Supplementary Materials – Documentation for PSE Plugin (PSEv1)

Andrew M. Fiore
fiorea@mit.edu

PSEv1 is a HOOMD plugin containing a GPU implemention of the Positively Split Ewald (PSE) algorithm for calculation of the Rotne-Prager-Yamakawa (RPY) hydrodynamic mobility and stochastic thermal displacements. The plugin is provided as is, and has been compiled and validated using HOOMD version 1.3.2 and cuda-7.5.

1 File List

The included directories and files are

- `run.py` – Sample script to perform dynamic simulation using the PSEv1 plugin.
- `Configs/` – Hard sphere configurations for 512000 particles at volume fractions $\phi =$ 0.1, 0.2, 0.3, 0.4, 0.5
  - `hsconfig_N512000_phi0.10.xml` – Configuration for $\phi = 0.1$
  - `hsconfig_N512000_phi0.20.xml` – Configuration for $\phi = 0.2$
  - `hsconfig_N512000_phi0.30.xml` – Configuration for $\phi = 0.3$
  - `hsconfig_N512000_phi0.40.xml` – Configuration for $\phi = 0.4$
  - `hsconfig_N512000_phi0.50.xml` – Configuration for $\phi = 0.5$
- `PSEv1/` – Plugin code (files and organization based on the HOOMD pluging template)

2 Software Requirements

The PSEv1 plugin requires the following additional software:

- HOOMD v1.3.2, compiled with CUDA
- CUDA-7.5
- LAPACKE v3.6.1

The plugin requires a working version of HOOMD v1.3.2 against which to be compiled. The HOOMD installation must have GPU acceleration enabled, and HOOMD must be compiled as a shared library in order for plugins to be built against it. The build has been tested using cuda-7.5. In addition to HOOMD and CUDA, the software package LAPACKE must also be installed. Version 3.6.1 of LAPACKE has been used successfully.

The required software is available from the following locations

- HOOMD source: `https://github.com/joaander/hoomd-blue/releases/tag/v1.3.2`
- CUDA: `https://developer.nvidia.com/cuda-75-downloads-archive`
- LAPACKE: `http://www.netlib.org/lapack/`

3 Compiling The Plugin

The `README` file within the `PSEv1` directory contains directories for compiling plugins.

4 Using the Plugin

A sample script to perform a dynamic simulation in HOOMD using the PSEv1 plugin is:

```
from hoomd_script import *
from hoomd_plugins import PSEv1

... ( System setup, Force fields, etc. )

# Set up the PSE integrator
all = group.all()
integrate.mode_standard(dt = dt)

PSEv1.integrate.PSEv1(group=all, seed = 0, T = 1, xi = 0.5, error = 10**(-3.0))

... ( Set up output for desired quantities, run the simulation )
```

The required inputs are

- `group` – The particle group to be integrated and for which the hydrodynamic calculation will be performed. This should be set to `all`, as shown in the sample script.
- `T` – The system temperature, given in reduced units. The reduced temperature is defined as $T^* = k_B T/\mathcal{E}$, where $\mathcal{E}$ is the fundamental unit of energy within the simulation.

The optional inputs are (default values given in parentheses)

- `seed (0)` – Initial seed for random number generation used in the stochastic calculation
- `xi (0.5)` – Ewald splitting parameter. The default value of 0.5 is roughly optimal for many hard sphere configurations. Using smaller values, e.g. 0.2, places more of the calculation in the real space neighbor list-based calculation, while using larger values, e.g. 0.8, shifts calculation cost toward the wave space FFT calculations. Asking for too small a value of xi can lead to a runtime error if the resulting cutoff radius is larger than half the box length. Asking for too large a value of xi can result in memory errors when allocating the FFT. It is recommended to keep xi between 0.1 and 1.0, and close to 0.5.

- `error (0.001)` – Error tolerance for truncation errors, quadrature errors, and Lanczos iteration residual. All calculations are performed in single precision.

If an optional input is not specified, the default value is assumed within the code.

4.1 Sample Problem

A sample simulation for hard spheres at equilibrium can be performed with the provided files by simply copying one of the configuration files into the same directory as the `run.py` file, and changing the name of the input file within `run.py` to match the configuration file. Then, with HOOMD installed, the simulation can be run with the command `PATH_TO_HOOMD/bin/hoomd run.py` where `PATH_TO_HOOMD` is the location of the HOOMD installation. The script creates a directory called `data` in which the particle trajectory (in the form of .dcd and .xml files) and the stress output is saved. The result of computing the pressure from the stress output for each of the volume fractions compared to the Carnahan-Starling result is shown in Figure 1.

Some unit conversion is needed to convert the Carnahan-Starling result into the same units as HOOMD. The dimensionless pressure output by HOOMD is $P^* = a^3 P/k_B T$, where a is the radius of a particle and k_B is Boltzmann's constant. In the Carnahan-Starling equation, the normalization is $P^*_{\mathrm{CS}} = P/nk_B T$, where $n = 3\phi/4\pi a^3$ is the particle number density. Therefore, $P^* = (3\phi/4\pi)\, P^*_{\mathrm{CS}}$. Because HOOMD returns only the virial contribution to the stress, i.e. the stress due to interparticle forces $\langle \mathbf{xF} \rangle / V$, the ideal gas contribution, $nk_B T$ is subtracted from the equation of state when comparing to simulation results, i.e. the following comparison should be made between the virial pressure from HOOMD and the Carnahan-Starling equation of state:

$$\frac{\langle \mathbf{xF} \rangle}{V} = \frac{3\phi}{4\pi}\left(P^*_{CS} - 1\right) = \frac{3\phi}{4\pi}\left(\frac{1 + \phi + \phi^2 - \phi^3}{\left(1-\phi\right)^3} - 1\right) \tag{1}$$

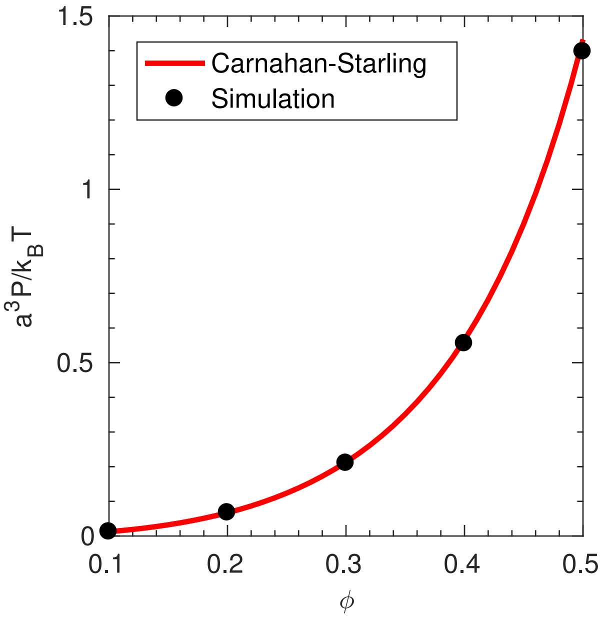

Figure 1: Pressure as a function of volume fraction for the simulation (circles) and Carnahan-Starling prediction (line).