The sample  $\Gamma_n$  is then produced by storing positions and momenta via the command *fix store/state*. Once a sample is generated, the mother trajectory is halted (*unfix nvt*), and the script cycles over the daughters. Each daughter run is characterized by the following sequence of operations: set the system to state  $\Gamma_n$  via *set atom*, where the target values are the outputs of the *fix store/state*; modify the initial state accordingly to the mapping; set the nonequilibrium dynamics, call the function for the computation of the desired quantities (*fix ave/time*, *fix ave/chunk*, etc), and run the simulation. At the end of each NEMD trajectory, the nonequilibrium dynamics is halted as well as the computation of the response (*unfix ave/time*, *unfix ave/chuck*, etc). The process is cycled over all the four daughters. After the nonequilibrium runs have been generated, the system is set back to the state  $\Gamma_n$  and the mother trajectory is recovered. The system is then followed over the equilibrium dynamics until the next sample  $\Gamma_{n+1}$  overwrites the previous state  $\Gamma_n$ , when the cycle starts again. The loop is repeated the desired number of times. The scheme is shown in the Algorithm 2 and

#### Algorithm 2 LAMMPS SCRIPT.

```

1: initialize system (null ext. field)
2: run  $N_t$  timestep
3: save state
4: for  $n = 1, \dots, N_{\text{daughters}}$  do
5:   load system state
6:   run  $N_s$  timestep
7:   save system state
8:   for  $m = 1, \dots, N_{\text{mappings}}$  do
9:     load system state
10:    apply mapping
11:    switch on ext. field
12:    set computation of the response
13:    run  $N_d$  timestep
14:    switch off ext. field
15:   end for
16: end for

```

in Fig. 2. The outputs of the functions *fix ave/time*, *fix ave/chunk* are not written to file, but passed to the Python interface through which the LAMMPS script is launched. The loss in efficiency caused by the Python wrapper is abundantly compensated by a easier management of the simulations, since only few files are generated, and by avoiding reading/writing on files, which is typically inefficient both on a local machine and on a HPC cluster [34] Since *fix store/state* does not save the value of the thermostats, the periodically halted and resumed mother trajectory diverges from the one obtained via a single, uninterrupted run. The effect occurs at the transition from mother to daughter runs as well. The issue is irrelevant for the mother trajectory if the runs between samples are sufficiently long to fully regain an equilibrium canonical distribution. Moreover, it has an undetectable effect on the shear pressure, but it should potentially be accounted for if other quantities, such as the temperature, are monitored. In that case, the initial states should be stored and loaded via the commands *write\_restart* and *read\_restart*, and daughter and mother trajectories should have identical *fix nvt* commands. However, not every *fix* is stored in a *restart* file, and hence the transition from mother to daughter trajectory might not always be totally consistent.

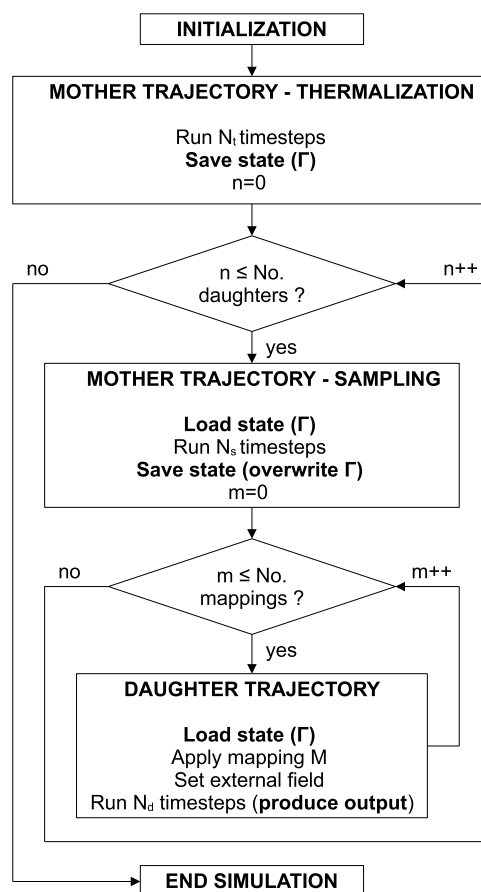


Fig. 2. Flowchart of the LAMMPS script for TTCF calculation through a single uninterrupted run. Note that the sequence of commands *save state* and *load state* at the end of the thermalization and at the beginning of the outer cycles, as well as the first decorrelation (Run  $N_s$  timesteps) are redundant, but necessary to preserve the correct structure of the flow.

Since the thermalization is typically very long, both setups can be optimized by running a smaller number of independent thermalizations over multicore runs. For instance, instead of 1000 thermalizations over single core jobs, 100 thermalizations with 10-core independent jobs can be performed. After the process is completed, the workflow is switched back to single core jobs. Hence, for each original trajectory, 10 further mother trajectories must be generated, on which the sampling of the initial states takes place. It is easy to generate further initial conditions from a single state by adding to the velocities a small random kick. Since the system is chaotic, the decorrelation between the trajectories is fast, and a small perturbation should not alter the thermodynamic state of equilibrium achieved through the thermalization. The precise gain in efficiency of this method is rather situational, since it depends, among other things, on the specific architecture of the cluster used for the simulation.

The key part of the Python code to run the TTCF is shown in Algorithm 3. The *tfcf* object is loaded from the TTCF module to handle data accumulation, integration, normalisation as well as plotting and writing to file. Each step required data for the TTCF calculation is added through *tfcf.add\_mappings* function, which in this example is global data (single time averages for the simulations such as temperature) and profile data (here the velocity at 100 equally sized chunks spanning the *y* direction). By separation into a module, both *tfcf* and *utils* can be unit tested by continuous integration system on Github Actions, used to validate basic functionality remains correct as the project evolves. The TTCF module is independent of LAMMPS so could be used by another MD solver. However, a key feature of TTCF4LAMMPS is the extrac-

tion of data from LAMMPS on the fly, using the `utils.get_fix_data` which can retrieve both global (ave/time) and 1D profile (ave/chunks) using `pylamm` interface to LAMMPS `extract_fix` function. The LAMMPS instance, `Imp`, manages the MD simulation and is run forward by a number of timesteps as set by the `Delay` variable, before data is extracted. In this way, writing to file is avoided as all collected data can be extracted directly into Python. This approach works both in serial and parallel, where each Message Passing Interface (MPI) process creates its own LAMMPS instances, collects its own data and this is reduced at the end of the simulation. The embarrassingly parallel nature of the problem is preserved while totally avoiding the need to write to disk, except for occasional checkpoints. Further continuous integration testing ensures `pylamm` can be imported correctly, before an integration test verifies the TTCF with LAMMPS collects the expected velocity data for a given strain rate, as well as ensures this gives better statistics than the DAV approach when strain rate is low. The example script below is designed to be easily extensible, with the `tcf` and `ultis` provides as simple helper functions applicable to any outputs. These save and load the LAMMPS state to allow mother/daughter trajectory checkpointing, apply mapping and simplify the process of turning on applied forcing or data collection (e.g. SLLOD or boundary driven as described in the next two subsections). However, the LAMMPS input syntax is retained as far as possible, making it easy for LAMMPS users to make changes. For example, to add addition variables to calculate profiles for by appending to `setlist` e.g. `setlist.append("compute my_compute all [details]")` and then add this to `profile_variables = ["vx", "c_my_compute"]` to collect the TTCF data. The Github page for the project includes an extensive README.md file to explain these features in detail.

Both scripts can also be run using graphical processing units (GPUs) by exploiting the GPU calculation package available in LAMMPS. The `lamm` scripts, both through the C++ and within the Python interface should be modified by adding the command `package gpu` at the beginning of the script and by modifying the definition of the atomic potential from `pair_style lj/cut` to `pair_style lj/cut/gpu`. The gain in efficiency depends on several factors but can be substantial for specific systems. A comparison of the efficiency of CPU and GPU calculation will be included in the next section.

### Example 1: shear of bulk systems

The bulk system is based on that used in a previous study by Borzák et al. [17]. It consists of 256 particles placed in a cubic box of fixed volume. Periodic boundary conditions are applied in all three Cartesian directions. The evolution is driven by the SLLOD equations of motion [35,1,12,36,7]. In LAMMPS, the dynamics is described by the following command: `fix deform` which deforms the simulation box according to the selected shear rate, in order to give the fluid a linear Couette flow profile. This approach is essentially equivalent to the popular Lees-Edwards sliding-brick boundary conditions [37]. Finally `fix nvt/slod` produced a thermostatted SLLOD dynamics, where the temperature is controlled with a Nosé-Hoover thermostat [38,39], and the equations integrated using the velocity-Verlet (or Störmer-Verlet) algorithm [40] (using the `run_style verlet` command). The thermostatted SLLOD dynamics is described by the following set of differential equations

$$\begin{aligned} \dot{\mathbf{r}}_i &= \frac{\mathbf{p}_i}{m_i} + \mathbf{i}\dot{\gamma}y_i \\ \dot{\mathbf{p}}_i &= -\sum_j \nabla\phi_{ij} - \mathbf{i}\dot{\gamma}p_{yi} - \alpha\mathbf{p}_i \\ \dot{\alpha} &= \frac{1}{Q} \left( \sum_i \mathbf{p}_i^2 - 3Nk_B T \right) \end{aligned} \quad (5)$$

where  $\alpha$  is the Nosé-Hoover thermostat multiplier [38,39] and  $Q$  its damping factor. The particles interact via the Weeks-Chandler-Andersen (WCA) potential [41], which is a truncated and shifted Lennard-Jones potential [42].

$$\phi(r_{ij}) = \begin{cases} 4\epsilon \left[ \left( \frac{\sigma}{r_{ij}} \right)^{12} - \left( \frac{\sigma}{r_{ij}} \right)^6 \right] + \phi_c, & \text{if } r_{ij} \leq r_c \\ 0, & \text{if } r_{ij} > r_c \end{cases} \quad (6)$$

with  $r_{ij}$ ,  $\sigma$ ,  $\epsilon$  the particles' distance, particles' diameter and potential well, respectively. The interaction radius  $r_c$  is  $2^{1/6}\sigma$  and the shift  $\phi_c$  is the constant required to guarantee the continuity of the potential in  $r_c$ . The dissipation function associated to the system and the resulting TTCF is then

$$\Omega = -\frac{\dot{\gamma}V}{k_B T} P_{xy} \quad \langle B(t) \rangle = \langle B(0) \rangle - \frac{\dot{\gamma}V}{k_B T} \int_0^t \langle P_{xy}(0)B(s) \rangle ds \quad (7)$$

with  $V$  the volume of the system,  $\dot{\gamma}$  the shear rate and  $P_{xy}$  the shear pressure. The simulations were carried out at the Lennard-Jones triple point ( $\rho^* = \rho\sigma^3 = 0.8442$ ;  $T^* = k_B T/\epsilon = 0.722$ ). We varied the reduced strain rate,  $\gamma^* = \gamma(m/\epsilon)^{1/2}\sigma$  from 1 to  $10^{-7}$ . The shear viscosity  $\eta$  was computed with the following TTCF expression

$$\langle \eta(t) \rangle = -\frac{\langle P_{xy}(t) \rangle}{\dot{\gamma}} = \frac{V}{k_B T} \int_0^t \langle P_{xy}(0)P_{xy}(s) \rangle ds \quad (8)$$

The timestep ( $\Delta t^* = \Delta t(m/\epsilon)^{1/2}/\sigma$ ) was 0.0025. Each transient NEMD segment was 600 timesteps long, and we generated  $4 \times 30000 = 120000$  of them, to yield reasonable statistics, with 30000 independent initial conditions and the corresponding mappings as described previously. The delay between samples along the equilibrium trajectory was 1000 timesteps. We have used reduced units throughout, unless stated otherwise. This first example is designed to be run on a local machine. As a result, the system is very small, and arguably the simplest system from which a realistic fluid response can be obtained. The total simulation time is approximately 4.5 hours on a single core run for each shear rate tested. Since the method is fully parallelized, it is expected that a four-core run would be completed in approximately an hour.

It is crucial that at the start of the daughter trajectory the correct velocity profile is superimposed on the momenta. This operation is required for the correct transient response, but is not automatically implemented in the LAMMPS routines for SLLOD dynamics. The operation can be performed with the command `set atom. velocity ramp` must be avoided for TTCF computation, since it overwrites the existing momenta with a linear profile, rather than superimposing it.

Fig. 3 displays the shear viscosity computed via direct average (DAV) and TTCF. The data are in good agreement at high shear rates, where shear thinning occurs. As previously anticipated, the computed viscosity shows the feature of TTCF formalism of generating data whose accuracy is not affected by the magnitude of the external force. In contrast, the direct average (DAV) rapidly loses precision as the shear rate decreases. The effect will be explored more extensively in the next section. We are aware that the current stable LAMMPS release (August 2023) contains several issues in the implementations of the SLLOD dynamics. These bugs are noticeable in the small discrepancy between the DAV and TTCF viscosities displayed, where the TTCF calculation is systematically smaller than its direct counterpart. These issues are currently being analysed and addressed by Prof Debra Bernhardt (Searles) and her group.<sup>1</sup> We tested the efficiency of GPU force calculation for both the original LAMMPS script and the corresponding Python interface. Fig. 4 shows the average time per particle spent to compute a single timestep. This is measured as a function of the number of particles in the system. The data have been obtained from a run of  $3 \times 10^4$  timesteps subdivided in  $2 \times 10^4$  for the mother trajectory and  $10^4$  timesteps for the daughter. In both setups, the GPU force calculation can speed up the simulation for large systems ( $> 4000$  particles), but performs worse in smaller systems. The decrease in performance

<sup>1</sup> Private communications.

**Algorithm 3** Python code in run\_TTCF.py which uses pyLAMMPS interface together with the TTCF package.

```

1 lmp = lammmps(comm=MPI.COMM_SELF, cmdargs=args)
2
3 #Run equilibration
4 lmp.file("system_setup.in")
5 lmp.command("timestep " + str(dt))
6 utils.run_mother_trajectory(lmp,Nsteps_Thermalization,Thermo_damp)
7
8 #Save snapshot to use for daughters
9 state = utils.save_state(lmp, "snapshot")
10
11 #Loop over all sets of daughters
12 for Nd in range(Ndaughters):
13
14     #Run mother starting from previous sample to generate the next sample
15     utils.load_state(lmp, state)
16     utils.run_mother_trajectory(lmp,Nsteps_Decorrelation,Thermo_damp)
17     state = utils.save_state(lmp, "snapshot")
18
19     #Branch off daughters for each mapping
20     for Nm in range(Nmappings):
21
22         #Load child state
23         utils.load_state(lmp, state)
24
25         #Apply mapping
26         utils.apply_mapping(lmp, Maps[Nm])
27
28         #Apply forces and setup outputs
29         utils.set_list(lmp, setlist)
30
31         #Run zero to setup case
32         lmp.command("run 0 pre yes post yes")
33
34         #Extract profile and time averaged (global) data from LAMMPS
35         data_profile[0, :, :] = utils.get_fix_data(lmp, "Profile_variables",
36                                                     profile_variables, Nbins)
37         data_global[0, :] = utils.get_fix_data(lmp, "Global_variables",
38                                               global_variables)
39         omega = data_global[0, -1]
40
41         #Run over time
42         for t in range(1, Nsteps):
43             lmp.command("run " + str(Delay) + " pre yes post no")
44             data_profile[t, :, :] = utils.get_fix_data(lmp, "Profile_variables",
45                                                         profile_variables, Nbins)
46             data_global[t, :] = utils.get_fix_data(lmp, "Global_variables",
47                                                    global_variables)
48
49         #Turn off forces and outputs
50         utils.unset_list(lmp, setlist)
51
52         #Sum the mappings together
53         ttcf.add_mappings(data_profile, data_global, omega)
54
55     #Perform the integration
56     ttcf.integrate(dt*Delay)

```

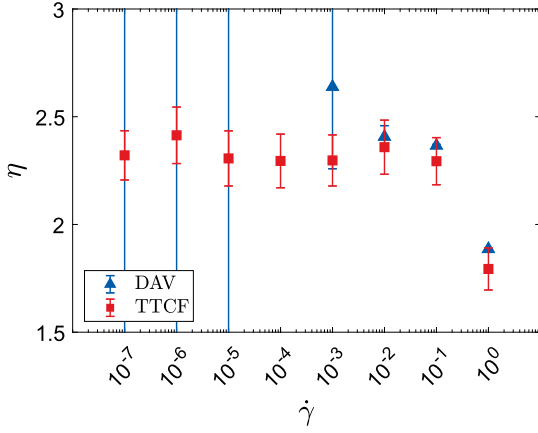
of the Python interface is relevant for very small systems, but becomes negligible for more common system sizes. The tests have been run on local machines (CPU Intel Core i7-11800H and GPU NVIDIA GeForce RTX 3035 Ti), where writing/reading on files is more efficient than on HPC clusters. The reason for the lower speed is due to the requirement of the pylammps package to halt and restart the daughter trajectory whenever the output is produced (in the presented results, the pressure is computed every 5 timesteps). The command *run pre yes post no* is required to obtain the correct output calculation, whereas *run pre no post no* would result in similar performances for both the original script and the Python implementation. The issue can be lessened by decreasing the frequency of output generation, as the correlation function can typically be integrated with a very large step, up to 50 or more times larger than the timestep set for the simulation. This option can also increase the efficiency of GPU force calculation. Moreover, different LAMMPS output calculations might not need the option *pre yes post no*. It is advised these

options be checked for a specific case, starting with *run pre yes post no* and comparing results to see if optimisation is possible.

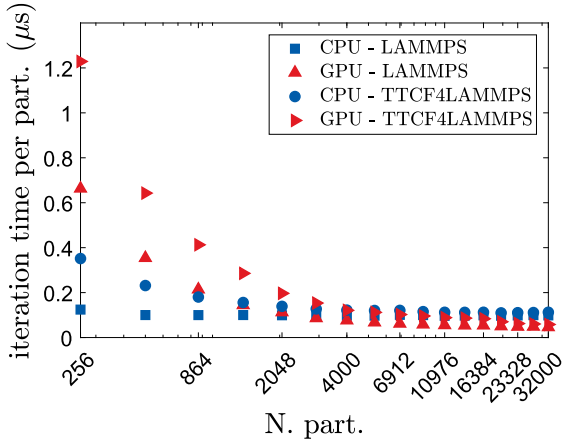
**Example 2: shear of confined systems**

The confined system is based on that used in a previous study by Maffioli et al. [27], where a narrow three-dimensional channel was analysed. The system is composed of Lennard-Jones particles, and the walls are kept in position by tethering the particles to lattice sites via harmonic springs. The boundary-driven Couette flow is generated by moving the lattices in opposite directions at constant velocity  $v$  [43]. The LJ potential is defined as

$$\phi(r)_{ij} = 4\epsilon \left[ \left( \frac{\sigma}{r_{ij}} \right)^{12} - c_{ij} \left( \frac{\sigma}{r_{ij}} \right)^6 \right] \quad (9)$$



**Fig. 3.** Shear viscosity for the bulk system at different shear rates ( $5 \times 10^4$  -  $5 \times 10^{11}$  in SI units). Both DAV and TTCF signals can detect the reduction in the viscosity for high shear rates (shear thinning). The accuracy of the direct average rapidly decreases for  $\dot{\gamma} \rightarrow 0$ . In contrast, the TTCF signal retains the same accuracy regardless of the magnitude of the driving force.



**Fig. 4.** Efficiency (microseconds per particle per timestep) of the CPU and GPU setups, for both original LAMMPS script and the Python interface. GPU force calculation is more efficient for medium and large systems. The Python interface decreases the performances for small systems, but it's efficiency is comparable to the original LAMMPS script for large systems.

where the wetting parameter  $c_{ij}$  is modulated to promote slip between wall and fluid particles, and to enhance cohesion within the fluid. The equations of motion describing the dynamics of the system are:

$$\begin{aligned}
 \dot{\mathbf{r}}_i^f &= \frac{\mathbf{p}_i^f}{m_i} \\
 \dot{\mathbf{p}}_i^f &= - \sum_j \nabla \phi_{ij} \\
 \dot{\mathbf{r}}_i^w &= \frac{\mathbf{p}_i^w}{m_i} \\
 \dot{\mathbf{p}}_i^w &= - \sum_j \nabla \phi_{ij} - k(\mathbf{r}_i^w - \mathbf{r}_i^l) - \alpha \mathbf{p}_i^w \\
 \dot{\alpha} &= \frac{1}{Q} \left( \sum_i \mathbf{p}_i^{w2} - 3N^w k_B T \right) \\
 \dot{\mathbf{r}}_i^l &= (\pm v, 0, 0) \\
 \dot{\mathbf{p}}_i^l &= (0, 0, 0)
 \end{aligned} \tag{10}$$

with superscript  $f$ ,  $w$  and  $l$  denoting the fluid, wall, and lattice particles respectively.  $\mathbf{r}$  and  $\mathbf{p}$  are the positions and the momenta of the particles,

$k$  is the stiffness of the harmonic spring tethering the wall particles to the lattice sites.  $\alpha$  is the Nosé-Hoover thermostat multiplier, acting on  $N^w$  wall particles, and  $T$  the target temperature of the walls. The associated dissipation function is

$$\Omega = - \frac{1}{k_B T} \sum_i k(\mathbf{r}_i^w - \mathbf{r}_i^l) v \tag{11}$$

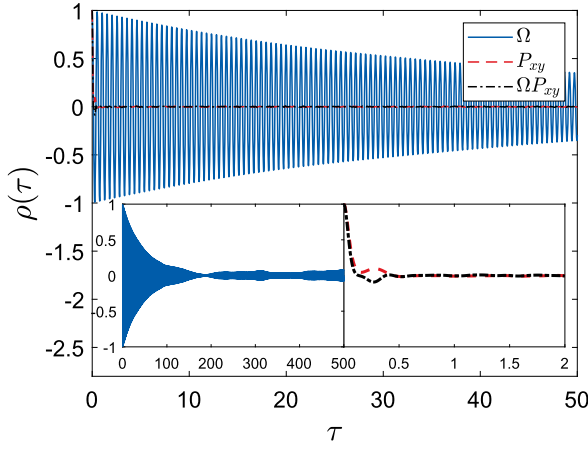
summed over both walls. In our previous work [27], we showed that it is also valid for inhomogeneous systems kept at constant pressure by means of a barostat [44]. The wall temperature is set to 1. The system is composed of 6800 fluid particles and 2600 wall particles with diameter  $\sigma = 1$  and interaction radius  $r_c = 2.5$ , at the densities of 0.7 and 0.8 respectively in reduced units. The wetting parameter is 0.6325 for the wall-fluid interactions and 1.2 for the fluid-fluid potential [45]. The system is approximately  $30\sigma$  long in the  $x$  and  $z$  direction, and the channel width is set to  $10\sigma$ . This is expected to be sufficiently large in the lateral dimensions to avoid size effects on the liquid-solid friction [46]. The equations of motion were integrated with the standard velocity-Verlet [40] algorithm with integration step equal to 0.005.

We adopted the first scheme described in the previous section, and  $10^4$  independent mother trajectories were generated. Each of them was thermalized for 5000 time units ( $10^6$  timesteps) before the sampling took place. The starting points were produced with a lag of 5 time units from each other.  $10^2$  initial states were generated for each mother, for a total of  $4 \times 10^6$  nonequilibrium daughter trajectories. Each nonequilibrium system was monitored for 12.5 time units, or 2500 time steps. Hence, a total of  $2.1 \times 10^{10}$  timesteps were required for each simulation.

The simulations were performed on the Swinburne University supercomputer OzSTAR, which features Intel Gold 6140 processors. Each run, composed of  $2.1 \times 10^6$  time steps, was performed in approximately 6 hours. Since the resources needed for each single run were very limited, the total simulation was performed in a very short time. If a higher efficiency is needed, each mother trajectory can be thermalized for shorter than 5000 time units, which is arguably very conservative. The mother trajectory was generated via the usual `fix_nvt` command. The same dynamics was applied on the daughter trajectories. The only difference in the dynamics between equilibrium and nonequilibrium runs was the constant velocity in the  $x$  direction imposed to the lattice sites in the second case. As previously described, `write_restart` and `read_restart` commands were used to store and load the initial states. This also allowed us to run the mother trajectories one time only, and use the same sample for all the shear rates we tested. We computed the shear pressure  $P_{xy}$  at the wall-fluid interface, using the method of planes [47], and the velocity profile. We also compared the calculation of the friction coefficient using equilibrium methods with the direct measurement over the NEMD trajectories. The equilibrium method is described in previous works [48,11], and is based on the following definition:

$$\langle F_x \rangle = \xi_0 A \langle \Delta v \rangle \tag{12}$$

where  $\xi_0$  is the intrinsic friction coefficient and  $A$  the interface surface.  $\langle F_x \rangle$  and  $\langle \Delta v \rangle$  are respectively the instantaneous average force between the wall and the fluid particles located in a slab adjacent to the wall, and the instantaneous average slip velocity of the slab. The data for the calculation were obtained from the window of 500 time units along the mother trajectory, over which the starting points for the NEMD segments were generated. The DAV and TTCF data were instead produced by a direct nonequilibrium computation of  $\langle F_x \rangle$  and  $\langle \Delta v \rangle$  from the daughter trajectories. We investigated the fluid response to a shear rate spanning over 15 order of magnitude, from  $10^{-1}$  to  $10^{-15}$  in reduced units ( $5 \times 10^{10}$  -  $5 \times 10^{-4}$  in SI units). This range is possibly the full window over which the system can be examined by numerical simulations. Higher shear rates trigger a resonance with the harmonic bonds tethering the wall particles, resulting in wide wall oscillations and making the setup inefficient. For weaker fields, the response might become hard to detect due to the finite precision of the floating-point representation, which is approximately  $10^{-15}$  for double precision numbers.

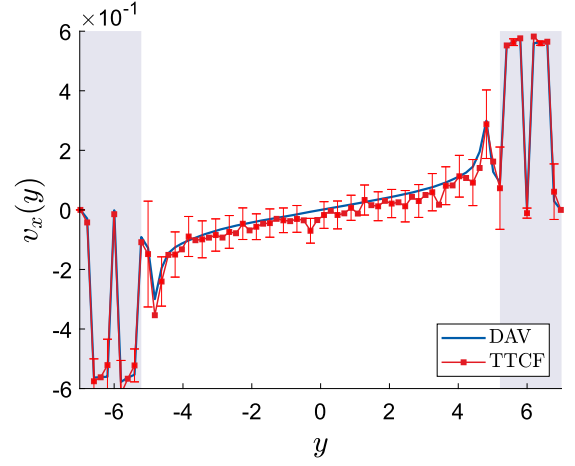


**Fig. 5.** Autocorrelation of the dissipation function  $\Omega$ , the shear pressure  $P_{xy}$ , and the integrand function  $\Omega P_{xy}$ . Bottom left is the autocorrelation of  $\Omega$  for a maximum lag of 500 time units. Bottom right, the autocorrelation of  $P_{xy}$  and  $\Omega P_{xy}$ , magnified with a narrower band of 2 time units.

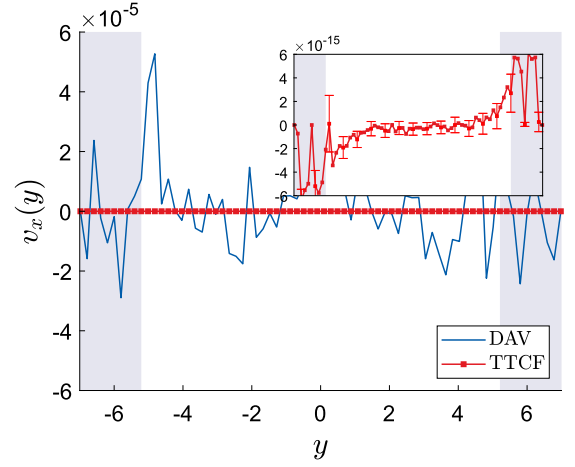
In Fig. 5, the autocorrelation function for various quantities is displayed. The variables have been computed from a single equilibrium trajectory. Since at equilibrium the dissipation function is identically null, only the total wall-lattice harmonic force is displayed as  $\Omega$  (cf. Eq. (11)). The presence of the harmonic springs combined with a weak wall-fluid interaction makes the wall effectively a solid and hence the autocorrelation function of  $\Omega$  persists for a long time, in the order of 200 time units, but does not fully decay. On the other hand,  $P_{xy}$  and  $\Omega P_{xy}$  decorrelate in less than 1 time unit, meaning that the starting points are reasonably independent for  $T_s \geq 1$ . If the samples are correlated, the resulting signal might be both biased and more dispersed. The short range decay in the autocorrelation is also possibly an indicator for the success of the TTCF method as it prevents the fluid from being in, or close to, the solid state, as noted earlier.

Fig. 6 shows the velocity profile at  $t = 12.5$  for  $\dot{\gamma} = 10^{-1}$  and  $10^{-15}$  computed via DAV and TTCF. The error bars for the direct average have been omitted since the signal is either extremely accurate for the high shear, or extremely noisy for the weak field. The sigmoid shape of the velocity profile is the result of the small channel width combined with the Lennard-Jones interaction parameters chosen, and it persists in the steady state. In Fig. 7 the transient of the shear pressure at the wall-fluid interface is displayed from  $t = 0$ . The error bars for the direct average have been omitted again. Both the velocity and the pressure show that for  $\dot{\gamma} = 10^{-15}$  the DAV signal is essentially composed of random noise. In Fig. 8 the shear pressure and the slip velocity as a function of the shear rate are displayed. The slip velocity is defined as the difference of the velocity of the innermost wall layer and the outermost fluid layer. The data show that the DAV can produce acceptable results for  $\dot{\gamma} \geq 10^{-5}$ . For weaker fields, the statistical fluctuations drown out the response, and the signal is constant across the different shear rates.

On the other hand, all the phase variables displayed suggest that the TTCF method is able to identify the response at any level of the shear rate. The accuracy of the TTCF method increases as the shear rate decreases and the signal-to-noise ratio (SNR), defined as the ratio between the signal and its standard deviation, remains approximately constant for any magnitude of the shearing force. The property can be proven as follows. For simplicity, the fluctuations of any quantity (e.g.  $P_{xy}$ ) are assumed to be constant for every shear rate. This condition is valid for weak shear rates, and it essentially holds across the entire range tested in this work. Secondly, the dissipation function at  $t = 0$  is proportional to the shear rate. This relation is exact and independent of the rate itself, and it immediately follows from Eq. (7) and (11), where in the expression of the dissipation function two terms are clearly detectable. The first term is a pressure/force contribution ( $P_{xy}$  for the



(a)



(b)

**Fig. 6.** Velocity profile at  $t = 12.5$  for  $\dot{\gamma} = 10^{-1}$  (a), and  $\dot{\gamma} = 10^{-15}$  (b) (resulting wall velocity  $v \approx \pm 10^2$  m/s and  $v \approx \pm 10^{-12}$  m/s). The error bars are four times the standard error. The shaded regions indicate the solid walls.

bulk systems,  $x$  component of the spring forces for the channel), and in statistical mechanics is usually denoted as dissipative flux conjugate to the generalized external field. For our purpose, we note that this is computed at  $t = 0$ , that is, an *equilibrium* state, and hence is independent of the external field. The second term is the effective external force and is equal, or proportional, to  $\dot{\gamma}$  or  $v$ . As a result,  $\Omega(0)$  is proportional to the magnitude of the external driving force, and we have the following set of relations, shown for simplicity only for the shear pressure and the dissipation function:

$$\begin{aligned} \langle \Omega(0) \rangle &= 0 & \sigma(\Omega(0)) &= k_1 \dot{\gamma} \\ \langle P_{xy}(t) \rangle &= \hat{P}_{xy}(\dot{\gamma}) & \sigma(P_{xy}(t)) &= k_2 \end{aligned} \quad (13)$$

where  $\sigma$  is the standard deviation,  $k_i$  are arbitrary constants, and  $\hat{P}_{xy}$  is the real shear pressure. If we restrict, for convenience, our analysis to a steady state, the functions  $\Omega(0)$  and  $P_{xy}$  are uncorrelated and the variance  $\sigma^2$  of the integrand function in Eq. (7) and (11) is:

$$\begin{aligned} \sigma^2(\Omega(0)P_{xy}(t)) &= \\ &= (\sigma^2(\Omega(0)) + \langle \Omega(0) \rangle^2) \sigma^2(P_{xy}(t)) \\ &+ \langle P_{xy}(t) \rangle^2 - \langle \Omega(0) \rangle \langle P_{xy}(t) \rangle \\ &= (k_1^2 \dot{\gamma}^2) (k_2^2 + \hat{P}_{xy}^2) \end{aligned} \quad (14)$$



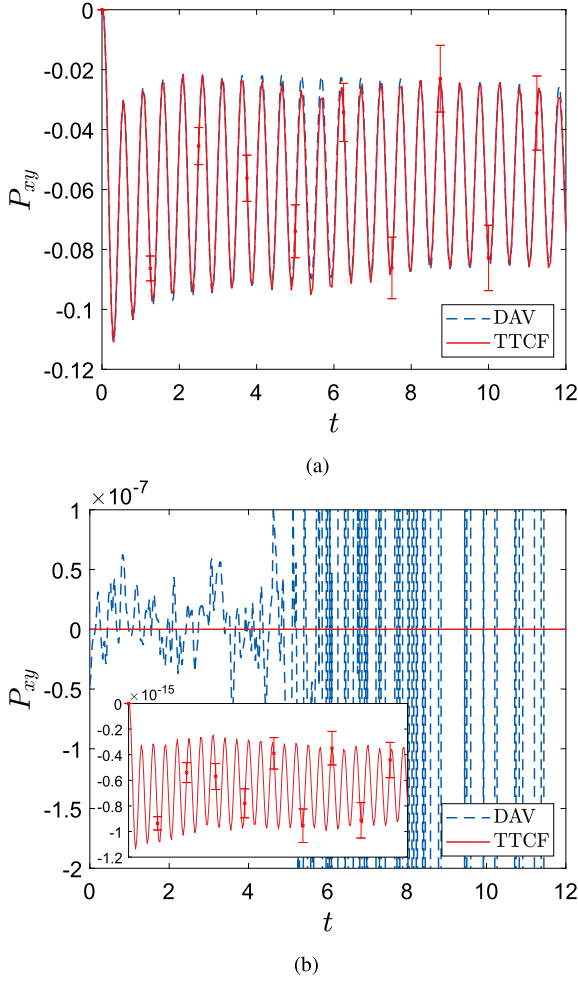


Fig. 7. Shear pressure as a function of time for  $\dot{\gamma} = 10^{-1}$  (a), and  $\dot{\gamma} = 10^{-15}$  (b) ( $5 \times 10^{10} \text{s}^{-1}$  and  $5 \times 10^{-4} \text{s}^{-1}$  in SI units). The error bars are four times the standard error.

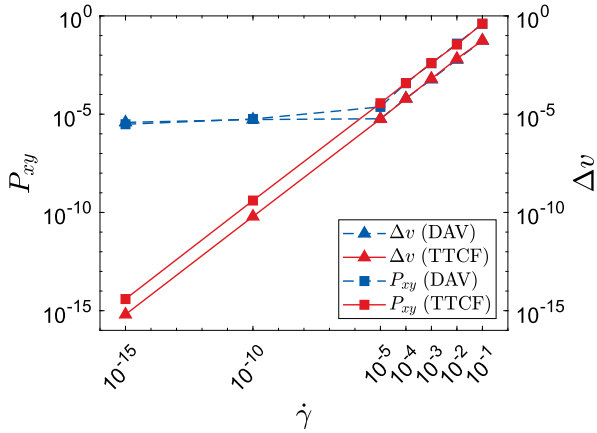


Fig. 8. Shear pressure and velocity profile for DAV and TTCF for every shear rate tested spanning from  $5 \times 10^{10} \text{s}^{-1}$  to  $5 \times 10^{-4} \text{s}^{-1}$  in SI units. Error bars have been omitted.

The effect of the integration can hardly be modelled by simple functions, but it depends solely on the time  $t$ , and hence the SNR of the shear pressure

$$\text{SNR} = \frac{\langle P_{xy}(t) \rangle}{\sigma(P_{xy}(t))}$$

is expressed by the following identities:

$$\text{SNR}_{\text{DAV}} = \frac{\hat{P}_{xy}(\dot{\gamma})}{k_2}$$

$$\text{SNR}_{\text{TTCF}} = K(t) \frac{\hat{P}_{xy}(\dot{\gamma})}{k_1 \dot{\gamma} \sqrt{k_2^2 + \hat{P}_{xy}^2(\dot{\gamma})}} = \frac{K(t)}{k_1 \dot{\gamma}} \frac{\text{SNR}_{\text{DAV}}}{\sqrt{1 + \text{SNR}_{\text{DAV}}^2}} \quad (15)$$

where  $K(t)$  is a function accounting for the integration process. Since  $\hat{P}_{xy} \simeq c\dot{\gamma}$  for  $\dot{\gamma} \rightarrow 0$ , we have

$$\lim_{\dot{\gamma} \rightarrow 0} \text{SNR}_{\text{DAV}} = 0$$

$$\lim_{\dot{\gamma} \rightarrow 0} \text{SNR}_{\text{TTCF}} = \frac{K(t)c}{k_1} \quad (16)$$

and

$$\lim_{\dot{\gamma} \rightarrow \infty} \text{SNR}_{\text{DAV}} = \infty$$

$$\lim_{\dot{\gamma} \rightarrow \infty} \text{SNR}_{\text{TTCF}} = 0. \quad (17)$$

The last relations suggest that the TTCF methods are eventually outperformed by a simple direct average, and that they progressively lose accuracy as the external field is increased.

Fig. 9 summarizes the various results: the standard error  $\text{SE}$  of  $P_{xy}(t)$  and  $\Omega(0)P_{xy}(t)$  are displayed for all the shear rates ( $\text{SE} = \sigma/\sqrt{N}$  with  $N = 4 \times 10^6$  the total number of daughter trajectories). The latter is approximately constant along the NEMD segment, and proportionally increases with  $\dot{\gamma}$ . In the steady states, the DAV fluctuations are independent of  $\dot{\gamma}$ , while they exponentially increase from a null value during the transient. This effect occurs because the mappings selected guarantee that  $\langle \Omega(0) \rangle = 0$ , but also  $\langle P_{xy}(0) \rangle = 0$ . In other words, the averaging over the mappings artificially eliminate the instantaneous fluctuations of the pressure, and the accuracy of the direct average at  $t = 0$  is virtually infinite. This correlation is retained for the first moments of the NEMD trajectory, whose signal is still characterized by low statistical uncertainty. The chaotic nature of the system makes the trajectories exponentially diverge, and, as a result, the standard error increases until the trajectories become uncorrelated, and the fluctuations stabilize to a steady value. After this, a measure obtained from a sample of  $N$  initial states along with their three mappings is equivalent to a set of  $4N$  independent states. Similar features are exploited in previous works [49], where, in order to improve the accuracy of traditional NEMD simulations, the nonequilibrium segments are paired with equilibrium ones starting from the same initial condition. The quantity of interest is monitored over both trajectories and the final result is the difference between the two measures. At the start of the run, where the two trajectories are highly correlated, the procedure eliminates from the NEMD signal the random fluctuations, obtained from the equilibrium signal. However, the systems studied in NEMD simulation are almost invariably chaotic, hence the trajectories decorrelate rapidly, and the obtained signal may lose most of its accuracy well before the system has reached a steady state. The rate of divergence of the trajectories is also driven by the magnitude of the shear: strong fields make the system more chaotic. This so-called *background subtraction* technique was demonstrated to be unsuitable for practical systems of interest [12].

In Fig. 10 the signal-to-noise ratio is displayed, for the shear pressure, the slip velocity and the friction coefficient for both DAV and TTCF. The threshold below which the TTCF outperforms the direct average is confirmed to be  $\dot{\gamma} = 10^{-5}$  for all the quantities. A SNR of the order of 1, or below, entails that the magnitude of the statistical fluctuations are comparable or larger than the signal, and no information can be obtained from the data.

Fig. 11 displays the friction coefficient, computed using the equilibrium and nonequilibrium methods. The black line represents the data computed via the equilibrium method. Since the equilibrium calculation is not associated to a specific shear rate, the line is constant across

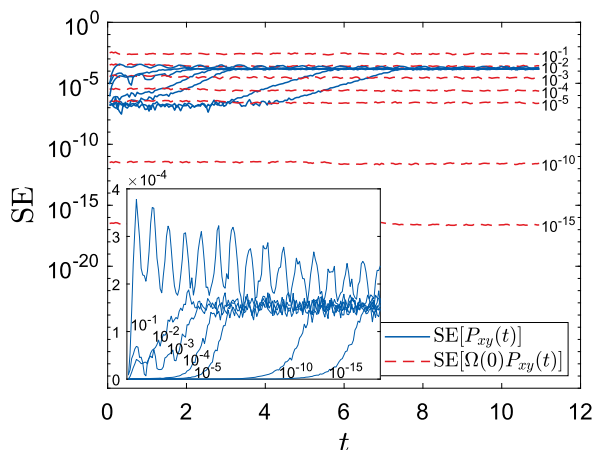


Fig. 9. Standard error of  $P_{xy}(t)$  and of  $\Omega(0)P_{xy}(t)$ . The inset shows the first quantity only, magnified. The different curves are for different shear rates.

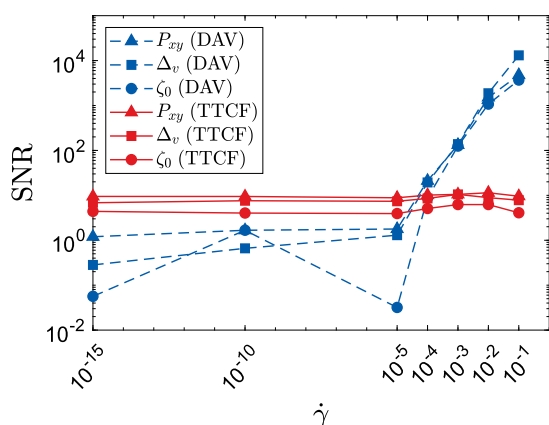


Fig. 10. Signal-to-noise ratio for various quantities for DAV and TTCF as a function of the reduced shear rate.

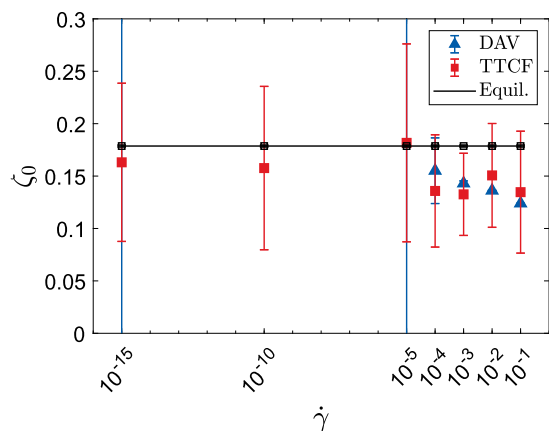


Fig. 11. Friction coefficient as a function of the shear rate, computed with DAV and TTCF. The black squares indicate the (constant) equilibrium calculation. The DAV signal at  $\dot{\gamma} = 10^{-5}$  and  $\dot{\gamma} = 10^{-15}$  does not appear in the range of the plot, and for  $\dot{\gamma} = 10^{-10}$  the confidence interval itself is out of scale.

the figure, and displayed for the mere purpose of comparison with the NEMD measurements. The data show the same behaviour as previous quantities, with DAV being progressively less accurate and the TTCF retaining the same precision. In comparison, the equilibrium method is drastically better at any level. For shear rates beyond  $10^5$  the accuracy

of the DAV is comparable to the equilibrium method, and the friction coefficient starts to deviate from the linear response.

The friction coefficient is particularly hard to estimate via direct average because a ratio of two random variables can be highly unstable. The effect is severe at low shear rate, where the signal may even be out of the numeric range displayed here.

The results confirm the model for the SNR previously derived, with the caveat that the numerator of the direct average must be replaced by the real response, computed, for instance, with the TTCF method. The TTCF signal-to-noise ratio is approximately constant for every shear rate, and it is expected to decrease for higher shear rates, when deviations from linear response become relevant. The effect might already start at  $\dot{\gamma} = 10^{-1}$ , where the SNR slightly decreases. We note that for more complex molecular fluids, such as alkanes or polymer melts in solutions, nonlinear effects manifest at much weaker shear rates, making TTCF the technique of choice for its superior statistical accuracy at experimental strain rates.

## Conclusion

We applied the TTCF technique in the investigation of an atomic fluid confined in a narrow channel undergoing planar Couette flow. We have shown that the method is readily implementable in simulation software such as LAMMPS and can be generalized to molecular fluids, and we listed the key ingredients for its efficient usage. We highlighted the key features and advantages of the TTCF formalism: i) TTCF measurements retain the same signal-to-noise ratio for arbitrarily weak fields, to the point that the lower bound is now set by the finite precision of the computer arithmetic. We provided a simple proof of this phenomenon and, given the generality of the mechanism under which it occurs, we expect it to hold for other types of driving fields. ii) While equilibrium methods may be computationally more efficient, they are limited to the linear response regime, whereas TTCF is valid for any magnitude of the external force. iii) The dissipation function is exclusively related to the nature of the external field, and not to the interatomic forces. As such, it is immediately extendable to more complex systems. Particularly, the scripts we provided can be extended to molecular fluids by simply redefining the atomic interactions and bonds.

The application of the TTCF formalism demands significant computational resources, but it rapidly outperforms the direct average for weak driving fields, which correspond to those that can be experimentally probed in the laboratory. As a comparison, the resources needed to monitor the system response at  $\dot{\gamma} = 10^{-15}$  via DAV could be as much as  $10^{20}$  times larger than those needed for the TTCF algorithm. Additionally, we have shown that the simulations can be split into a massive number of short, independent tasks, hence dramatically increasing their efficiency.

## CRediT authorship contribution statement

**Luca Maffioli:** Writing – original draft, Software, Methodology, Investigation, Formal analysis. **James P. Ewen:** Writing – review & editing, Methodology. **Edward R. Smith:** Writing – review & editing, Software. **Sleebea Varghese:** Investigation. **Peter J. Davis:** Writing – review & editing, Supervision, Methodology, Funding acquisition. **Daniele Dini:** Writing – review & editing, Supervision, Funding acquisition. **B.D. Todd:** Writing – review & editing, Supervision, Methodology, Funding acquisition.

## Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Luca Maffioli reports financial support was provided by Australian Research Council (Grant No. DP200100422). Billy Todd reports financial support was provided by Australian Research Council (Grant No.

DP200100422). Peter Davis reports financial support was provided by Australian Research Council (Grant No. DP200100422). Daniele Dini reports financial support was provided by Australian Research Council (Grant No. DP200100422). James Ewen reports financial support was provided by The Royal Society (Grant No. IES/R3/170/233). Daniele Dini reports financial support was provided by Royal Academy of Engineering. James Ewen reports financial support was provided by Royal Academy of Engineering. Daniele Dini reports financial support was provided by Shell. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

All data used in this work are freely reproducible using the program provided. The source files are also available at the link <https://github.com/edwardsmith999/TTCF4LAMMPS>.

## Acknowledgements

We thank the Australian Research Council for a grant obtained through the Discovery Projects Scheme (Grant No. DP200100422) and The Royal Society for support via International Exchanges, Grant No. IES/R3/170/233. J.P.E. was supported by the Royal Academy of Engineering (RAEng) through their Research Fellowships scheme. D.D. was supported through a Shell/RAEng Research Chair in Complex Engineering Interfaces. The authors acknowledge the Swinburne OzSTAR Supercomputing facility and the Imperial College London Research Computing Service (DOI:<https://doi.org/10.14469/hpc/2232>) for providing computational resources for this work. We thank Debra Bernhardt and Stephen Sanderson (University of Queensland) for useful discussions regarding the implementation of SLLD in LAMMPS.

## References

- [1] D.J. Evans, W.G. Hoover, Flows far from equilibrium via molecular dynamics, *Annu. Rev. Fluid Mech.* 18 (1986) 243–264, <https://doi.org/10.1146/annurev.fl.18.010186.001331>.
- [2] E.R. Smith, A molecular dynamics simulation of the turbulent Couette minimal flow unit, *Phys. Fluids* 27 (2015) 115105, <https://doi.org/10.1063/1.4935213>.
- [3] D. Savio, L. Pastewka, P. Gumbsch, Boundary lubrication of heterogeneous surfaces and the onset of cavitation in frictional contacts, *Sci. Adv.* 2 (2016) e1501585, <https://doi.org/10.1126/sciadv.1501585>.
- [4] A. Martini, H.Y. Hsu, N.A. Patankar, S. Lichter, Slip at high shear rates, *Phys. Rev. Lett.* 100 (2008) 206001, <https://doi.org/10.1103/PhysRevLett.100.206001>.
- [5] S. Bair, C. McCabe, P.T. Cummings, Comparison of nonequilibrium molecular dynamics with experimental measurements in the nonlinear shear-thinning regime, *Phys. Rev. Lett.* 88 (2002) 058302, <https://doi.org/10.1103/PhysRevLett.88.058302>.
- [6] V. Jadhao, M.O. Robbins, Probing large viscosities in glass-formers with nonequilibrium simulations, *Proc. Natl. Acad. Sci. USA* 114 (2017) 7952–7957, <https://doi.org/10.1073/pnas.1705978114>.
- [7] B. Todd, P.J. Daivis, *Nonequilibrium Molecular Dynamics: Theory, Algorithms and Applications*, Cambridge University Press, 2017.
- [8] J.P. Ewen, D.M. Heyes, D. Dini, Advances in nonequilibrium molecular dynamics simulations of lubricants and additives, *Friction* 6 (2018) 349–386, <https://doi.org/10.1007/s40544-018-0207-9>.
- [9] M.S. Green, Markoff random processes and the statistical mechanics of time-dependent phenomena. II. Irreversible processes in fluids, *J. Chem. Phys.* 22 (1954) 398–413, <https://doi.org/10.1063/1.1740082>.
- [10] R. Kubo, Statistical-mechanical theory of irreversible processes. I. General theory and simple applications to magnetic and conduction problems, *J. Phys. Soc. Jpn.* 12 (1957) 570–586, <https://doi.org/10.1143/JPSJ.12.570>.
- [11] S. Varghese, J.S. Hansen, B.D. Todd, Improved methodology to compute the intrinsic friction coefficient at solid–liquid interfaces, *J. Chem. Phys.* 154 (2021) 184707, <https://doi.org/10.1063/5.0040191>.
- [12] D. Evans, G. Morriss, *Statistical Mechanics of Nonequilibrium Liquids*, Academic Press, London, 1990.
- [13] D. Evans, D. Searles, S. Williams, *Fundamentals of Classical Statistical Thermodynamics: Dissipation, Relaxation, and Fluctuation Theorems*, John Wiley & Sons, 2016.
- [14] S.S. Sarman, D.J. Evans, P.T. Cummings, Recent developments in non-Newtonian molecular dynamics, *Phys. Rep.* 305 (1998) 1–92, [https://doi.org/10.1016/S0370-1573\(98\)00018-0](https://doi.org/10.1016/S0370-1573(98)00018-0).
- [15] G.P. Morriss, D.J. Evans, Application of transient correlation functions to shear flow far from equilibrium, *Phys. Rev. A* 35 (1987) 792–797, <https://doi.org/10.1103/PhysRevA.35.792>.
- [16] D.J. Evans, G.P. Morriss, Transient-time-correlation functions and the rheology of fluids, *Phys. Rev. A* 38 (1988) 4142–4148, <https://doi.org/10.1103/PhysRevA.38.4142>.
- [17] I. Borzák, P.T. Cummings, D.J. Evans, Shear viscosity of a simple fluid over a wide range of strain rates, *Mol. Phys.* 100 (2002) 2735–2738, <https://doi.org/10.1080/00268970210137275>.
- [18] C. Desgranges, J. Delhommelle, Accurate determination of normal stress differences via transient-time correlation function – non-equilibrium molecular dynamics (ttcf-nemd) simulations, *Mol. Simul.* 35 (2009) 405–408, <https://doi.org/10.1080/08927020802575598>.
- [19] G. Pan, C. McCabe, Prediction of viscosity for molecular fluids at experimentally accessible shear rates using the transient time correlation function formalism, *J. Chem. Phys.* 125 (2006) 194527, <https://doi.org/10.1063/1.2364899>.
- [20] O.A. Mazyar, G. Pan, C. McCabe, Transient time correlation function calculation of the viscosity of a molecular fluid at low shear rates: a comparison of stress tensors, *Mol. Phys.* 107 (2009) 1423–1429, <https://doi.org/10.1080/00268970902922625>.
- [21] C. Desgranges, J. Delhommelle, Rheology of liquid fcc metals: equilibrium and transient-time correlation-function nonequilibrium molecular dynamics simulations, *Phys. Rev. B* 78 (2008) 184202, <https://doi.org/10.1103/PhysRevB.78.184202>.
- [22] C. Desgranges, J. Delhommelle, Shear viscosity of liquid copper at experimentally accessible shear rates: application of the transient-time correlation function formalism, *J. Chem. Phys.* 128 (2008) 084506, <https://doi.org/10.1063/1.2829872>.
- [23] B.D. Todd, Application of transient-time correlation functions to nonequilibrium molecular-dynamics simulations of elongational flow, *Phys. Rev. E* 56 (1997) 6723–6728, <https://doi.org/10.1103/PhysRevE.56.6723>.
- [24] R. Hartkamp, S. Bernardi, B.D. Todd, Transient-time correlation function applied to mixed shear and elongational flows, *J. Chem. Phys.* 136 (2012) 064105, <https://doi.org/10.1063/1.3684753>.
- [25] J. Delhommelle, P.T. Cummings, Simulation of friction in nanoconfined fluids for an arbitrarily low shear rate, *Phys. Rev. B* 72 (2005) 172201, <https://doi.org/10.1103/PhysRevB.72.172201>.
- [26] S. Bernardi, S.J. Brookes, D.J. Searles, D.J. Evans, Response theory for confined systems, *J. Chem. Phys.* 137 (2012) 074114, <https://doi.org/10.1063/1.4746121>.
- [27] L. Maffioli, E.R. Smith, J.P. Ewen, P.J. Daivis, D. Dini, B.D. Todd, Slip and stress from low shear rate nonequilibrium molecular dynamics: the transient-time correlation function technique, *The Journal of Chemical Physics* 156, <https://doi.org/10.1063/5.0088127>.
- [28] S. Bernardi, D.J. Searles, Local response in nanopores, *Mol. Simul.* 42 (2016) 463–473, <https://doi.org/10.1080/08927022.2015.1049174>.
- [29] J. Delhommelle, P.T. Cummings, J. Petracic, Conductivity of Molten sodium chloride in an arbitrarily weak dc electric field, *J. Chem. Phys.* 123 (2005) 114505, <https://doi.org/10.1063/1.2035085>.
- [30] N.J. English, D.A. Mooney, S.W. O'Brien, Electrical conductivity and dipolar relaxation of binary dimethylimidazolium chloride–water solutions: a molecular dynamics study, *J. Mol. Liq.* 157 (2010) 163–167, <https://doi.org/10.1016/j.molliq.2010.10.001>.
- [31] N.J. English, D.A. Mooney, S. O'Brien, Ionic liquids in external electric and electromagnetic fields: a molecular dynamics study, *Mol. Phys.* 109 (2011) 625–638, <https://doi.org/10.1080/00268976.2010.544263>.
- [32] C. Desgranges, J. Delhommelle, Molecular simulation of transport in nanopores: application of the transient-time correlation function formalism, *Phys. Rev. E* 77 (2008) 027701, <https://doi.org/10.1103/PhysRevE.77.027701>.
- [33] A.P. Thompson, H.M. Aktulga, R. Berger, D.S. Bolintineanu, W.M. Brown, P.S. Crozier, P.J. in't Veld, A. Kohlmeyer, S.G. Moore, T.D. Nguyen, T.D. Nguyen, R. Shan, M.J. Stevens, J. Tranchida, C. Trost, S.J. Plimpton, LAMMPS - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales, *Comput. Phys. Commun.* 271 (2022) 108171, <https://doi.org/10.1016/j.cpc.2021.108171>.
- [34] D. Huang, Z. Qin, Q. Liu, N. Podhorszki, S. Klasky, Identifying challenges and opportunities of in-memory computing on large hpc systems, *J. Parallel Distrib. Comput.* 164 (2022) 106–122, <https://doi.org/10.1016/j.jpdc.2022.02.002>.
- [35] D.J. Evans, G.P. Morriss, Nonlinear-response theory for steady planar Couette flow, *Phys. Rev. A* 30 (1984) 1528–1530, <https://doi.org/10.1103/PhysRevA.30.1528>.
- [36] P.J. Daivis, B.D. Todd, A simple, direct derivation and proof of the validity of the SLLD equations of motion for generalized homogeneous flows, *J. Chem. Phys.* 124 (19) (2006) 194103, <https://doi.org/10.1063/1.2192775>.
- [37] A.W. Lees, S.F. Edwards, The computer study of transport processes under extreme conditions, *J. Phys. Part C: Solid State Phys.* 5 (1972) 1921–1929, <https://doi.org/10.1088/0022-3719/5/15/006>.
- [38] S. Nosé, A molecular dynamics method for simulations in the canonical ensemble, *Mol. Phys.* 52 (1984) 255–268, <https://doi.org/10.1080/00268978400101201>.
- [39] W.G. Hoover, Canonical dynamics: equilibrium phase-space distributions, *Phys. Rev. A* 31 (1985) 1695–1697, <https://doi.org/10.1103/PhysRevA.31.1695>.
- [40] L. Verlet, Computer experiments on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules, *Phys. Rev.* 159 (1967) 98–103, <https://doi.org/10.1103/PhysRev.159.98>.

- [41] J.D. Weeks, D. Chandler, H.C. Andersen, Role of repulsive forces in determining the equilibrium structure of simple liquids, *J. Chem. Phys.* 54 (1971) 5237–5247, <https://doi.org/10.1063/1.1674820>.
- [42] J.E. Jones, On the determination of molecular fields.-ii. from the equation of state of a gas, *Proc. R. Soc. Lond., Ser. A* 106 (1924) 463–477, <https://doi.org/10.1098/rspa.1924.0082>.
- [43] W.T. Ashurst, W.G. Hoover, Dense-fluid shear viscosity via nonequilibrium molecular dynamics, *Phys. Rev. A* 11 (1975) 658–678, <https://doi.org/10.1103/PhysRevA.11.658>.
- [44] C. Gattinoni, S. Mackowiak, D.M. Heyes, A.C. Branka, D. Dini, Boundary-controlled barostats for slab geometries in molecular dynamics simulations, *Phys. Rev. E* 90 (2014) 043302, <https://doi.org/10.1103/PhysRevE.90.043302>.
- [45] P.A. Thompson, M.O. Robbins, Shear flow near solids: epitaxial order and flow boundary conditions, *Phys. Rev. A* 41 (1990) 6830–6837, <https://doi.org/10.1103/physreva.41.6830>.
- [46] K. Ogawa, H. Oga, H. Kusudo, Y. Yamaguchi, T. Omori, S. Merabia, L. Joly, Large effect of lateral box size in molecular dynamics simulations of liquid-solid friction, *Phys. Rev. E* 100 (2019) 023101, <https://doi.org/10.1103/PhysRevE.100.023101>.
- [47] B.D. Todd, D.J. Evans, P.J. Daivis, Pressure tensor for inhomogeneous fluids, *Phys. Rev. E* 52 (1995) 1627–1638, <https://doi.org/10.1103/PhysRevE.52.1627>.
- [48] J.S. Hansen, B.D. Todd, P.J. Daivis, Prediction of fluid velocity slip at solid surfaces, *Phys. Rev. E* 84 (2011) 016313.
- [49] G. Ciccotti, J. Jacucci, I.R. McDonald, “Thought-experiments” by molecular dynamics, *J. Stat. Phys.* 21 (1979) 1–22.