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, 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, though is more commonly labelled as the '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. 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 ϕ; (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. The integration is ordinarily done using a timestepping algorithm such as Velocity-Verlet. When desired, a bulk stress tensor $\mathbb{\Sigma}$ is calculated from the interaction forces and particle positions, thus generating rheology data *viz.* the stress $\mathbb{\Sigma}$ 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 ϕ.

II. DIMENSIONAL ANALYSIS FOR RATE-INDEPENDENT SUSPENSIONS

It is instructive to consider first a dimensional analysis. This also introduces all of the quantities (and their units) that we will define when setting up a simulation. We consider a dense, non-Brownian 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 particle radius a [length], the particle density ρ [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 η_f [mass/(length×time)] and the particle-particle friction coefficient μ [dimensionless]. The relevant macroscopic quantities are the size of the simulation box L [length] and the volume fraction ϕ [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 $\mathbb{\Sigma}$ [mass/(time^2×length] (that we measure). For convenience we write a scalar velocity gradient as $\dot{\gamma}$ ($\equiv \partial u_x/\partial y$) and a scalar stress as Σ_{xy} (the xy component of $\mathbb{\Sigma}$). A list of nondimensional parameters necessary to fully define a given suspension under given flow conditions is then:

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

Setting (i)-(iii) to be $\ll 1$ ensures, respectively, bulk conditions, stiff particles and no particle inertia. Under these conditions, and assuming μ 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. Not included in this list of parameters are, for instance, the particle shape and size distribution.

III. PARTICLE-LEVEL FORCES AND SHEARING

Rate-independent rheology that follows Boyer *et al.* [2] for dense suspensions ($\phi \gtrsim 0.4$) is obtained by subjecting particles to three types of force and torque: Stokes drag, pairwise lubrication and pairwise contact (for dilute suspensions a more detailed account of the hydrodynamics is required). 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 $\boldsymbol{u}^\infty(\boldsymbol{x}_i)$:

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

Neighbouring particles i and j with centre-to-centre vector $\boldsymbol{r}_{i,j}$ (Fig. 1(b)[i]) experience lubrication forces (see [14, 15]) that have a leading term proportional to the inverse of the gap h between them and to the normal component of their velocity difference (Fig. 1(b)[ii]). The force on particle i is:

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

Overlapping particle pair i and j (Fig. 1(c)[i]) experience contact forces dependent upon the scalar overlap δ (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 stress contribution from drag forces is proportional to the velocity gradient. The α, β 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 $\nabla\boldsymbol{u}$. For instance, when the only nonzero element of $\nabla\boldsymbol{u}$ is an off-diagonal, shearing is applied by tilting the *triclinic* box (at fixed volume) according to $L_\mathrm{xy}(t) = L_\mathrm{xy}(t_0) + L_y\dot{\gamma}t$, 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.

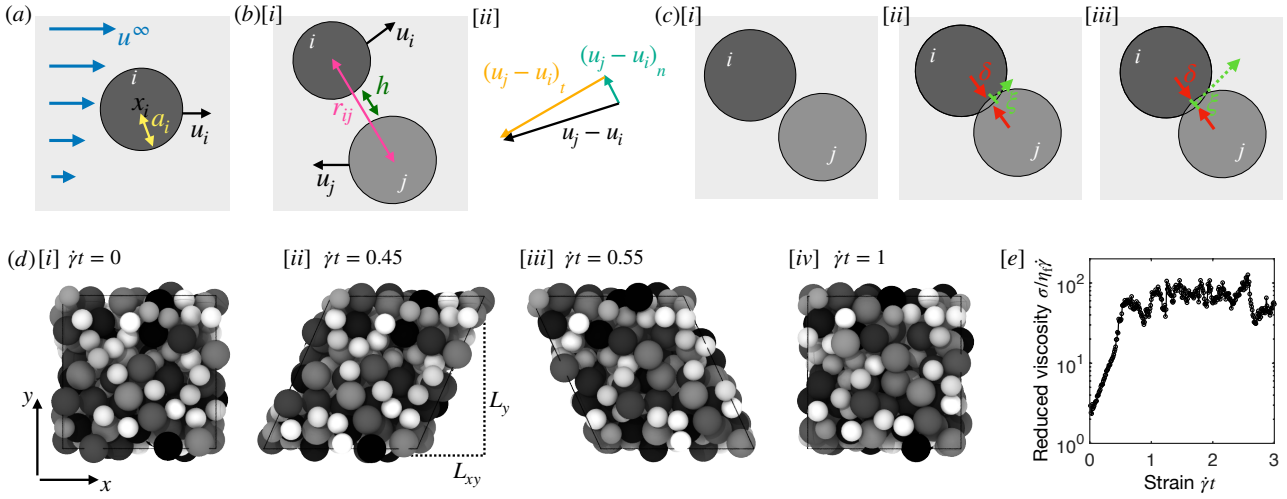

FIG. 1. Particle-level physics in a simplified dense suspension. (a) A particle (radius a_i, position $\boldsymbol{x}_i$, velocity $\boldsymbol{u}_i$) in a fluid with streaming velocity $\boldsymbol{u}^\infty$; (b) Pairwise lubrication interaction showing [i] particle velocities $\boldsymbol{u}_i$, $\boldsymbol{u}_j$, 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 δ and initiation of sliding with $\boldsymbol{\xi} = 0$; [iii] contact with overlap δ 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.

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 dense non-Brownian suspensions into two distinct simulations. The first generates non-overlapping packings in a cuboidal, periodic domain of fixed ϕ; the second applies the deformation. This decomposition allows one to build a library of configurations at different ϕ 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].

Creating particle packings.– For the time being, it isn't necessary to specify the 'full' physics described above. Rather, we need just enough detail to create assemblies of non-overlapping particles. We therefore omit lubrication forces at this stage. We first randomly generating particles. Their overlaps generate contact forces (following Eq 3) that lead to motion; damping against a stationary background fluid (i.e. $\boldsymbol{u}^\infty = 0$) extracts energy until the system comes to rest. The initial configuration of the particles doesn't really matter: we are not trying to sample an 'equilibrium' configuration (this is not relevant for non-Brownian systems) but simply create a system with no overlaps. Below is an example input script ('#' indicates comments) to generate a suspension with $\phi = 0.5$.

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 INTEGRATION SCHEME, THE TIMESTEP AND RUN
fix 1 all nve/sphere
fix 2 all viscous 0.1
timestep 0.0001
run 1000000

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

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 of particles of types 1 and 2 to be added $N_1 = N_2 = 100$; (iii) the radii of particle types 1 and 2: $2a_1 = 2$ and $2a_2 = 2.8$, (iv) the particle density ρ; thus setting the volume fraction as $\phi = 4/3\pi(N_1 a_1^3 + N_2 a_2^3)/L^3 = 0.5$; (v) the particle stiffness $k_n = 10000$ (tangential stiffness $k_t = 7000$, a conventional choice); (vi) the particle-particle friction coeffient $\mu = 0.1$; (vii) the fluid viscosity $\eta_f = 0.1$; (viii) the timestep; (ix) the number of timesteps to run. Highlighted in blue are the random seeds used to generate the particle positions. New realisations can be generated by rerunning the simulation with different numbers here. This script produces a text file (data.file) containing a snapshot of the system to be read in by future scripts. The file contains a list of the particle ID, diameter, density, coordinates and velocity components.

Shearing particle packings.– Below is an input script that takes data.file as its input and applies a shear strain to the sample. The key inputs to this script are the parameters related to the particle-particle interaction (k_n, k_t

and η_f) and the shear rate $\dot{\gamma}$. The remaining content of the input script is concerned with specifying the bulk stress calculation and requesting it as an output (see below).

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 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 data.dump id x y z radius

# 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 are the fluid viscosity $\eta_f = 0.1$ [mass/(length×time)], the particle stiffness $k_n = 10000$ [mass/time^2] and the shear rate $\dot{\gamma} = 0.01$ [1/time]. The script runs at this $\dot{\gamma}$ for 3000000 timesteps, each of duration 0.0001 [time]. Thus the total strain is $\dot{\gamma}t = 0.001 \times 3000000 \times 0.0001 = 3$.

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 $\mathbb{\Sigma}$, but also e.g. the average particle contact number. As specified in `in.run` above, the log file ('log.lammps') 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):

log.lammps

```
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 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 $\eta_f\dot{\gamma}$. Shown in Fig. 1(e) is a plot of the reduced viscosity as a function of the accumulated strain. The reduced viscosity is simply taken as the xy component of the stress (Σ_{xy}, the 4th column of the stress outputs in the log file) divided by $\eta_f\dot{\gamma}$ 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 $\eta_f\dot{\gamma}/P$ (with P the mean of the diagonal components of $\mathbb{\Sigma}$ (Σ_{xx}, Σ_{yy}, Σ_{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 'data.dump', which contains a list of all the particle IDs, positions

and radii. 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).

data.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 ϕ (e.g. by changing the value of L in `in.create`) and taking time (and ensemble) averages of the components of $\mathbb{\Sigma}$, one can reproduce the rate-independent rheology of [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 simplified description of how to use the molecular dynamics code LAMMPS to generate dense, non-Brownian suspension rheology data.

C.N. acknowledges support from the Royal Academy of Engineering under the Research Fellowship scheme and useful discussions with many collaborators and students. Contact: chris.ness@ed.ac.uk.

[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).