# Simulating dense, non-Brownian suspension rheology using LAMMPS

Christopher Ness

*School of Engineering, University of Edinburgh, Edinburgh, EH9 3JL, United Kingdom*

Dense suspensions are widespread in nature, manufacturing and process engineering. Particle-based simulations have proven to be an invaluable complement to experimental rheological characterisation, serving as a virtual rheometer that enables rapid exploration of parameter space and detailed scrutiny of microscopic dynamics. To maximise the utility of such simulations, it can be advantageous to exploit pre-existing, well-optimised, well-documented codes. Here we provide a simple description of how to use LAMMPS to study the rheology of dense, non-Brownian suspensions.

## I. INTRODUCTION

Suspensions of micron-sized particles with solid and liquid mixed in roughly equal proportion present intriguing flow properties that challenge physicists and engineers of all kinds [1]. A useful starting point for characterising their rheology is to understand the rate-independent behaviour (relevant for solid particles of $\approx 10-1000\mu$m size), demonstrated experimentally by Boyer *et al.* [2] and later reviewed by many others (e.g. [3]). A number of particle-based simulations reproduce the rate-independent rheology (e.g. [4, 5]), providing (i) corroboration of the experimental result; (ii) a source of particle-resolved data inaccessible experimentally; and (iii) a platform for examining systematically more complex microphysics, for instance particle inertia [6], stress-induced friction [7] and adhesion [8]. Many simulation techniques and codes are available, and we do not review these here. In what follows, we describe how to generate numerical rheology data consistent with the rate-independent result using LAMMPS [9, 10].

The simulation technique shares many details with traditional molecular dynamics, and the reader is expected to be familiar with basic concepts including contact detection, neighbour listing, timestepping and so on. Our specific model is more commonly labelled as a 'discrete element method' owing to its similarity to approaches used in granular physics, specifically the absence of thermal forces and the inclusion of particle-particle friction (both appropriate for the size range mentioned above). The basic approach for obtaining a numerical rheology measurement is to (i) initialise the system with a packing of non-overlapping spherical particles of desired size distribution at a desired volume fraction $\phi$; (ii) evaluate the trajectory of each particle $i$ by numerically solving "$F = ma$" in the presence of a prescribed background fluid velocity gradient $\nabla\boldsymbol{u}^\infty$, and a set of pairwise hydrodynamic and contact interactions. (We assume $\phi \gtrsim 0.4$ throughout, otherwise a more detailed account of the hydrodynamics is required.) When desired, a bulk stress tensor $\Sigma$ is calculated from the interaction forces and particle positions, thus generating rheology data *viz.* the stress $\Sigma$ as a function of deformation rate $\mathbb{E}$ (with $\mathbb{E} \equiv \frac{1}{2}(\nabla\boldsymbol{u}^\infty + \nabla\boldsymbol{u}^{\infty\mathrm{T}})$) and volume fraction $\phi$.

## II. DIMENSIONAL ANALYSIS FOR RATE-INDEPENDENT SUSPENSIONS

It is instructive to consider first a dimensional analysis. This introduces all of the quantities (and their units) that we will define when setting up a simulation. We consider a dense, non-Brownian (so we omit $k_b T$ from the following) suspension of stiff spheres under a flow with imposed shear rate. The principle particle properties (these set the length, mass and time scales) are the characteristic particle radius $a$ [length], the particle density $\rho$ [mass/length$^3$] (taken throughout to be equal to the fluid density), and the particle normal stiffness $k_n$ [mass/time$^2$] (this has a tangential counterpart $k_t$). With respect to these quantities, 1 time unit corresponds to the inverse frequency of a mass $m = 1$ on a linear spring with stiffness $k_n = 1$. The remaining material properties to be defined are the fluid viscosity $\eta_\mathrm{f}$ [mass/(length×time)] and the particle-particle friction coefficient $\mu$ [dimensionless], relevant for micron sized (and larger) particles. The relevant macroscopic quantities are the size of the simulation box $L$ [length] and the volume fraction $\phi$ [dimensionless] therein. The background fluid flow is characterised by a velocity field $\boldsymbol{u}^\infty$ [length/time] and its gradient (a tensor) $\nabla\boldsymbol{u}^\infty$ [1/time] (that we specify, and take to be spatially uniform), the time $t$ for which the flow was applied, and a stress tensor $\Sigma$ [mass/(time$^2$×length] (that we measure). We write a scalar velocity gradient as $\dot{\gamma}$ ($\equiv \partial u_x/\partial y$) and a scalar stress as $\Sigma_{xy}$ (the $xy$ component of $\Sigma$). A list of nondimensional parameters necessary to fully define a given suspension under given flow conditions is then:

(i) $a/L$    (ii) $a\dot{\gamma}/\sqrt{k_n/(\rho a)}$    (iii) $\rho\dot{\gamma}a^2/\eta_\mathrm{f}$    (iv) $\mu$    (v) $\phi$    (vi) $\eta_r \equiv \Sigma_{xy}/\eta_\mathrm{f}\dot{\gamma}$    (vii) $\dot{\gamma}t$

Setting (i)-(iii) to be $\ll 1$ ensures, respectively, bulk conditions, stiff particles and no particle inertia. Under these conditions, and assuming $\mu$ is constant (i.e. particle friction is Coulombic) and we shear to steady state ($\dot{\gamma}t \to \infty$), we have simply that $\eta_r = \eta_r(\phi)$, hence the label 'rate-independent'. This is consistent with the result of Boyer *et al.* [2], and will be the focus of our example below.

## III. PARTICLE-LEVEL FORCES AND SHEARING

Rate-independent rheology is obtained by subjecting particles to three types of force and torque: Stokes drag, pairwise lubrication and pairwise contact. The full form of these is reported by several authors [5, 11–13] and need not be repeated here. Instead we describe the forces in simplified terms.

The Stokes drag (Fig. 1(a)) on particle $i$ (radius $a_i$) is proportional to the difference between its velocity $\boldsymbol{u}_i$ and the fluid streaming velocity at its centre $\boldsymbol{u}^\infty(\boldsymbol{x}_i)$:

$$\boldsymbol{F}_i^d = 6\pi\eta_{\mathrm{f}} a_i(\boldsymbol{u}_i - \boldsymbol{u}^\infty(\boldsymbol{x}_i)). \tag{1}$$

This force is essentially what induces flow in the simulation, causing particles to conform to the streaming velocity set by $\boldsymbol{u}^\infty$. Similarly, a torque acts to cause the particles to rotate with angular velocity set by $\frac{1}{2}(\nabla \times \boldsymbol{u}^\infty)$. Neighbouring particles $i$ and $j$ with centre-to-centre vector $\boldsymbol{r}_{i,j}$ (Fig. 1(b)[i]) experience lubrication forces (see [14, 15]) dependent on the gap $h$ between them and their relative velocity (Fig. 1(b)[ii]). The leading term of the force on particle $i$ scales with $1/h$ and the normal component of the pairwise velocity difference:

$$\boldsymbol{F}_{i,j}^l = \frac{3}{2}\pi a_i^2 \eta_{\mathrm{f}} \frac{1}{h}(\boldsymbol{u}_j - \boldsymbol{u}_i)_n. \tag{2}$$

These lubrication forces oppose relative motion between particle pairs. They are prevented from diverging at contact by setting a lower limit on the allowed value of $h$ (typically $\mathcal{O}(10^{-3}a_i)$). A torque also acts to resist relative rotation between $i$ and $j$. Overlapping particle pair $i$ and $j$ (Fig. 1(c)[i]) experience repulsive contact forces dependent upon the scalar overlap $\delta$ (Fig. 1(c)[ii]) and the tangential displacement accumulated over the duration of the contact $\boldsymbol{\xi}$ (Fig. 1(c)[iii]):

$$\boldsymbol{F}_{i,j}^c = k_n \delta \boldsymbol{r}_{i,j}/|\boldsymbol{r}_{i,j}| - k_t \boldsymbol{\xi}. \tag{3}$$

The friction coefficient $\mu$ sets an upper bound on $\boldsymbol{\xi}$ through $|\boldsymbol{\xi}| \leq \mu k_n \delta/k_t$

The stress contribution from drag forces is proportional to $\mathbb{E}$. The $\alpha, \beta$ component of the stress due to lubrication and contact is found, respectively, by summing $(F_{i,j}^{l,\alpha} r_{i,j}^\beta + F_{i,j}^{l,\beta} r_{i,j}^\alpha)/2$ and $F_{i,j}^{c,\alpha} r_{i,j}^\beta$ over all pairs. The forces are summed on each particle and the trajectories are then updated according to Newtonian dynamics, using a numerical scheme with timestep chosen to be small compared to $\sqrt{m/k_n}$.

In LAMMPS the simulation box deforms according to the specified $\nabla \boldsymbol{u}^\infty$. For instance, when the only nonzero element of $\nabla \boldsymbol{u}^\infty$ is an off-diagonal, shearing is applied by tilting the *triclinic* box (at fixed volume) according to
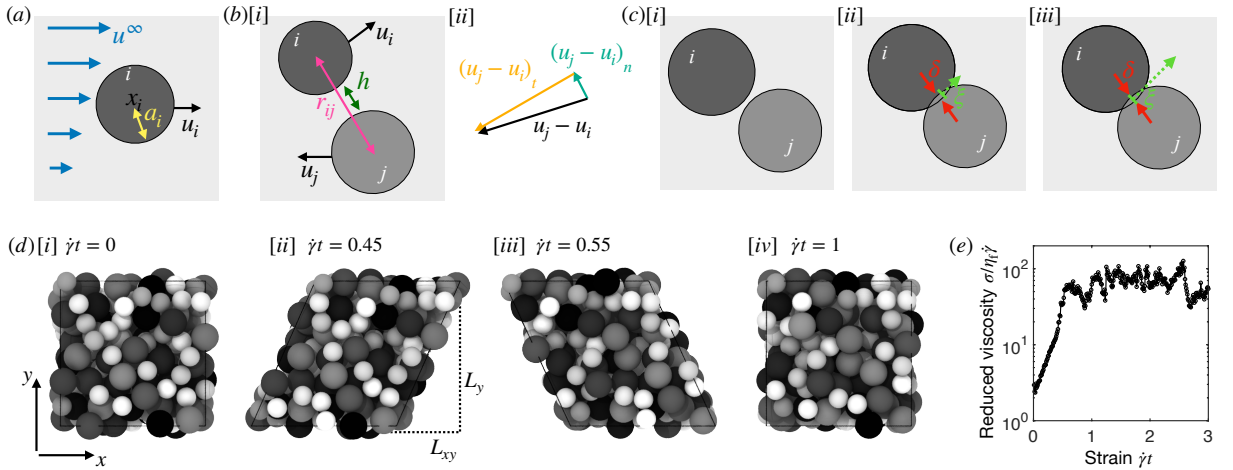


FIG. 1. Particle-level physics in a simplified dense suspension. (a) A particle (radius $a_i$, position $\boldsymbol{x}_i$, velocity $\boldsymbol{u}_i$) in a fluid with streaming velocity $\boldsymbol{u}^\infty$; (b) Pairwise lubrication interaction showing [i] particle velocities $\boldsymbol{u}_i$, $\boldsymbol{u}_j$, centre-to-centre vector $\boldsymbol{r}_{ij}$ and surface separation $h$; [ii] relative velocity $(\boldsymbol{u}_j - \boldsymbol{u}_i)$ and its components normal $(\boldsymbol{u}_j - \boldsymbol{u}_i)_n$ and tangential $(\boldsymbol{u}_j - \boldsymbol{u}_i)_t$ to $\boldsymbol{r}_{ij}$; (c) Contact force showing [i] particles approaching; [ii] particles entering contact with overlap $\delta$ and initiation of sliding with $\boldsymbol{\xi} = 0$; [iii] contact with overlap $\delta$ and accumulated, nonzero, tangential stretch $\boldsymbol{\xi}$. (d) Subjecting the suspension to a steady simple shear. Shown are increments of the strain ($\dot{\gamma}t = 0$ [i], 0.45 [ii], 0.55 [iii], 1 [iv]) illustrating the remapping procedure used by LAMMPS. (e) Reduced viscosity $\sigma/\eta_{\mathrm{f}}\dot{\gamma}$ as a function of the acculated strain $\dot{\gamma}t$.

$L_{xy}(t) = L_{xy}(t_0) + L_y\dot{\gamma}t$, Fig. 1(d). When the strain ($\gamma = \dot{\gamma}t$) reaches 0.5 in this example, the system is remapped to a strain of -0.5. This has no effect on the particle-particle forces or on the stress, and is simply a numerical tool to permit unbounded shear deformation while preventing the domain from becoming elongated in one axis.

## IV. LAMMPS INPUTS AND OUTPUTS

The above physics are implemented in LAMMPS (Large-scale Atomic/Molecular Massively Parallel Simulator) [9], a classical molecular dynamics code written in C++. The LAMMPS documentation should be referred to at all times [10]. A skeletal set of instructions for downloading and compiling LAMMPS and running a dense, non-Brownian suspension rheology simulation are provided at Ref [16]. Our strategy is to separate the generation and shearing of suspensions into two distinct simulations. The first defines dimensionless numbers (i) and (v) above, producing non-overlapping particle packings in a cuboidal, periodic domain of fixed $\phi$; the second defines (ii), (iii), (iv) and (vii) then applies the deformation allowing us to measure (vi). This decomposition allows one to build a library of configurations at different $\phi$ that can be reused many times for different deformation protocols. LAMMPS takes as its input a text file containing a list of commands and their arguments. In the following we don't describe each of these in detail, but instead provide minimal scripts, indicating where the physics above enters. Each of the commands is fully described in the LAMMPS documentation [10].

```
in.create

# SETTINGS
atom_style sphere
comm_modify mode single vel yes

# GENERATE A CUBIC, PERIODIC SIMULATION BOX
boundary p p p
region reg prism 0 14.6381 0 14.6381 0 14.6381 0 0 0 units box
create_box 2 reg

# GENERATE THE PARTICLES AND SPECIFY THEIR SIZE
create_atoms 1 random 100 123456 NULL
create_atoms 2 random 100 123457 NULL
set type 1 diameter 2.8
set type 2 diameter 2
set type 1 density 1
set type 2 density 1

# SPECIFY THE PARTICLE-PARTICLE INTERACTION
pair_style granular
pair_coeff * * hooke 10000 0 tangential linear_history 7000 0 0.1

# SPECIFY THE OUTPUTS
thermo 10000
dump id all custom 10000 create.dump id x y z radius
log create.log

# SPECIFY THE TIMESTEP, THE INTEGRATION SCHEME AND RUN
timestep 0.0001
fix 1 all nve/sphere
fix 2 all viscous 1.88
run 1000000

# WRITE AN OUTPUT FILE
write_data data.file
```

*Creating particle packings.–* For the time being, it isn't necessary to specify the 'full' physics described above, nor do we need to output the stresses. Rather, we need just enough detail to create assemblies of non-overlapping particles. We therefore omit lubrication forces at this stage or simplicity. We first generate particles (of two types, each having a different radius) with random coordinates in a box of set dimensions. Their overlaps generate contact forces (following Eq. 3) that lead to motion; damping against a stationary background fluid (i.e. $\boldsymbol{u}^\infty = 0$ in Eq. 1) extracts energy until the system comes to rest. The properties of the configuration of particles produced by this script don't really matter: we are not trying to sample an 'equilibrium' configuration (this is not relevant for non-Brownian

systems) but simply create a packing with no (or minimal) overlaps. Shown in the `in.create` panel is an example input script ('#' indicates comments) to generate a suspension with $\phi = 0.5$.

Highlighted in red from top to bottom are (with units as stated earlier): (i) the size of the cubic simulation box $L = 14.6381$; (ii) the numbers and radii of particles of types 1 and 2: $N_1 = N_2 = 100$, $2a_1 = 2$ and $2a_2 = 2.8$, thus setting the volume fraction as $\phi = 4/3\pi(N_1 a_1^3 + N_2 a_2^3)/L^3 = 0.5$; (iii) the particle density $\rho$; (iv) the particle stiffness $k_n = 10000$ (with $k_t = 7k_n/10$, a conventional choice); (v) the particle-particle friction coeffient $\mu = 0.1$; (vi) the fluid viscosity $\eta_f = 0.1$ (the number we put in the script is $6\pi\eta_f a$); (vi) the timestep; (vii) the number of timesteps to run. Highlighted in blue are the seeds used to generate the initial particle positions. New realisations can be generated by rerunning the simulation with different numbers here. This script produces `data.file`, containing a snapshot of the system after the final timestep, to be read in by future scripts. The file contains a list of the particle ID, diameter, density, coordinates $(x, y, z)$ and velocity components $(x, y, z)$. It also produces a dump file (create.dump) that can be visualised using e.g. Ovito.

*Shearing particle packings.–* The second script (`in.run`, below) takes `data.file` as an input and applies a deformation to the sample. The key inputs to this script are the parameters related to the particle-particle interaction ($k_n$, $k_t$, $\mu$ and $\eta_f$) and the shear rate, set by specifying the components of $\nabla\boldsymbol{u}^\infty$. Through these we set the values of dimensionless control parameters (ii), (iii), (iv) and (vii) listed above. The remaining content of the input script is concerned with specifying the bulk stress calculation and requesting it as an output, necessary for obtaining (vi).

```
in.run

# SETTINGS
atom_style sphere
comm_modify mode single vel yes
newton off

# READ THE PARTICLE CONFIGURATION
read_data data.file

# SPECIFY THE PARTICLE-PARTICLE INTERACTION
pair_style hybrid/overlay granular lubricate/bmpoly 0.1 1 1 0.001 0.05 1 0
pair_coeff * * granular hooke 10000 0 tangential linear_history 7000 0 0.1
pair_coeff * * lubricate/bmpoly

# DO THE STRESS CALC
compute Temp all temp/profile 1 1 1 xyz 10 10 10
compute str all pressure Temp

# SPECIFY THE OUTPUTS
thermo_style custom time c_str[1] c_str[2] c_str[3] c_str[4] c_str[5] c_str[6]
thermo 10000
thermo_modify format line "%.0f %12.4e %12.4e %12.4e %12.4e %12.4e %12.4e"
dump id all custom 10000 run.dump id x y z radius
log run.log

# SPECIFY THE TIMESTEP, THE INTEGRATION SCHEME AND RUN
timestep 0.0001
fix 1 all nve/sphere
fix 2 all deform 1 xy erate 0.01 remap v
run 3000000
```

Highlighted from top to bottom (units as before) are the fluid viscosity $\eta_f = 0.1$, the particle stiffness $k_n = 10000$ ($k_t = 7000$) and friction coefficient $\mu = 0.1$ and the shear rate $\dot{\gamma} = 0.01$. Using the command `fix deform` we have specified just the $xy$ component of $\nabla\boldsymbol{u}^\infty$, $\dot{\gamma}$ as defined above. The other components are 0 by default, so this leads to a simple shear with flow in $x$ and gradient in $y$ i.e. $\boldsymbol{u}^\infty = (\dot{\gamma}y, 0, 0)$. The script runs at this $\dot{\gamma}$ for 3000000 timesteps, each of duration 0.0001. Thus the total shear strain is $\dot{\gamma}t = 0.001 \times 3000000 \times 0.0001 = 3$. Note that the `fix viscous` command is not required here because the drag force is applied within the lubrication pair style.

*Outputs from shearing simulation.–* There are two different types of output produced by LAMMPS during a simulation run, *log* files and *dump* files. Log files are typically described as containing thermodynamic data, but for our purposes we can interpret this as bulk suspension or derived properties, usually the components of the stress tensor $\Sigma$, but also e.g. the average particle contact number. As specified in `in.run` above, the log file (`run.log`) contains the accumulated simulation time $t$, followed by the 6 unique components of the stress tensor in order ($xx$, $yy$, $zz$, $xy$, $xz$, $yz$), output every 10000 timesteps (specified by the `thermo` command):

```
run.log

Time c_str[1] c_str[2] c_str[3] c_str[4] c_str[5] c_str[6]
0 5.3321e-03 8.0767e-03 1.9993e-02 4.9005e-03 -3.7919e-03 -1.1931e-02
1 1.1406e-04 5.8445e-05 1.7531e-04 -1.8081e-04 -5.3399e-05 9.7051e-06
2 1.0354e-04 4.2638e-05 5.3653e-05 -1.6733e-04 -3.1683e-05 -8.8105e-06
3 7.5441e-05 9.2299e-06 5.3539e-05 -2.1836e-04 -1.2350e-05 2.1177e-06
...
```

These stress components have units [mass/(time$^2$×length)]. In order to express in dimensionless units (the "reduced viscosity" as it is conventionally represented) for comparison to experimental data, one must divide by $\eta_\mathrm{f}\dot{\gamma}$. Shown in Fig. 1(e) is a plot of the reduced viscosity $\eta_r = \Sigma_{xy}/\eta_\mathrm{f}\dot{\gamma}$ as a function of the accumulated strain. The reduced viscosity is in this instance taken simply as the $xy$ component of the stress ($\Sigma_{xy}$, the 4th column of the stress outputs in the log file) divided by $\eta_\mathrm{f}\dot{\gamma}$ whereas the strain is the accumulated time $t$ multiplied by $\dot{\gamma}$. From the content of the log file one might also compute e.g. the viscous number $\eta_\mathrm{f}\dot{\gamma}/P$ (with $P$ the mean of the diagonal components of $\Sigma$ ($\Sigma_{xx}$, $\Sigma_{yy}$, $\Sigma_{zz}$), the normal stress differences $\Sigma_{xx} - \Sigma_{yy}$, $\Sigma_{yy} - \Sigma_{zz}$ and so on.

Dump files contain particle-level information (positions, velocities, radii) or contact level information (forces, relative positions), usually output at fixed intervals. In the script above this is specified (by the `dump` command) to be every 10000 timesteps. This example script produces `run.dump`, which lists the particle IDs, positions and radii at specified time intervals. This file might be used for post-processing, for instance to compute structural properties, to follow particle trajectories or to be read directly into various visualisation packages (Fig. 1(d) was generated using Ovito [17], for instance).

```
run.dump

ITEM: TIMESTEP
0
ITEM: NUMBER OF ATOMS
200
ITEM: BOX BOUNDS xy xz yz pp pp pp
0.0000000000000000e+00 1.4638100000000000e+01 0.0000000000000000e+00
0.0000000000000000e+00 1.4638100000000000e+01 0.0000000000000000e+00
0.0000000000000000e+00 1.4638100000000000e+01 0.0000000000000000e+00
ITEM: ATOMS id x y z radius
1 1.60897 10.2815 13.7433 1.4
2 1.15501 3.33547 8.51895 1.4
3 10.2117 13.6329 5.68012 1.4
...
```

By repeating this pair of simulations at a range of $\phi$ (e.g. by changing the value of $L$ in `in.create`) and taking time (and ensemble) averages of the components of $\Sigma$, one can reproduce the rate-independent rheology of Boyer *et al.* [2]. Moreover, one may relax the conditions specified in the dimensional analysis to explore additional physics: (i) introducing particle inertia by increasing the value of $\rho\dot{\gamma}a^2/\eta_\mathrm{f}$; (ii) introducing particle softness by increasing the value of $a\dot{\gamma}/\sqrt{k_n/(\rho a)}$, and so on.

## V.  CLOSING REMARKS

We have provided a brief description of how to use the molecular dynamics code LAMMPS to generate dense, non-Brownian suspension rheology data. Examples of the use of this approach to study the physics of dense suspensions can be found in Refs XMoreover, the flexibility of the code allows one to simulate more complex geometries, both within LAMMPS and within derivative codes such as LIGGGHTS.

[1] C. Ness, R. Seto,  and R. Mari, arXiv 2105.04162 (2021).
[2] F. Boyer, É. Guazzelli,  and O. Pouliquen, Physical Review Letters **107**, 188301 (2011).

[3] É. Guazzelli and O. Pouliquen, Journal of Fluid Mechanics **852**, P1 (2018).

[4] S. Gallier, E. Lemaire, F. Peters, and L. Lobry, Journal of Fluid Mechanics **757**, 514–549 (2014).

[5] O. Cheal and C. Ness, Journal of Rheology **62**, 501 (2018).

[6] M. Trulsson, B. Andreotti, and P. Claudin, Physical Review Letters **109**, 118305 (2012).

[7] R. Seto, R. Mari, J. F. Morris, and M. M. Denn, Physical Review Letters **111**, 218301 (2013).

[8] A. Singh, S. Pednekar, J. Chun, M. M. Denn, and J. F. Morris, Physical Review Letters **122**, 098004 (2019).

[9] S. Plimpton, Journal of Computational Physics **117**, 1 (1995).

[10] S. Plimpton, "Lammps molecular dynamics simulator," Accessed: 16-07-2021.

[11] M. Trulsson, B. Andreotti, and P. Claudin, Physical Review Letters **109**, 118305 (2012).

[12] R. Mari, R. Seto, J. F. Morris, and M. M. Denn, Journal of Rheology **58**, 1693 (2014).

[13] Z. Ge and L. Brandt, arXiv 2005.12755 (2020).

[14] S. Kim and S. J. Karrila, *Microhydrodynamics: principles and selected applications* (Butterworth Heinemann, 1991).

[15] D. Jeffrey, Physics of Fluids A: Fluid Dynamics **4**, 16 (1992).

[16] C. Ness, "christopherjness.github.io/lammps," Accessed: 16-07-2021.

[17] A. Stukowski, Modelling and Simulation in Materials Science and Engineering **18**, 015012 (2009).